Journal of Information Science and Engineering 39, 91-109 (2023)
DOI: 10.6688/JISE.202301_39(1).0006

A Novel Variable Lie Hypergraph Technique for Cluster Based Routing in Opportunistic Networks

Supriya Sridharan[1], Swaminathan Venkataraman[1,+] and Somasundaram K.[2]
[1]*Department of Mathematics*
School of Arts, Humanities, Science and Education
SASTRA Deemed to be University
Thanjavur, 613401 India
[2]*Department of Mathematics, Amrita School of Engineering*
Amrita Vishwa Vidyapeetham
Coimbatore, 641112 India
E-mail: swaminathan@src.sastra.edu; supriya@src.sastra.ac.in; s_sundaram@cb.amrita.edu

In recent years, breakneck advancements in technology and the proliferation of wireless handheld devices have drawn tremendous interest to study. An opportunistic network (OppNets) refers to a number of wireless nodes opportunistically communicating with each other which does not rely on any fixed structure. Due to this, routing packets from source to destination in OppNets remain a challenging issue. This paper proposes a multi-objective optimization approach for cluster based routing in OppNets that maximizes average delivery ratio, minimizes both the hop count and average delivery delay. We propose a novel Variable Lie hypergraph theory for a unanimous way of clustering and routing protocol to obtain the optimal solution. A variable hypergraph is constructed by combining the Lie commutator. Variable hyperedges are the clusters, and the variable hypergraph transversal is the required set of cluster heads. Nodes of the variable hyperedges are positioned appropriately in an upper triangular matrix which is an element of upper triangular matrix Lie algebra. Furthermore, we propose the upper triangular routing matrix algorithm that finds the path in identifying the neighbour node by its location inside the upper triangular matrix using Lie commutators. Simulation results using real mobility traces are presented, manifesting the effectiveness of the proposed scheme with very less time.

Keywords: OppNets, variable hypergraph, Lie algebra, routing, commutators

1. INTRODUCTION

Networks are now omnipresent for days, and many real-world applications need to explore information in these networks. The fundamental evolution of the Delay Tolerant Network is an opportunistic network which is self-organizing with high mobility [1]. The transfer of data between nodes is a major issue due

Received September 1, 2021; revised November 4, 2021; accepted December 13, 2021.
Communicated by Xiaohong Jiang.
+ Corresponding author: swaminathan@src.sastra.edu

to intermittent communication links [2]. All nodes in the network are capable of movement and can be dynamically connected arbitrarily, which makes routing critical [3]. The significant challenges in routing are to find its relay node to carry forward packets to the destination depending on the connectivity.

In order to better understand the limitations of opportunistic network data sharing, lack of awareness of node's mobility pattern, inconsistency in connectivity and network topology and to eradicate, this paper proposes Multi-Objective Optimization (MOO) [4] which combines many objectives providing optimized cluster-based routing in OppNets. The Pareto optimal solution for MOO is obtained by a novel variable hypergraph. A variable hypergraph is introduced with the spark from variable set theory [5] and hypergraph. Variable hypergraph is constructed by Lie commutators (operator of Lie algebra). Lie theory is concerned with different areas of pure mathematics and various mathematical applications such as algebra, analysis, topology, fractional, ordinary and partial differential equations and many more but real time applications in Lie theory is very limited as not yet much explored.

In the literature, Lie algebra has not been associated with an n-ary structure called hypergraph (as well as variable hypergraph). But, representing structures by hypergraph-based methods has been recently increasing because of its n-ary relations [6], that is, it allows vertices to be multiply connected by hyperedges. Hypergraphs can be modelled in plenty of ways, as in [7]. For this reason, various practical problems like image processing [8], DNA sequencing [9], networking [10] and so on use this representation. Here, we introduce variable hypergraph theory, which handles parameters like time for clustering the nodes of OppNets. Cluster Heads (CH) are elected using the variable hypergraph transversal property. Finally, a novel routing algorithm is proposed using Lie commutators which identifies relay nodes by the position of node inside an upper triangular matrix.

1.1 Related Work

Some significant works in the realm of routing in OppNets have been given in this section.

Wireless communication is the skyrocketing technology that greatly increases the data transmission, for which numerous routing protocols are proposed. Recently, Yang Xu *et al*. [11] proposed a secure routing with the help of incentive jammers. Source rewards selfish jammers in artificial jamming by designing an incentive mechanism. Later the Stackelberg game of two stages is utilized to circumscribe jamming power and rewards, finally, routing is employed by Dijkstra's or Bellman-Ford algorithm. In [12], Yang Xu *et al*. proposed QoS data transfer scheme by incentive mechanism for security improvement in artificial jamming for multi-hop wireless networks.

OppNet is the rapid evolution of wireless technology in short-range, Epidemic routing is the most basic protocol for OppNets proposed by Vahdat and Becker [13]. It is a flooding-based protocol; here, an exchange of messages is done when two nodes are encountered. It suffers from high overhead to overcome this Spyropoulos *et al*. [14] proposed Spray and Wait technique which combines the direct and epidemic routing. In the Spray phase, copies of messages are sprayed to neighbour nodes, and in the Wait phase message is transmitted to a destination node.

Later, fully context-aware protocols are developed to enhance the performance of the routing. In [15], Musolesi and Cecilia introduced Context-Aware Routing (CAR) to select the best forwarded for message transmission by Kalman-filter-based prediction approach. In HBPR [16], the pattern of nodes mobility is prophesied by node's characteristics. A utility metric is measured based on the prior behaviour of the node to transfer data.

Sharma *et al*. [17] unite the benefits of two routing schemes, namely, context-aware and oblivious, for a higher delivery ratio with reduced overhead. kROp is proposed by employing k-means clustering with a novel evaluation function to select the best neighbours to forward the message. To overcome the message flooding in earlier proposed methods, Khalid *et al*. [18] designed a Fuzzy-based Check-and-Spray Geocast (FCSG) protocol, in which relay nodes are selected by fuzzy-controller with check and spray mechanism.

Unlike previously discussed work, this paper proposes a novel cluster based routing protocol by introducing the variable hypergraph theory for clustering and matrix position based best forwarder is elected for routing by Lie commutators of an upper triangular matrix.

1.2 Major Contributions

- Formulating in concert both clustering and routing problems in OppNets by Multi-Objective Optimization problem with objectives, maximizing average delivery ratio and minimizes both the average delivery delay and hop count.
- Novel variable hypergraph construction using Lie commutator is introduced for optimized clustering. Cluster heads are determined by variable hypergraph transversal property.
- Novel Upper Triangular Routing Matrix protocol (UTRM), which finds the relay node between source to destination using the element of upper triangular matrix Lie algebra, is proposed.

1.3 Paper Organization

The remnant of the paper is organized as follows. Section 2 reviews some of the preliminaries and Section 3 describes the system model with multi objective optimization problem. Section 4 presents a novel theoretical framework of variable hypergraph construction using Lie commutator and the algorithms for unanimous clustering and routing technique. Section 5 gives experimental results and discussion. Section 6 concludes with a short overview of contributions.

