Wireless Personal Communications (2022) 122:2081–2112
https://doi.org/10.1007/s11277-021-08982-3

A 4-step Blockchain Equipped Approach to Energy Efficiency and Routing in Homing Pigeon Based Delay Tolerant Network

Priyanka Das[1] · **Tanmay De**[2]

Accepted: 6 August 2021 / Published online: 25 August 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

At a time when the Delay Tolerant Network (DTN) Research Group is making great strides in establishing the much anticipated Solar System Internet (SSI) in space, its terrestrial counterpart is not far behind. The Earth bound DTN has been dedicated to connecting remote, heterogeneous, fluctuating, mobile and undisciplined networks like sensor, military and disaster struck areas. The Homing Pigeon Based DTN (HoP-DTN), a variant of DTN, was specifically conceptualized to serve some of these networks whose characteristics are different from the general delay or disruption tolerant networks. HoP-DTN uses special message carrying nodes, pigeons, to proactively route messages around the network. As it works in areas like disaster response or military, energy conservation is imperative here. Till now providing a solution to this energy issue has been lacking in HoP-DTN. Also routing in HoP-DTN is a Traveling Salesman Problem. This paper proposes a 4-step mechanism to address both of these issues. In view of that some changes to the classical modeling of HoP-DTN has been done to accommodate the variations. The deployment area has been divided into zones with pigeons following a 4-step detailed course plan and scheduling strategy to achieve better routing and more energy efficiency. To further enhance energy efficiency we introduce a Blockchain inspired energy sharing scheme. Mathematical modeling and simulation results further affirm our proposition.

Keywords Routing · Energy drainage · Replication · Delivery probability · Buffer-time · Latency

✉ Priyanka Das
priyanka.das.2206@gmail.com

Tanmay De
tanmayd12@gmail.com

[1] Department of Computer Science and Engineering, NSHM Knowledge Campus Durgapur, Durgapur, India

[2] Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur, India

1 Introduction

Though globalization has increased, there still are places which remain disconnected most of the time. Networks like military, disaster rescue, sensor and rural need special protocols to combat problems such as remoteness, unavailability of continuous link, enormous message delay, unreliable carriers and more such. The Homing Pigeon Based DTN (HoP-DTN) [1, 2] is a variant of the classical DTN, which can address these problems. Each node in HoP-DTN is a cluster represented by a cluster head which fairly remain static. The responsibility of routing the messages henceforth lies on some special custodian nodes in the network which move around delivering messages. These special nodes are known as pigeons who collect messages from their dedicated home nodes, move around the network by reaching every destination for delivery and comes back to its home node when the job is done. As the node positions are almost static and only the pigeons move around to deliver messages, this setup is perfect to handle disaster response areas, military, sensor and rural networks. Pigeons could be drones or any device capable of data relegation, even humans carrying data like in the Wizzy Digital Courier project [3] in South Africa that transfers e-mail messages and Web searches on a USB key that is carried by a bicycle or motorcycle rider between schools. Networks like HoP-DTN are always energy hungry but any energy efficient solution to this domain remains unexplored (to the best of our knowledge) and any solution would be very beneficial in terms of sustainability and enhancing the overall performance. Quite separate from this, was the recent advent of a technology called Blockchain. Mainly used in commerce and more recently in cryptocurrencies, Blockchain is a mechanism of keeping a digital ledger of all the transactions made by the participants without the involvement of any central authority. Maintaining Blockchain is very energy draining for the participating nodes due to its high consumption of power while using the systems. Yet this novel concept of Blockchain can be used for energy efficiency. This paper aims to utilize this concept and provide a much needed solution to scheduling and routing in HoP-DTN in an energy efficient manner. The next section discusses some existing mechanisms that deal with the routing and energy efficiency related issues. Taking a cue from these, the paper goes on to state its motivation and objectives for this work. Section 5 elucidates the proposed approach. To validate our claims we present the comprehensive simulation analysis section after that and end the paper with the conclusion.

2 Background Check

This section is divided into the three main aspects and domains that this paper focuses on, namely routing in DTNs, energy efficiency in DTNs and the HoP-DTN.

2.1 Previous Work on Routing in DTNs

Routing of messages in DTN can be achieved in two ways, proactive and reactive. In the reactive approach the nodes itself are responsible for delivery and the purpose is achieved by their movement around the network; hence also known as random mobility model. Examples of such protocols are epidemic [4] and spray and wait [5]. In proactive approach special auxiliary nodes are employed to achieve the purpose of message delivery. This scheme is also known as controlled mobility as the movement of these auxiliary nodes can be controlled in favour of delivery. HoP-DTN comes under this category, with its pigeons as auxiliary nodes. Other such proactive mechanisms are Message Ferry and Data Mules. In Message Ferry Routing [6, 7], special nodes known as MF (Message Ferry) move around the network collecting and delivering messages throughout the time, a mechanism that is also known as 'push data'. These nodes are public in nature and do not have a proprietary relationship with any node in specific. As these are public, once a source node hands over its messages to them they cease to have any control over their fate (unlike HoP-DTN where messages are either in possession of source node's pigeons or the destination node). Data Mules [8, 9] incorporate MULEs (Mobile Ubiquitous LAN Extension) that are special intermediary nodes that collect and pass messages to and from base stations and other sensor nodes; this mechanism is referred to as 'pull data'. The HoP-DTN differs from Message Ferries [6] and Data Mules [8] in the fact that pigeons are private unlike the very public nature of the former two, hence more secure and suitable for sensitive networks like military and disaster response centers.

2.2 Previous Work on Energy Efficiency in DTNs

Some of the most prominent energy efficient proactive approaches use small battery powered special nodes called 'throwboxes' [10] which creates a new contact opportunity by acting as a router for nodes that pass by the same location but at a different time. In Power Saving Management [11] a node can be either in receiving/transmitting mode or in power saving management (PSM) mode in addition to using throwboxes. When in PSM mode, the nodes switch between sleep and search states periodically in an attempt to lower the overall energy consumption. The work in [12] discusses ways of server placement to reduce energy consumption. Some notable work in converting reactive approaches to be energy efficient are like the one by Yong Li et al [13] that proposes a continuous-time Markov framework to model the message dissemination in DTN in an energy-aware fashion. Denis Rodrigues-Silva et al [14] provide an energy impact analysis over some popular reactive-based DTN protocols.

2.3 More on HoP-DTN

From the literature present we deduce the fact that pigeons as auxiliary nodes are either employed in HoP-DTN [15] or in networks simply known as pigeon networks [16]. The main difference between the two is that in HoP-DTN the pigeons carry messages from their respective home nodes to prospective destination nodes and return back to their home node whereas in a pigeon network the pigeon sets out to collect messages destined for its home node across the network. Pigeon networks actually have one Headquarter (the home node) and a disaster area from which it collects messages. The pigeon travels to and from these locations, collecting messages at one end and dumping them at the headquarter. The other pigeon assisted network, Homing pigeon based DTN derives its name from the ancient messaging system of employing pigeons to deliver messages to various destinations and come back to its home. As of date, most of the research work in HoP-DTN dealt with scheduling of pigeons in order to find an optimality between reduced waiting time of messages and number of pigeons used by a node to do so. To broadly classify the mechanisms explored till date, there is the multiple pigeon and single pigeon theory [17, 18]. As the name suggests, in multiple pigeon model a node owns not one but a set of pigeons, hence messages do not have to wait for the return of the solo pigeon to its home, as is in the case of the single pigeon model. Primarily there are two ways to schedule these pigeons namely, the on demand and the storage based strategy [18]. In the on demand strategy, as soon as a demand is created for a message to get delivered a pigeon takes off with that message. The traditional storage based pigeons do bulk message transfers and wait until its buffer of predefined size is full. The paper [19] suggests further ways of scheduling the storage based pigeons like the exhaustive service, gated service and adaptive service schemes. The exhaustive service discipline, makes its pigeons pick up messages that arrive before the departure of the message carrier in the current trip whereas in gated service discipline, it forces the messages to wait until the next trip. The adaptive service tries to give an optimal solution by combining exhaustive and gated services according to specific situations. Priority Based scheduling (PriorityB) [20] is a bulk messaging scheme wherein n number of copies of a message is created if its priority is n, to fill up the pigeon buffer. Thus the waiting time of a message has an inversely proportional relationship to its priority. Threshold Triplet Incorporated scheduling (TTI) [21] uses three thresholds (t, T and k) to establish a scheme that provides an upper bound on the worst case waiting time that can be experienced by any message. This is the only approach which provides an upper bound on the waiting time of messages. As routing in HoP-DTN and pigeon networks is a Traveling Salesman Problem (TSP), researchers have ignored doing any special work regarding it. Like paper [1] used Ant Colony Optimization and [16] uses Shortest TSP due to its simplicity. We would be using the concept of Blockchain for energy efficiency in this paper. Though not designed for it, its concept can be used to govern energy sharing between participants that could be both nodes and pigeons (or any other auxiliary node). In fact experiments have been started recently (during the end of 2017) in Europe, by using Blockchain technology amongst homes and electric cars for sharing energy. But accommodating that concept in a DTN is very difficult. We would in this work adapt, set rules and create the pathway for Blockchain in HoP-DTN as it seems the way to go for energy efficiency.