2. PRELIMINARIES

Definition 1 *A Lie algebra* $\mathbf{g}$ *is a vector space with a second bilinear inner composition law* $([.,.])$ *called the bracket product or Lie bracket, which satisfies* $[\theta, \theta] = 0,\ \ \textit{for all}\ \ \theta \in \mathbf{g}$ *and* $J(\theta, \gamma, \omega) = 0,\ \ \textit{for all}\ \ \theta, \gamma, \omega \in \mathbf{g}$ *where* J *is the Jacobiator defined as,* $J(\theta, \gamma, \omega) = [[\theta, \gamma], \omega] + [[\gamma, \omega], \theta] + [[\omega, \theta], \gamma]$ *known as Jacobi identity.*

Table 1. Notations.

$\mathbf{g}$	Lie algebra
$\mathcal{B}$	Basis of $\mathbf{g}$
$\mathbf{g_d}$	Lie algebra of upper triangular matrices
$\mathcal{B}_d$	Basis of $\mathbf{g}_d$
$\mathcal{H}$	Hypergraph
$\mathcal{V}_{\mathcal{H}}$	Variable hypergraph
$\mathcal{V}_{\mathcal{S}}$	Variable set
$\mathcal{V}_{\mathcal{X}} = \{o_i \mid i = 1, 2, \ldots, p\}$	Variable vertex set of $\mathcal{V}_{\mathcal{H}}$
$\mathcal{V}_{\mathcal{E}} = \{\alpha_i \mid i = 1, 2, \ldots, q\}$	Variable hyperedges of $\mathcal{V}_{\mathcal{H}}$
CH	Cluster Head
$\lvert\alpha_i\rvert$	Number of vertices in the variable hyperedge α_i
$V(\alpha_i)$	Vertices inside α_i
$\mathcal{V}_{\mathcal{H}\mathcal{T}}$	Variable hypergraph transversal
$\mathcal{T}$	Transversal of hypergraph $\mathcal{H}$

Definition 2 *Given $d \in \mathbb{N}$, the Lie algebra $\mathbf{g_d}$ is the matrix algebra consisting of all $d \times d$ upper triangular matrices. This algebra is solvable [19] and of dimension $\frac{d(d+1)}{2}$. Its vectors are expressed as,*

$$\mathbf{g_d}(y_{r,s}) = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1d} \\ 0 & y_{22} & \cdots & y_{2d} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & y_{dd} \end{pmatrix}, \qquad y_{rs} \in \mathbf{R}. \tag{1}$$

From here on, the basis is $\mathcal{B}_d = \{Y_{i,j} = \mathbf{g_d}(y_{r,s})\}_{1 \le i \le j \le d}$ of $\mathbf{g_d}$, where $y_{r,s}$ takes 1 if $(r,s) = (i,j)$ otherwise 0. The law with respect to the basis $\mathcal{B}_d$ for distinct i,j and k with $1 \le i < j < k \le d$ is,

$$\begin{aligned} &[Y_{i,j}, Y_{j,k}] = Y_{i,k}(Type\ 1),\ [Y_{i,i}, Y_{i,j}] = Y_{i,j}(Type\ 2), \\ &[Y_{i,j}, Y_{j,j}] = Y_{i,j}(Type\ 3). \end{aligned} \tag{2}$$

Definition 3 *A hypergraph on $X = \{x_1, x_2, \ldots, x_m\}$ is a family, $\mathcal{H} = (E_1, E_2, \ldots, E_n)$ of subsets of X such that $E_i \neq \emptyset$, $i = 1, 2, \ldots, n$ and $\bigcup_{i=1}^{n} E_i = X$. The elements $x_1, x_2, \ldots, x_m$, of X are called vertices, and the sets $E_1, E_2, \ldots, E_n$ are the edges of the hypergraph.*

Definition 4 *Let $\mathcal{H} = (X, E)$ be a hypergraph. A set $\mathcal{T} \subseteq X$ is called a transversal of $\mathcal{H}$ if $\mathcal{T} \cap E_i \neq \emptyset, \forall E_i \in E$.*

Definition 5 *A set $\mathcal{V}_{\mathcal{S}}$ is variable set that contains variables that vary over time τ or with some parameters. As time interval varies cardinality of $\mathcal{V}_{\mathcal{S}}$ fluctuates, depending on this, we have following types of $\mathcal{V}_{\mathcal{S}}$ for any instances of time τ_1 and τ_2 with $\tau_1 \neq \tau_2$.*

- *$\mathcal{V}_{\mathcal{S}}$ is said to be increasing variable set when $\tau_1 < \tau_2$ then $|\mathcal{V}_{\mathcal{S}}(\tau_1)| < |\mathcal{V}_{\mathcal{S}}(\tau_2)|$.*
- *$\mathcal{V}_{\mathcal{S}}$ is said to be decreasing variable set when $\tau_1 < \tau_2$ then $|\mathcal{V}_{\mathcal{S}}(\tau_1)| > |\mathcal{V}_{\mathcal{S}}(\tau_2)|$.*

2.1 Associating Combinatorial Structures with Lie Algebras

Given a d-dimensional Lie algebra $\mathbf{g}$ with basis $\mathcal{B} = \{c_i\}_{i=1}^{d}$, recall the method introduced in [20, 21] for associating a combinatorial structure with $\mathbf{g}$. If $[c_x, c_y] = \sum_{z=1}^{d} f_{x,y}^{z} c_z$, a combinatorial structure can be associated with $\mathbf{g}$ as follows:

a) Make vertex x for each $c_x \in \mathcal{B}$.

b) Given three vertices $x < y < z$, draw the full triangle xyz if and only if $(f_{x,y}^{z}, f_{y,z}^{x}, f_{x,z}^{y}) \neq (0, 0, 0)$. Edges xy, yz and xz have weight $f_{x,y}^{z}$, $f_{y,z}^{x}$ and $f_{x,z}^{y}$ respectively.

   b1) Use a discontinuous line (named a ghost edge) for edges with weight zero.

   b2) If two triangles xyz and xyl satisfy that $f_{x,y}^{z} = f_{x,y}^{l}$, draw only one edge between vertices x and q shared by the two triangles.

c) Given two vertices $x < y$, draw a directed edge from y to x if $f_{x,y}^{x} \neq 0$ or a directed edge from x to y if $f_{x,y}^{y} \neq 0$.

For upper triangular matrix Lie algebra $\mathbf{g}_d$, Ceballos in [22], have defined an order for associating each vertex with a vector from the basis $\mathcal{B}_d$ of $\mathbf{g}_d$. More concretely, the order is the one of the elements of each row of matrix $g_d(y_{r,s})$ in Eq. (1) is given as, $\{Y_{1,1}, Y_{1,2}, \ldots, Y_{1,d}\}$ with $\{c_1, c_2, \ldots, c_d\}$, $\{Y_{2,2}, Y_{2,3}, \ldots, Y_{2,d}\}$ with $\{c_{d+1}, c_{d+2}, \ldots, c_{2d-1}\}$, …, $\{Y_{d,d}\}$ with $\{c_{\frac{d(d+1)}{2}}\}$.

3. SYSTEM MODEL

OppNets can be modeled composing of p number of nodes, $(o_1, o_2, o_3, \ldots, o_p)$ and can cooperative via wireless technologies. Assuming the data transfer between nodes occurs through the intermediate node by enabling the connection (C_{o_a}, C_{o_b}) with sufficient energy and buffer to store. Communications amidst the nodes are contact opportunities, and these opportunities are considered to be auto- nomous. Contact duration is mentioned below in Eq. (3).

$$CD = \frac{D_{rt}}{(D_r + D_t)/2} \tag{3}$$

where $D_{rt} = \sum_{i=1}^{L_p} d_{rt}(i) = \sum_{i=1}^{L_p} (d_{rt_{end}}(i) - d_{rt_{start}}(i))$, D_{rt} is a total contact duration between relay/source (r) and the destination node (t) along the length of the path L_p, $d_{rt_{start}}(C_{o_a}, C_{o_b})$ and $d_{rt_{end}}(C_{o_a}, C_{o_b})$ are starting and ending connection time between the relay node and destination node respectively. D_s and D_t is a total duration of time of relay node and destination node that is in contact with all other nodes respectively. The connectivity of nodes is assumed as symmetric.

3.1 Deliberated Routing Objectives

The method proposed in this paper considers the Hop Count (HC), Average Delivery Delay (AD), and Delivery ratio (DR) as routing decision objectives, where the HC, calculates the number of nodes that are needed to transfer packets from source to destination and is given as, $HC = \sum_{i,j \in L} e_{ij} - 1$ where L is the set of connections or links between nodes in the network and e_{ij} takes 1 if link (i, j) is used in the path 0 otherwise.

Average Delivery Delay is a time taken to transfer a packet from source to destination. Here, end-to-end delay, taken in each hop is considered for a packet transfer. The Average Delivery Delay is, $AD = \frac{1}{TPT} \sum_{i=1}^{NH} D_i$ where $D_i = ET_i - ST_i$ AD $\rightarrow$ Average Delivery Delay, NH $\rightarrow$ Number of Hops, TPT $\rightarrow$ Total number of packets transmitted successfully, D_i $\rightarrow$ Delay at node i, ST_i $\rightarrow$ Starting time at which the packet enters hop i and ET_i $\rightarrow$ Time at the packet delivered at hop i.

The delivery ratio is a significant measure of the number of packets received by the number of packets sent. The average delivery ratio is, $ADR = \frac{1}{NPS} \sum_{i=1}^{p} NPR_i$ where $p \rightarrow$ Number of nodes, $NPR_i \rightarrow$ Number of packets received at hop node i, $NPS \rightarrow$ Number of packets sent from source.

3.2 Multi-Objective Function

Multi-objective optimization encompasses more than one objective function subject to a variety of constraints. Objective functions to be minimized or maximized are,

$$maximize/minimize\, X_a(s), a = 1, \ldots, M \, subject\, to\, y_j(s) \geq 0, j = 1, \ldots, J$$
$$z_k(s) = 0, k = 1, \ldots, K; s_i^l \leq s_i \leq s_i^u, i = 1, \ldots, n$$

where M is number of objective functions with $M \geq 2$, $g_j(s)$ and $h_k(s)$ represents constraint functions, J inequalities and K equality constraints. Here, s is a solution of n decision variables with s_i^l and s_i^u as lower and upper bound. A set of solutions found for the above optimization problem is known as pareto optimal sets.

This paper proposes the following multi-objective problem for cluster-based routing technique,

$$\begin{array}{ll}
minimize & HC \\
minimize & AD \\
maximize & ADR \\
subject\ to & deg(o_i) \geq 1 \qquad i = 1, 2, \ldots, p \\
 & CD(o_i, o_j) \geq \epsilon \qquad i, j = 1, 2, \ldots, p \\
 & LC = 3\ and\ deg(o_i), CD(o_i, o_j), LC \geq 0.
\end{array}$$

Classic scalarization methods based on mathematical programming designed for MOO include linear weighted methods. By pre-multiplying each performance metric with a weight, the linear weighted sum process scalarises several performance metrics into a single-objective function. Now, our problem becomes,

$$\psi(HC, AD, ADR) = W_1\psi_1 + W_2\psi_2 + W_3\psi_3$$

where $\psi_1 \propto \psi(HC) = HC$, $\psi_2 \propto \psi(AD) = AD$ and $\psi_3 \propto \frac{1}{\psi(ADR)} = \frac{1}{ADR}$ are the functions associated with HC, AD and ADR respectively, weights are equal and sum of the weights is equal to 1, that is, $\sum_{i=1}^{3} W_i = 1$. Here, we obtain the pareto optimal sets using the state-of-the-art variable hypergraph.

4. PROPOSED METHODOLOGY

In this section, the former part introduces the variable hypergraph, and the construction of variable hypergraph using Lie commutators, while in latter propose a novel unanimous approach of clustering and routing using variable hypergraph. In the clustering phase, nodes of the network are clustered using variable hyperedges and followed by fixing nodes inside Upper Triangular Matrix (UTM) and finally conferred a novel routing protocol with Lie commutators.

4.1 Variable Hypergraph Construction using Lie Commutator

Definition 6 *A Variable hypergraph is a pair $\mathcal{V_H} = (\mathcal{V_X}, \mathcal{V_E})$ with both $\mathcal{V_X}$ and $\mathcal{V_E}$ are variable sets where $\mathcal{V_X} = \{o_i \mid i = 1, 2, \dots, p\}$ and $\mathcal{V_E} = \{\alpha_i \mid i = 1, 2, \dots, q\}$ $\alpha_i \neq \emptyset$ and $\bigcup_{i=1}^{q} \alpha_i = \mathcal{V_X}$. The elements $o_1, o_2, \dots, o_p$ of $\mathcal{V_X}$ are called vertices, and the sets $\alpha_1, \alpha_2, \dots, \alpha_q$ are the variable hyperedges.*

Definition 7 *If $\mathcal{V_H} = (\mathcal{V_X}, \mathcal{V_E})$ is a variable hypergraph. A set $\mathcal{V_{HT}} \subseteq \mathcal{V_X}$ is called a variable transversal of $\mathcal{V_H}$ if*

$$\mathcal{V_{HT}} \cap \alpha_i \neq \emptyset, \forall\, \alpha_i \in \mathcal{V_E}. \tag{4}$$

That is, variable hypergraph transversal is a variable set which changes with respect to some parameter like time and has non-empty intersection with every variable hyperedge of $\mathcal{V_H}$.

4.1.1 Construction of variable hypergraph

A variable hypergraph $\mathcal{V_H}$ construction for Lie algebra $\mathbf{g}_d$ for arbitrary $d \in \mathbb{N}$ is introduced here with the Lie commutators of Type 1 and Type 2 defined in equation 2, and by defining a new commutator (Type 4) with respect to the basis $\mathcal{B}_d$ for distinct i, j and k with $1 \leq i < j < k \leq d$.

$$[Y_{i,j}, Y_{j,k}] = Y_{i,k}(Type\ 1),\quad [Y_{i,i}, Y_{i,j}] = Y_{i,j}(Type\ 2)\ and$$
$$[[Y_{i,i}, Y_{i,j}], Y_{j,j}] = Y_{i,j}(Type\ 4).$$

The construction of $\mathcal{V_H}$ for $\mathbf{g}_d$ is as follows:

a) Construct a variable hyperedge α_i for the vertices corresponding to the diagonal elements, by Type 4 by which each diagonal element is connected with rest of the diagonal elements.

b) Given three vertices i, j and k make a hyperedge α_i containing i, j and k if and only if corresponding basis elements persuade Type 1 or Type 2 or Type 4.