3 Motivation

Most of the existing work in HoP-DTN deal with devising different scheduling strategies for the pigeons. As routing here is a Traveling Salesman Problem, no specific research has been done towards it. The area of energy efficiency in HoP-DTN has not yet been investigated (to the best of our knowledge) though it is of mighty importance to the special networks that it caters to, namely military, rural and disaster rescue. Most importantly projects are now being sanctioned especially in remote third world countries where large amount of data is being transferred between stationery centers via men on bike [3]. The HoP-DTN is perfect for such scenarios. This forms the motivation for this research work.

4 Objectives

The following are the objectives of this research work:

1. To design an energy efficient alternative to the existing HoP-DTN.
2. To bring an improvement in the TSP routing by utilizing and taking advantage of some features intrinsic to HoP-DTN.
3. To develop and appropriate a scheduling technique that can support the above two objectives.

5 Proposed Approach: Zonified HoP-DTN

This paper aims to provide a solution to both the problems of routing and energy efficiency in HoP-DTN by implementing a 4-step mechanism. In this 4-step course plan, we would modify the scheduling strategy, segregate the deployment area, apply Blockchain and introduce new rules in a bid to work around the TSP routing and make the network more energy efficient. Also the first initiative to energy saving is that we employ battery efficient inexpensive pigeons with fairly limited storage capacity. Devices that could use solar energy as a means to recharge would be favourable. The 4 step mechanism is as follows (Fig. 1).

5.1 Step 1: Pizza Boy Strategy

The pizza delivery boys, upon getting delivery orders, set out to deliver these to the respective destinations. Instead of randomly distributing the orders among themselves they share them based on the nearness of one order to another order and to which part of the town/city the destinations lay. Their strategy is a time as well as an energy efficient take on the delivery problem by ensuring that they do not have to travel more for the same set of orders. We would take on the same strategy adopted by the pizza delivery boys and apply that to the pigeon network. To do so we would divide the application area into zones. Pigeons now will set out to deliver messages to nodes based on zones. So instead of having to travel to scattered destinations, a pigeon in most cases will travel to a particular locality and be back to its home node. So on an average it travels far less than it had to otherwise. That is exactly what pizza delivery boys do; they distribute the orders amongst them based on the locality and then set out for delivery. This is something which humans have been doing for a very long time. This paper utilizes this simple human mechanism to achieve better results.

To reduce average message delay in the pigeon buffers, the buffer size s should preferably not be very high and the number of pigeons dedicated to each node should be high. Employing low cost energy efficient devices with moderate storage capacity will thus serve our purpose.