Lemma 1 *If $\mathcal{V}_{\mathcal{H}}$ is associated with one of the elements of matrix Lie algebra $\mathbf{g}_d$ by Type 1 and Type 4, then the vertices corresponding to diagonal elements have degree d, and the remaining vertices have degree $d-1$.*

Proof: Let $\mathcal{V}_{\mathcal{H}}$ be associated with an element of $\mathbf{g}_d$ consisting of all $d \times d$ upper triangular matrices. The possibility of combining diagonal elements by Type 4 is $\binom{d}{2}$, and it is evident that each element occurs $d-1$ times, and by first part of the construction of $\mathcal{V}_{\mathcal{H}}$, the α_i contains all the diagonal elements in $d \times d$ matrices of $\mathbf{g}_d$. Hence the degree of vertices corresponding to diagonal elements is $d-1+1 = d$.

There are $\binom{d}{3}$ ways of aggregating non-diagonal elements of one of the $d \times d$ matrices of $\mathbf{g}_d$, in which each element occurs $d-2$ times, and by basis elements $Y_{i,i}$ of Type 4, vertices corresponding to every non-diagonal elements is already incident with a variable hyperedge, yields the desired result.

Theorem 2 *If $\mathcal{V}_{\mathcal{H}}$ is a variable hypergraph of n vertices associated with an element of Lie algebra of upper triangular matrix $\mathbf{g_d}$ with $d \geq 2$ then the number of variable hyperedges is,*

$$q = \begin{cases} \binom{d}{3} + 2 * \binom{d}{2} + 1, & \text{if } n = \frac{d(d+1)}{2} \\ \left(\binom{d}{3} - \left\{\sum\limits_{i=1}^{n_0}(d-(i+1))\right\}\right) + 2 * \left(\binom{d}{2} - n_0\right) + 1, & \text{if } \frac{d(d-1)}{2} \leq n \leq \frac{d(d+1)}{2} \end{cases} \tag{5}$$

Proof: If $n = \frac{d(d+1)}{2}$, according to the method expounded in Section 3, $\mathcal{V}_{\mathcal{H}}$ associated with an element of $\mathbf{g}_d$ that has $\frac{d(d+1)}{2}$ vertices. In virtue of Lemma 1, d vertices have degree d, and remaining vertices have degree $d-1$ then the number of variable hyperedges with respect to Types 1 and 4 commutator is given by

$$\frac{d(d-1)}{3} + \frac{d*1}{d} + \frac{\left(\frac{d(d+1)}{2} - d\right) * (d-2)}{3} + \frac{\left(\frac{d(d+1)}{2} - d\right) * 1}{3}$$
$$= \frac{d(d-1)}{3}(1 + \frac{1}{2}) + 1 + \frac{d(d-1)(d-2)}{6} = \binom{d}{2} + \binom{d}{3} + 1.$$

By encompassing the Type 2 commutator, we have $\binom{d}{3}$ hyperedges therefore, the number of variable hyperedges is $\binom{d}{2} + 2 * \binom{d}{3} + 1$.

Now, to prove for the case $\frac{d(d-1)}{2} < n < \frac{d(d+1)}{2}$. Let us suppose that $n_0 = \frac{d(d+1)}{2} - n$. If $n_0 = 1$ then $y_{1d} = 0$, $n_0 = 2$ then $y_{1d} = 0$ and $y_{2d} = 0, \ldots, n_0 = d-1$ then $y_{1d} = 0,\ y_{2d} = 0, \ldots, y_{(d-1),d} = 0$. It remains that there are $\binom{d}{2}$ ways of fusing the Type 4 and $\binom{d}{3}$ ways of Type 1.

If $y_{1d} = 0$ then $(d-2)$ variable hyperedges does not exist. Similarly for $y_{2d} = 0, \ldots, y_{(d-2),d} = 0,\ y_{(d-1),d} = 0$ then $(d-3), \ldots, 1, 1$ variable hyperedges are non exant respectively. Now, this totally accounts $\sum_{i=1}^{n_0}(d-(i+1))$ which proceeds to $\binom{d}{3} - \sum_{i=1}^{n_0}(d-(i+1))$. $\binom{d}{2}$ variable hyperedges gets diminished depending on n_0. If $n_0 = 1$ then single variable hyperedge is less. Hence it equals $\binom{d}{2} - n_0$ and by the first step of construction in $\mathcal{V}_{\mathcal{H}}$, all diagonal elements are made as a single hyperedge. Therefore, the number of variable hyperedges is $\left(\binom{d}{3} - \left\{\sum_{i=1}^{n_0}(d-(i+1))\right\}\right) + \left(\binom{d}{2} - n_0\right) + 1.$

Now, including the Type 2 commutator, we have the similar argument with respect to the case $n = d(d+1)/2$, the number of variable hyperedges is, $\left(\binom{d}{3} - \left\{\sum_{i=1}^{n_0}(d-(i+1))\right\}\right) + 2 * \left(\binom{d}{2} - n_0\right) + 1.$

Corollary 2.1 *The number of variable hyperedges in Case 1 of Eq. (5) becomes upper bound if* $n_o > \frac{d(d+1)}{2} - s$.

4.2 Proposed Clustering Scheme: VHC

The proposed variable hypergraph clustering (VHC) technique partitions the nodes in the opportunistic network as clusters. This clustering is made by constructing variable hyperedges with respect to time parameter τ, which quantifies the structural prominence of a node within the network. Algorithm 1 forges a clusters (α_i) from the network topology. Construction of variable hyperedges (clusters) does not exceed Eq. (5) in Theorem 2.

4.2.1 Cluster head election

Variable transversal $(\mathcal{V}_{\mathcal{HT}})$ of variable hypergraph $(\mathcal{V}_{\mathcal{H}})$ defined in Section 4.1 is effectual due to the time parameter. Cluster heads are determined using $\mathcal{V}_{\mathcal{HT}}$ and it is presented in Algorithm 2 with $\mathcal{V}_{\mathcal{HT}}$ the set of cluster heads.

4.3 Fixing Nodes Position Inside an Upper Triangular Matrix

The resulting variable hyperedges $\mathcal{V}_{\mathcal{E}} = \{\alpha_i \mid i = 1, 2, \ldots, w\}$ are now transformed into a component of the upper triangular routing matrix which is an element of upper triangular matrix Lie algebra through Type 1, Type 2 and Type 4 commutators. By constructing an upper triangular matrix, the best relay can be found using the Lie commutator with significantly less time to forward data from source to destination. Algorithm 3 fix the position inside the matrix of every node in the network.

4.4 Novel Routing Technique using Lie Commutators

Several routing protocols and limited cluster-based routing protocols have been designed for OppNets so far to ensure the shortest path between source and destination. In cluster-based routing, clustering and routing are handled as

Algorithm 1: VHC

INPUT: Network
OUTPUT: $\mathcal{V}_{\mathcal{E}} = \{\alpha_i \mid i = 1, 2, \ldots, w\}$ with $w \leq q$
Calculate degree of each node.
Find a vertex u with highest degree d from a vertex set $\mathcal{V}_{\mathcal{X}}$. (If more than one vertices say, $o_i, o_j, \ldots$ have highest degree then, if one of o_i's adjacent vertex has degree 2 then take the vertex o_i as u otherwise choose an arbitrary vertex)
Construct α_i containing all the vertices that are adjacent to u.
$L = \mathcal{V}_{\mathcal{X}}$- V($\alpha_i$)
for l_i *in* L **do**
  if $l_i \in L$ *is adjacent to any two vertices in* α_i **then**
    Construct α_x containing l_i with two vertices say o_a and o_b that are adjacent; x=x+1.
  if $l_i \in L$ *is adjacent to only one vertex in* α_i **then**
    Construct α_y containing l_i with its single adjacent vertex o_c; y=y+1.
  if *every* $l_i \in L$ *is not adjacent with any vertex in* α_i **then**
    if $l_j \in L$ *is adjacent to* l_i *in* α_p *and* $l_k \in L$ *is adjacent to* l_j **then**
      Construct α_r containing l_i, l_j and l_k; r=r+1.
    if $l_m \in L$ *is adjacent to* l_j *or* $l_k \in \alpha_r$ *and adjacent to* $l_n \in L$ **then**
      Construct α_s containing l_m, l_j and l_n or l_m, l_k and l_n; s=s+1.
    if $l_j \in L$ *is adjacent to* l_i *in* α_q *and* $l_k \in L$ *is adjacent to* l_j **then**
      Construct α_t containing l_i, l_j and l_k; t=t+1.
    if $l_m \in L$ *is adjacent to* l_j *or* $l_k \in \alpha_t$ *and adjacent to* $l_n \in L$ **then**
      Construct α_u containing l_m, l_j and l_n or l_m, l_k and l_n; u=u+1.

Algorithm 2: CH selection using $\mathcal{V}_{\mathcal{HT}}$

INPUT: $\mathcal{V}_{\mathcal{E}}$
OUTPUT: CHs
$\mathcal{V}_{\mathcal{HT}} = \emptyset$
for α_i *in* $\mathcal{V}_{\mathcal{E}}$ **do**
  select a vertex o_i from α_i and add o_i to $\mathcal{V}_{\mathcal{HT}}$ such that $\mathcal{V}_{\mathcal{HT}} \cap \alpha_i \neq \emptyset$ (By Equation 4)

Algorithm 3: $\mathcal{V}_{\mathcal{E}}$ to Upper Triangular Routing Matrix (UTRM)

INPUT: $\mathcal{V}_{\mathcal{E}} = \{\alpha_i \mid i = 1, 2, \ldots, w\}$ from Algorithm 1 where $w \leq q$ by Corollary 2.1.
OUTPUT: UTRM dimension=$max_i|\alpha_i|$
Create $dimension \times dimension$ UTRM matrix.
for α_i *in* $\mathcal{V}_{\mathcal{E}}$ **do**
  if $|\alpha_i|$==*2* **then**
    mate2.append(α_i)
  if $|\alpha_i|$==*3* **then**
    mate3.append(α_i)
  if $|\alpha_i|$==*dimension* **then**
    mate1.append(α_i)
s_i, s_j, s_k, $s_l \in \{1,\ 2\ \ldots\ dimension\}$
Procedure MATPOSITIONFIX1(mate1,mate2,mate3)
Procedure MATPOSITIONFIX2(mate1,mate2,mate3)
Procedure MATPOSITIONFIX3(mate1,mate2,mate3)

Algorithm 4: MATPOSITIONFIX1

for α_i *in mate3* **do**
  for $\alpha_j \neq \alpha_i$ *in mate3* **do**
    if $|\alpha_i \cap mate1| = 0$ *and* $|\alpha_j \cap mate1| = 0$ *and* $|\alpha_i \cap \alpha_j| \neq 0$ **then**
      for α_k *in mate3* **do**
        if $|\alpha_k \cap mate1| \neq 0$ *and* $|\alpha_i \cap \alpha_j| \neq$ **then**
          Place elements of V(α_k) in $s_i s_i$, $s_j s_j$ and $s_i s_j$ position.
          Place elements of V(α_j) in $s_i s_j$, $s_j s_k$ and $s_i s_k$ position.
          Place elements of V(α_i) in $s_j s_k$, $s_k s_l$ and $s_j s_l$ position.
          for m *in mate2* **do**
            if $|\alpha_k \cap m| \neq 0$ **then**
              Place elements in V(α_m) in ($s_i s_i$ *and* $s_i(s_i + 1)$) or ($s_j s_j$ *and* $s_j(s_j + 1)$) position.
      for α_m *in mate2* **do**
        if $|\alpha_j \cap m| \neq 0$ **then**
          Place elements of V(α_m) in $s_i s_i$ and $s_i(s_i + 1)$ position.
          Place elements of V(α_j) in $s_i(s_i + 1)$, $(s_i + 1)s_2$ *and* $s_i s_j$ position.
          Place elements of V(α_i) in $s_i s_j$, $s_j s_k$ *and* $s_i s_k$ position.

Algorithm 5: MATPOSITIONFIX2

for α_i *in mate3* **do**
  for $\alpha_j \neq \alpha_i$ *in mate3* **do**
    if $|\alpha_i \cap mate1|=0$ *and* $|\alpha_j \cap mate1| \neq 0$ *and* $|\alpha_i \cap \alpha_j| \neq 0$ **then**
      Place elements of V(α_i) in $s_i s_i$, $s_j s_j$ *and* $s_i s_j$ position.
      Place elements of V(α_j) in $s_i s_j$, $s_j s_k$ *and* $s_i s_k$ position.
      for α_k *in mate2* **do**
        if $|\alpha_k \cap \alpha_j| \neq 0$ **then**
          Place element of V(α_j) in ($s_i s_i$ *and* $s_i(s_i + 1)$) or ($s_j s_j$ *and* $s_j(s_j + 1)$) position.
  for α_k *in mate2* **do**
    if $|\alpha_i \cap \alpha_k| \neq 0$ **then**
      Place elements of V(α_k) in $s_i s_i$ *and* $s_i(s_i + 1)$ position.
      Place elements of V(α_i) in $s_i(s_i + 1)$, $(s_i + 1)s_j$ *and* $s_i s_j$ position.