From [1] we note that nodes in HoP-DTN are distributed uniformly over the deployment area. In such a case we find the centroid and divide the whole area into four quarters on the basis of this centroid point. Thus considering that the deployment area is a rectangle with uniformly distributed nodes (as modeled in [1]), from Fig. 2 we note that x_m and y_m are the maximum values of the x and y coordinates of the deployment area respectively. (x_c, y_c) are the coordinates of the centroid of this deployment area, wherein $x_c = \frac{0+x_m}{2}$ and $y_c = \frac{0+y_m}{2}$.

Therefore any point lying inside the coordinates $(x_0, y_c), (x_c, y_c), (x_c, y_m), (x_0, y_m)$ is in Zone 1. Similarly the coordinates of all the zones is presented in Table 1. For simplicity, we have considered 4 zones, but that could be changed to any suitable value the network demands. In essence, we are decreasing the size of the problem domain to its one fourth.

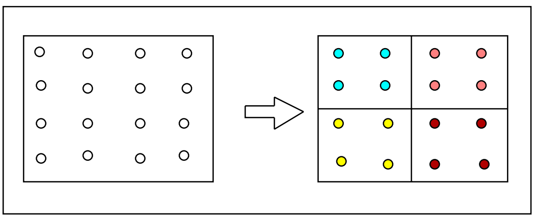


Fig. 1 Deployment area has to be partitioned into 4 zones

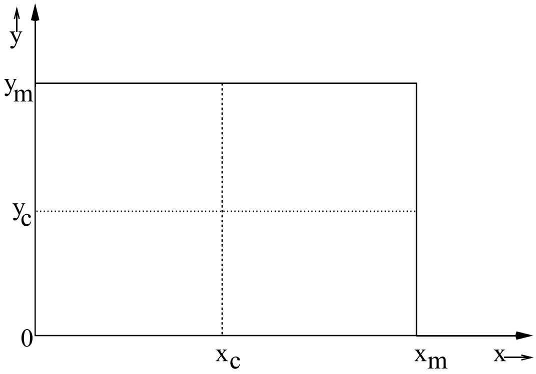


Fig. 2 Partitioning of deployment area into zones w.r.t. the centroid

Table 1 Coordinates of the 4 zones

Zone #	Coordinates
1	$(x_0, y_c), (x_0, y_m), (x_c, y_m), (x_c, y_c)$
2	$(x_c, y_c), (x_c, y_m), (x_m, y_m), (x_m, y_c)$
3	$(x_c, y_0), (x_c, y_c), (x_m, y_c), (x_m, y_0)$
4	$(x_0, y_0), (x_0, y_c), (x_c, y_c), (x_c, y_0)$

Continuing with the pizza boy strategy, we employ four pigeons for message delivery to the four zones. A pigeon should be labeled for a particular zone in an on demand basis. Suppose a message is generated for a node in zone 3 (and all pigeons are free) then a pigeon is selected and termed as P_3 for that cycle. Similarly if only one pigeon is with the home node at a particular time, it is designated a zone according to the first message generated. So basically, a node would operate with pigeons labeled as P_1, P_2, P_3 and P_4 when the number of zones are 4.

If though the arrangement of nodes is not uniform but shows a clustered behaviour, we can apply any clustering algorithm to find the zones. One problem that could arise while using some popular clustering algorithms like K-Nearest Neighbor would be that, we would not be able to control the number of clusters formed, hence increasing the zone-wise pigeon types that eventually might not turn out to be fruitful. For this reason the best mechanism we can choose could be some K-means algorithm where we can at least control the maximum number of clusters. This K should ideally be less than s, i.e. K<= s (s is the pigeon buffer size).