Algorithm 6: MATPOSITIONFIX3

for α_i *in mate3* **do**
  if $|\alpha_i \cap mate1|$=*0* **then**
    for α_i *in mate2* **do**
      a2.append($\alpha_i \cap \alpha_j$)
      a3.append(α_j)
      if $|a2|$=*2* **then**
        Place elements of V(a3) in $s_i s_i$, $s_i(s_i+1)$ *and* $s_j s_j$, $s_j(s_j+1)$ position where i+1=j.
        Place elements of V(α_i) in $s_i(s_i+1)$, $s_j(s_j+1)$ *and* $s_i(s_j+1)$ position.
Find the remaining elements of mate1 and fix it in $s_i s_i$, $s_j s_j$, $s_k s_k, \ldots$ position.
for α_i *in mate3* **do**
  $u1 = \alpha_i \cap mate1$
  Find position of elements in u1 say $s_i s_i$ *and* $s_j s_j$ and place the remaining element in $s_i s_j$ position.
  Find the hyperedges in mate2 intersection with i.
  Find the matrix position of intersected element say $s_i s_i$ place it in $s_i(s_i+1)$ position.
for α_i *in mate2* **do**
  Find the matrix position of intersected element say $s_i s_i$ place it in $s_i(s_i+1)$ position.

Algorithm 7: Upper Triangular Routing Matrix(UTRM) Algorithm

INPUT: Matrix Position of source and destination (say [i, j] and [k, l]) and UTRM from Algorithm 3.
OUTPUT: Optimized Route
sourcelen=cardinality of variable hyperedge containing source node.
destinationlen=cardinality of variable hyperedge containing destination node.
if *sourcelen=destinationlen=dimension* **then**
  Direct Path ($[i,\ j] \rightarrow [k,\ l]$)
else
  if *sourcelen=2 and 3* **then**
    Procedure PATHDETECTION1
  if *sourcelen=2* **then**
    Procedure PATHDETECTION2
  if *sourcelen=3* **then**
    Procedure PATHDETECTION3

Algorithm 8: PATHDETECTION1

Repeat the Algorithm 9.
if *destinationlen=3* **then**
  if *(i=a and j=b) or (i=δ or j=d)* **then**
    Path $[i,\ j] \rightarrow [k,\ l]$

different phases, but our approach unifies clustering and routing phases. It identifies the route in such a way that it can reduce data redundancy, selection of best next hop and also handle well in a dynamic network. By matrix principle, the nodes further reduce the overhead routing, which guarantees the free transfer of packets, lowers the packet loss and defend against unnecessary latency in the network. In the proposed routing strategy, the best relay node to forward the data packets is selected according to Lie commutators of Types 1, 2 and 4 in Section 4.1. Algorithm 7 presents the routing scheme for OppNets.

Algorithm 9: PATHDETECTION2

Let the matrix position of nodes in a hyperedge containing the source node be [i, j] and [β, γ].
if *(destinationlen=2) or (destinationlen=2 and dimension)* **then**
    if $\beta=\gamma$ *and* $k=i$ **then**
        Path $[i,\ j] \rightarrow [\beta,\ \gamma]$
    else
        Path $[i,\ j] \rightarrow [\beta,\ \gamma] \rightarrow [j,\ j]$
else if *destinationlen=2 and 3* **then**
    Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [k,\ k] \rightarrow [k,\ l]$
else if *destinationlen=3* **then**
    Let the matrix position of nodes in a hyperedge containing the destination node be [a, b], [δ, d] and [k, l].
    if $a{=}{=}b$ *and* $\delta{=}{=}d$ *and* $a!{=}\delta$ **then**
        Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [a,\ b] \rightarrow [k,\ l]$
    else if $\mid$ *Hyperedge containing* $[a,\ b]\ \mid=2$ **then**
        Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [a,\ a] \rightarrow [a,\ b] \rightarrow [k,\ l]$
    else if $\mid$ *Hyperedge containing* $[\delta,\ d]\ \mid=2$ **then**
        Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [\delta,\ \delta] \rightarrow [\delta,\ d] \rightarrow [k,\ l]$
    else if $\mid$ *Hyperedge containing* $[a,\ b]\ \mid=3$ **then**
        Let the matrix position of nodes in a hyperedge containing the [a, b] be [a, b], [e, f] and [g, h].
        if $\mid$ *Hyperedge containing* $[e, f]\ \mid=2$ **then**
            Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [e,\ e] \rightarrow [e,\ f] \rightarrow [a,\ b] \rightarrow [k,\ l]$
        else if $\mid$ *Hyperedge containing* $[g,\ h]\ \mid=2$ **then**
            Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [g.\ g] \rightarrow [g,\ h] \rightarrow [a,\ b] \rightarrow [k,\ l]$
    else if $\mid$ *Hyperedge containing* $[\delta,\ d]\ \mid=3$ **then**
        Let the matrix position of nodes in a hyperedge containing the [δ, d] be [δ, d], [e, f] and [g, h].
        if $\mid$ *Hyperedge containing* $[e,\ f]\ \mid=2$ **then**
            Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [e,\ e] \rightarrow [e,\ f] \rightarrow [a,\ b] \rightarrow [k,\ l]$
        else if $\mid$ *Hyperedge containing* $[g,\ h]\ \mid=2$ **then**
            Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [g,\ g] \rightarrow [g,\ h] \rightarrow [a.\ b] \rightarrow [k,\ l]$

Example 1. Consider the network scenario in Fig. 1 (a) for which Algorithm 1 generates the corresponding variable hypergraph given in Fig. 1 (b) and Algorithm 3 produces an element of upper triangular matrix Lie algebra whose dimension is 7×7.

$$UTRM = \begin{bmatrix} v8 & v11 & v12 & 0 & 0 & 0 & 0 \\ 0 & v7 & v10 & 0 & 0 & 0 & 0 \\ 0 & 0 & v3 & v2 & v1 & 0 & 0 \\ 0 & 0 & 0 & v4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & v6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v5 \end{bmatrix}$$

Algorithm 10: PATHDETECTION3

Let the matrix position of nodes in a hyperedge containing the source node be [i, j], [β, γ] and [r, s].
if *destinationlen=dimension or destinationlen=3 and dimension* **then**
  if *(i==k and j==r) or (j==k and i==β)* **then**
    Path $[i,\ j] \rightarrow [k,\ l]$
  else if *| Hyperedge containing $[\beta,\ \gamma]$ |=2 and β=k* **then**
    Path $[i,\ j] \rightarrow [\beta,\ \gamma] \rightarrow [k,\ l]$
  else if *| Hyperedge containing $[r,\ s]$ |=2 and r=k* **then**
    Path $[i,\ j] \rightarrow [r,\ s] \rightarrow kl$
  else if *| Hyperedge containing $[\beta,\ \gamma]$ |=2* **then**
    Path $[i,\ j] \rightarrow [\beta,\ \gamma] \rightarrow [\beta,\ \beta] \rightarrow [k,\ l]$
  else if *| Hyperedge containing $[r,\ s]$ |=2* **then**
    Path $[i,\ j] \rightarrow [r,\ s] \rightarrow [r,\ r] \rightarrow [k,\ l]$
else if *destinationlen=3* **then**
  Let the matrix position of nodes in a hyperedge containing the destination node be [a, b], [δ, d] and [k, l]. **if** *(β=k and γ=l) or (r=k and γ=l)* **then**
    $[i,\ j] \rightarrow [k,\ l]$
  if *β=γ and r=s and β!=r and a=b and δ=d and a!=δ* **then**
    if *β!=a and β!=δ and r!=a and r!=δ* **then**
      Path $[i,\ j] \rightarrow [\beta,\ \beta] \rightarrow [k,\ k] \rightarrow [k,\ l]$
    else if *| Hyperedge containing $[a,\ b]$ |=2* **then**
      Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [a,\ a] \rightarrow [a,\ b] \rightarrow [k,\ l]$
    else if *| Hyperedge containing $[\delta,\ d]$ |=2* **then**
      Path $[i,\ j] \rightarrow [i,\ i] \rightarrow [\delta,\ \delta] \rightarrow [\delta,\ d] \rightarrow [k,\ l]$
    else
      Path $[i,\ j] \rightarrow [\beta,\ \beta] \rightarrow [k,\ l]$

Let us suppose that source = $v1$ and destination = $v12$ then Table 2 shows the execution of the Algorithm 7 to get the path. Then the path from source to destination is $[3,5] \rightarrow [3,3] \rightarrow [1,1] \rightarrow [1,2] \rightarrow [1,3]$ that is $v1 \rightarrow v3 \rightarrow v8 \rightarrow v11 \rightarrow v12$.

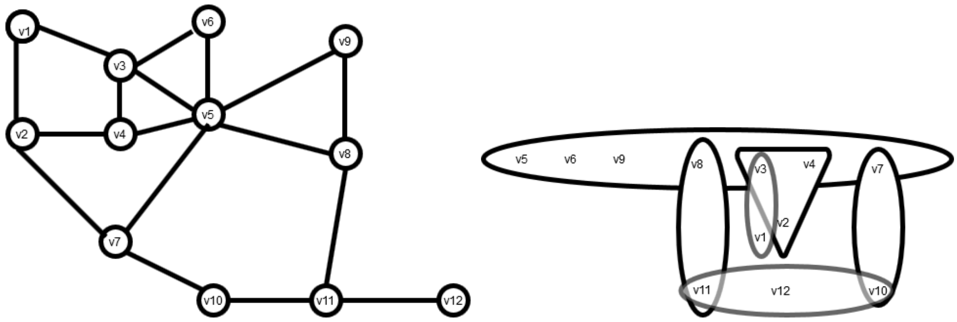
Table 2. Steps involved in execution of Algorithm 7.

	Source matrix position=[3,5]	Destination matrix position=[1,3]
Step 1	$[3,5] \rightarrow [3,3]$	[3, 3]=v3
Step 2	$[3,3] \rightarrow [1,1]$	[1, 1]=v8
Step 3	$[1,1] \rightarrow [1,2]$	[1, 2]=v11
Step 4	$[1,2] \rightarrow [1,3]$	[1, 3]=v12

5. RESULTS AND DISCUSSION

This section presents the simulation settings and performance analysis of the proposed technique. Python based simulator SimPy [23] is choosen for simulating our proposed work in Google Colaboratory. The performance is evaluated with Average Delivery Delay and Delivery Ratio as defined in Section 3.1.

In order to conduct, an informed design of network it is necessary to examine the frequency and duration of communications between human-carried communicating device. So, we have choosen the real connectivity trace from Crawdad



(a) Network topology. (b) Variable hypergraph for the network.

Fig. 1. $\mathcal{VH}$ for the network.

[24], by tiny handheld wireless radio (iMotes) tools that were distributed to various people. Table 3 summarizes the characteristic of the data set.

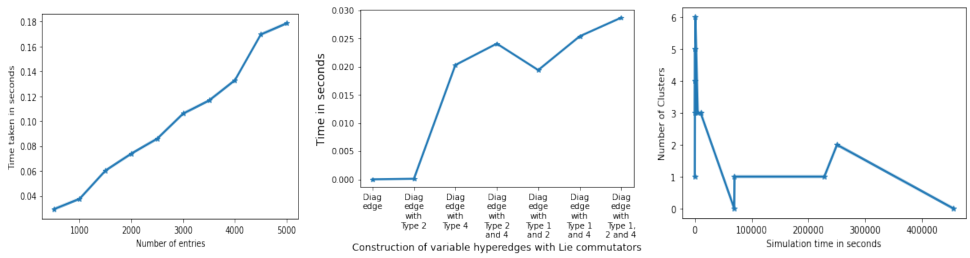
Table 3. Characteristic of dataset [24].

User Population	Cambridge
Device	iMote
Network type	Bluetooth
Duration(Seconds)	455609
Number of contacts	4228
Total number of entries	8456
Devices participated	12
Number of external contacts	2503

Fig. 2 (a) illustrates the time taken for the hypergraph construction with respect to the number of entries in [24]. In particular, the time for construction of an upper triangular matrix with respect to Lie commutators is depicted in Fig. 2 (b). Fig. 2 (c) presents the changes in the number of clusters against simulation time.

Fig. 3 (a) depicts the total elapsed time for the construction of variable hyperedges and fixing it inside an upper triangular matrix. Our work achieves a significant delivery ratio in routing packets from source to destination which is presented in Fig. 3 (b). Fig. 3 (c) presents the Average Delivery Delay with respect to simulation time.

A simulation scenario of 10 nodes have been setup for comparing with existing methods like Epidemic Routing (ER) [13], Contact-duration based ER (CDER) and Opportunistic Network Coding Routing (ONCR) [25]. It is shown from Fig. 4 (a), delivery ratio of the proposed method outperforms the ER, CDER and ONCR. From Fig. 4 (b) it is apparent that the delay in routing of the proposed method is much lower than that of the others, because of variable hypergraph construction with its position in a matrix for different instances of seconds. Our technique is effective to route OppNets in identifying neighbours by preserving the node position in the matrix for every second as the network shifts with minimal time.

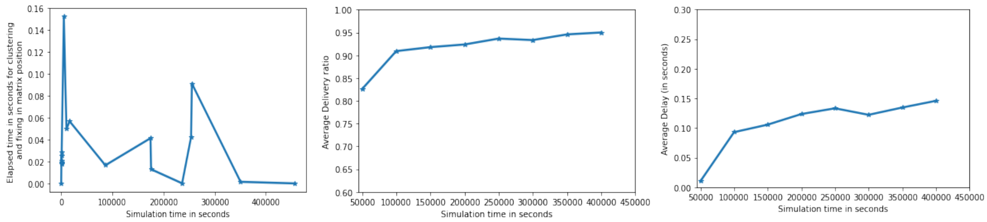


(a) Elapsed time for $\mathcal{V}_{\mathcal{HT}}$ vs. number of entries.

(b) Duration for construction of UTRM.

(c) Simulation time vs. number of clusters.

Fig. 2. Output of simulation.



(a) Simulation time vs. elapsed time.

(b) Simulation time vs. delivery ratio.

(c) Simulation time vs. average delivery delay.