In the pizza delivery system, the strategy is good as long as the pizza remains hot and edible (one obviously can't wait for days for another order to come from the same locality), similarly we need to deliver the messages before they expire. Hence a long wait for another message to arrive for the same zone could potentially hamper the performance of the network. As such we need to define a threshold limit on pigeon waiting time based on network demands and average message expiration time in the network. Most of this problem is already tackled by the TTI scheduling mechanism [21]. As already discussed in Sect. 2, this scheduling scheme is the only one to put an upper bound on the worst case message delay. According to TTI the maximum delay possible for the 1^{st} message that entered the pigeon buffer (as it suffers maximum delay/waiting) is always

$$< (k-1)T + (s-k+1)t \tag{1}$$

wherein t, T and k are thresholds that work w.r.t. the number of messages in pigeon P's buffer (P^b) in the following manner:

1. If $P^b < k$, and delay caused to the latest message in P has crossed T, then P takes-off.
2. If $P^b >= k$, and delay caused to the latest message in P has crossed t, then P takes-off.

There could be times when more than one pigeon does not get filled up after the threshold time expires. In such cases it would be wise to adopt the concept of clubbing. The rules for clubbing are as such:

1. If more than one pigeon has empty buffer slots such that it is filled less than half of s, then club two such pigeons together. Another such pigeon will only be clubbed with the previous batch iff,

$$\sum_{i=1}^{4} P_i^b <= \quad s \tag{2}$$

$$\text{such that } \forall P_i^b <= \quad s/2 \tag{3}$$

   where P_i^b denotes the number of messages in the buffer of pigeon P_i.
2. If though there exists a pigeon such that $s > P_i^b > s/2$ holds true, then clubbing its contents with another pigeon is not allowed as it would nullify all the efficiency of this scheme. Even if there is just one message with a pigeon and another has more than $s/2$ but less than s, we would let them travel separately as that would give the pigeon with one message the chance to deliver the message quicker, return quicker and save energy.

From the simulation result plot of Fig. 11 we validate the need for clubbing (more discussions in Sect. 7). We call this modified HoP-DTN, Zonified HoP-DTN (ZHoP). Figure 3 gives a diagrammatic view of ZHoP-DTN.

5.2 Step 2: Friend Zone

Friend zone might not be the choice of zone for many of us in real life, but here it is quite useful. If we observe a number of runs and calculate the average distance required to travel to the other zones, we could come to a useful conclusion which would further enhance the efficiency of our scheme. The zone which is quickest to reach from own zone could be termed as the Friend Zone, and henceforth messages to this zone could be clubbed with those of own zone due to obvious reasons (this is only when clubbing is required). Thus when choosing among more than one pigeon contents to club with, we would always give highest priority to the messages of the Friend zone bound pigeon.

5.3 Step 3: High-Low Pigeons

Though our scheme aims to be efficient in energy conservation and routing, it could hamper the average waiting time of some messages. There could be times when the

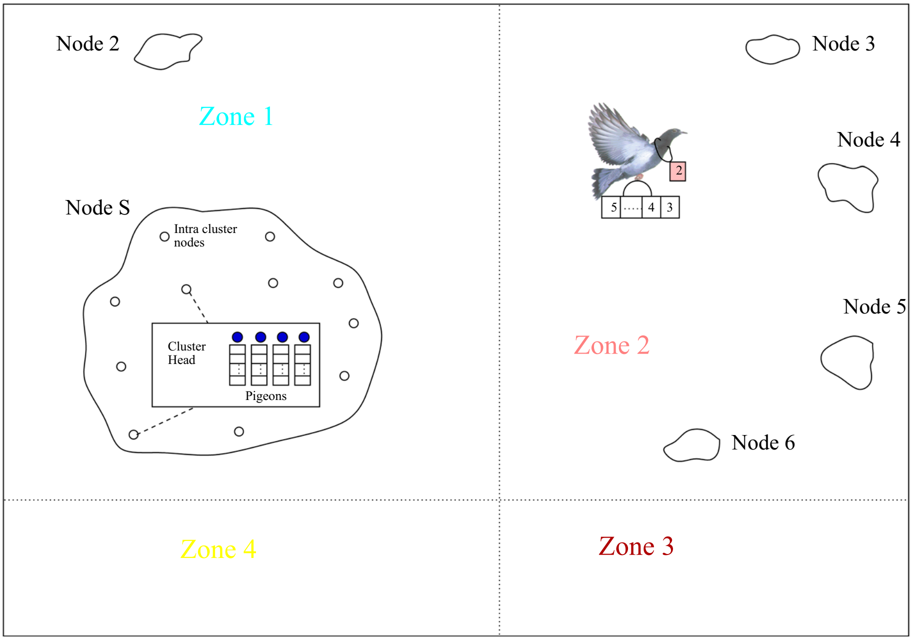


Fig. 3 A conceptual representation of the zonified HoP-DTN

timeliness of the message delivery determines its usefulness. For such high priority messages we would employ special pigeons known as High Pigeons, that would be generated on a demand basis. For messages other than high priority ones, we would employ normal pigeons known as Low Pigeons.

In case of implementing High-Low pigeons, we can more efficiently use the buffer space for quicker delivery takeoff. If s number of messages have not arrived at High pigeon and threshold to wait expires and there are messages waiting with other Low pigeons whose destination node (not zone) is common to any of the messages with the High pigeon, then those messages are to be added to the High pigeon buffer in a first-come-first-served manner (t, T and k for High pigeons are to be different for more efficiency). For efficiency, we have adapted the TTI scheduling to the High pigeons. A detailed explanation is given in Algorithm 1.

Algorithm 1: Implementing_High-Low_pigeons

if *a message M_h with high priority gets generated* **then**
    employ High pigeon P_h if not already initiated.
    Add M_h to its buffer.
    if $P_h^b >= k$ *and* $P_h^b < s$ **then**
        start timer since the last message arrived.
        if *time crosses t* **then**
            if *there are messages with Low pigeons whose destination node is similar to that of any message with P_h* **then**
                assign those messages to P_h in a FCFS manner.
                remove them from the buffers of the Low pigeon(s).
            pigeon P_h take-offs for message delivery.
    if $P_h^b < k$ **then**
        start timer since the arrival of last message.
        if *time crosses T* **then**
            if *there are messages with Low pigeons whose destination node is similar to that of any message with P_h* **then**
                assign those messages to P_h in a FCFS manner.
                remove them from the buffers of the Low pigeon(s).
            pigeon P_h take-offs for message delivery.

5.4 Step 4: Conditional Pickup

Classically a pigeon only collects messages from its designated home node. To optimize the whole situation in terms of energy conservation, we can allow conditional pickup by pigeons from nodes other than its home node, considering there are as many free spaces in its buffer. Let node A be the home node of pigeon P_A and node B is one of its message's destination node. $M_i^{X,Y}$ denotes a message from node X to node Y and i is the message number in the pigeon buffer, such that:

$$Y \in \{x \in E | x \neq X\} \tag{4}$$

wherein E is the set of all nodes in the network. and,

$$1 <= i <= s \tag{5}$$

When a pigeon reaches a destination, it can agree to deliver this node's message iff:

1. this node B has message(s) of the form $M^{B,A}$.
2. $M^{B,Y'}$ is picked up by P_A iff for all $M_i^{A,Y}$,

$$\gamma' \cap \gamma \neq \emptyset \tag{6}$$

   such that:
   $i = 1 \cdots (s-1)$,
   γ is the set of all Y with P_A and
   γ' is the set of all Y' with B.