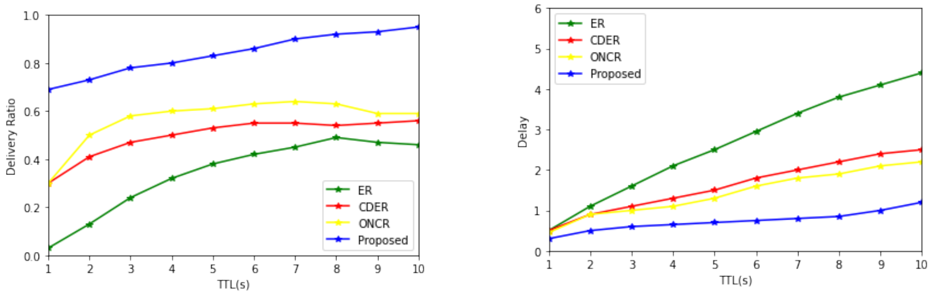
Fig. 3. Output of simulation.

6. CONCLUSION

In this paper, a novel routing protocol for OppNets has been proposed by a novel construction of variable hypergraph by the combination of Lie commutator. Variable hyperedges of variable hypergraphs are formed based on the connectivity of nodes which are clusters. The unification of clustering and routing is done by a significant theory called Lie algebra of upper triangular matrix by fixing nodes in the upper triangular matrix. Later UTRM protocol is proposed based on the position of every node in the upper triangular matrix where next-hop selection uses Lie commutators. It is shown that multi objective optimization in terms of delivery ratio, average delivery delay and hop count using variable hypergraph is the pareto sets which results in a higher delivery ratio with significantly less average delivery delay and hop count in very less elapsed time.

ACKNOWLEDGEMENT

The first author wishes to express sincere thanks to the INSPIRE fellowship (DST/INSPIRE Fellowship/2019/IF190271) for their financial support. The authors thank the Department of Science and Technology – Fund for improvement of S & T Infrastructure in Universities and Higher Educational Institutions, Government of India (SR/FST/MSI -107/2015). We would like to thank reviewers for their comments to improve the quality of the paper.



(a) Delivery ratio versus TTL. (b) Delay vs. TTL.

Fig. 4. Performance comparison.

REFERENCES

1. M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *IEEE Communications Magazine*, Vol. 48, 2010, pp. 126-139.
2. M. Conti and S. Giordano, "Mobile ad hoc networking: milestones, challenges, and new research directions," *IEEE Communications Magazine*, Vol. 52, 2014, pp. 85-96.
3. C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: a history based routing protocol for opportunistic networks," in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2007, pp. 1-12.
4. N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Engineering*, Vol. 5, 2018, p. 1502242.
5. M. Barr, C. Mclarty, and C. Wells, "Variable set theory," Technical Report No. (77, 400), Department of Mathematics and Statistics, McGill University, 1985.
6. K. Akimoto, T. Hiraoka, K. Sadamasa, and M. Niepert, "Cross-sentence n-ary relation extraction using lower-arity universal schemas," in *Proceedings of Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 6226-6232.
7. N. Yu, Y.-L. Gao, J.-X. Liu, J. Wang, and J. Shang, "Robust hypergraph regularized non-negative matrix factorization for sample clustering and feature selection in multi-view gene expression data," *Human Genomics*, Vol. 13, 2019, p. 46.
8. J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Transactions on Image Processing*, Vol. 21, 2012, pp. 3262-3272.
9. S. Venkatraman, G. Rajaram, and K. Krithivasan, "Unimodular hypergraph for dna sequencing: A polynomial time algorithm," in *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 2018, pp. 1-8.

10. B. Bai, L. Wang, Z. Han, W. Chen, and T. Svensson, "Caching based socially-aware D2D communications in wireless content delivery networks: A hypergraph framework," *IEEE Wireless Communications*, Vol. 23, 2016, pp. 74-81.
11. Y. Xu, J. Liu, Y. Shen, J. Liu, X. Jiang, and T. Taleb, "Incentive jamming-based secure routing in decentralized internet of things," *IEEE Internet of Things Journal*, Vol. 8, 2020, pp. 3000-3013.
12. Y. Xu, J. Liu, Y. Shen, X. Jiang, Y. Ji, and N. Shiratori, "Qos-aware secure routing design for wireless networks with selfish jammers," *IEEE Transactions on Wireless Communications*, Vol. 20, 2021, pp. 4902-4916.
13. A. Vahdat, D. Becker *et al.*, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Department of Computer Science, Duke University, 2000.
14. 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*, 2005, pp. 252-259.
15. M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, Vol. 8, 2008, pp. 246-260.
16. S. K. Dhurandher, D. K. Sharma, I. Woungang, and S. Bhati, "Hbpr: history based prediction for routing in infrastructure-less opportunistic networks," in *Proceedings of IEEE 27th International Conference on Advanced Information Networking and Applications*, 2013, pp. 931-936.
17. 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*, Vol. 10, 2019, pp. 1289-1306.
18. K. Khalid, I. Woungang, S. K. Dhurandher, J. Singh, and L. Barolli, "A fuzzy-based check-and-spray geocast routing protocol for opportunistic networks," *Journal of High Speed Networks*, 2021, pp. 1-12.
19. A. Malcev, "On solvable lie algebras," *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, Vol. 9, 1945, pp. 329-356.
20. A. Carriazo, L. Fernandez, and J. Nunez, "Combinatorial structures associated with lie algebras of finite dimension," *Linear Algebra and its Applications*, Vol. 389, 2004, pp. 43-61.
21. L. M. Fernandez and L. Martn-Martnez, "Lie algebras associated with triangular configurations," *Linear Algebra and its Applications*, Vol. 407, 2005, pp. 43-63.
22. M. Ceballos, J. Nunez, and A. Tenorio, "Combinatorial structures and lie algebras of upper triangular matrices," *Applied Mathematics Letters*, Vol. 25, 2012, pp. 514-519.
23. "Simpy," https://simpy.readthedocs.ioen/latest.
24. D.-G. Akestoridis, "CRAWDAD dataset uoi/haggle (v. 2016-08-28): derived from cambridge/haggle (v. 2009-05-29)," https://crawdad.orguoi/haggle2016 0828, 2016.

25. J. Yao, C. Ma, P. Wu, G. Du, and Q. Yuan, "An opportunistic network coding routing for opportunistic networks," *International Journal of Parallel Programming*, Vol. 45, 2017, pp. 157-171.

Supriya Sridharan has obtained BS and MS degrees in Mathematics in 2017 and 2019, respectively, from SASTRA Deemed University, Thanjavur, India. Currently, she is a DST-INSPIRE Fellow pursuing doctoral program in the field of Lie algebra and hypergraph with its applications to routing problems in wireless networks. Her research interest includes routing in opportunistic networks, wireless sensor networks and vehicular ad hoc networks.

Swaminathan Venkataraman received the Ph.D. degree from SASTRA Deemed University, Thanjavur, India. He has been a Researcher in Pure and Applied Mathematics. He has published a number of articles in the areas near-rings, closure spaces, and their application, and hybridization of intelligent techniques with algebraic and hypergraph components. His research interest includes noncommutative algebraic structures, lattices, triple systems, closure spaces, hypergraph, and hybridization of intelligent techniques with applicable mathematics for science and engineering problems.

Somasundaram K. currently serves as a Professor in the Department of Mathematics, School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India. He completed his Ph.D. from Bharathiyar University, India and Post Doctoral Research from University of Turku, Finland. His areas of research include multilinear algebra, graph theory, graph theory algorithms in network on chip.