5.5 Implementing Blockchain Technique to Enhance Energy Efficiency

Though Blockchain's primary beneficiaries lie in commerce, one can use its concept to aid other areas like that of energy efficiency. Recently some countries in Europe did a test trial of implementing this strategy to share energy between homes and electric cars [22]. To do so in ZHoP, each Cluster Head (CH) must be a prosumer (producer as well as consumer) of energy (preferably from solar photovoltaic systems). If there is a CH that produces too much energy in terms of solar energy such that it does'nt need all of it, it can then agree to sell/share this energy to incoming foreign pigeons. These foreign pigeons get charged up here and reduce the risk of dying due to energy shortage and might also not need to charge itself again when back home, thereby saving its home node's energy (as these networks are usually remote and CHs are stationed far off, the pigeons have rare chances of recharging otherwise). Speaking commercially, using any centralized Bitcoin payment (like SunExchange or EnergiMine) as incentives is not feasible in ZHoP/HoP-DTN deployment wise as that requires a constant Peer-to-Peer (P2P) contact with every participant node. So the incentive scheme that we have proposed for enabling and promoting Blockchain in ZHoP/HoP-DTN are as follows:

1. Service Incentive: The pigeon can agree to deliver some of this CH's messages considering there are free buffer space and the destination of the message is at least in the same zone as this zone or the pigeon's home zone.
2. Token Incentive: A token system can be used. A token of the amount of energy that is being taken from a CH by this pigeon would be issued. This token can be later used by this CH's pigeons to buy energy from the home node of that foreign pigeon. In a way we are again creating a Blockchain of tokens where the transaction is all about energy instead of money. Both parties will have a copy each of this transaction for safe keeping and verification. Additionally these tokens can be protected using cryptographical techniques.

Figure 4 shows a pictorial representation of the venture. For more efficiency, we can make the CH (apart from serving the mobile pigeons) act like the Utilities and Grid operators providing energy to other intra cluster nodes (both consumers or prosumers).

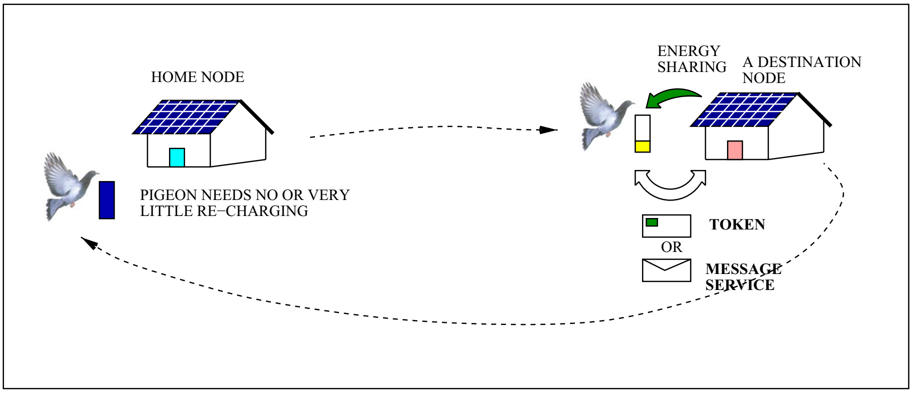


Fig. 4 Blockchain Sharing of Energy in ZHoP-DTN

6 Mathematical Modeling

We assume that the message generation follows a Poisson process. We also consider that the system can generate any number of messages i.e. capacity is infinite and the buffer capacity (of node) too is unlimited. Messages are taken up by the pigeons in a First Come First Serve (FCFS) basis. Here pigeons are the servers and our proposed system has five types of pigeons, namely P_H, P_1, P_2, P_3 and P_4. Thus the model that we have proposed in this paper is a $M/G/c/\infty/\infty/FIFO$ queueing model according to Kendall's Notation [23]. Here M, G, $FIFO$ have the usual scientific meaning and $c = 5 \times s$ as we have five types of pigeons with capacity s that serve the incoming messages.

This is a queueing model with multiple waiting lines (every pigeon has its own queue), and can be represented as a 5×5 switch model [24]. Since the selection of pigeons among the five types of pigeons is dependent on the algorithm, i.e. any message can be assigned to any pigeon (broadly speaking), the system becomes a 5×5 switch model. 5×5 is a discrete time queueing system with five parallel servers (pigeons) and five types of jobs (messages).

If i denotes the type of jobs (i can be Zone numbers or High pigeon), then probability of it arriving is given by r_i.

Therefore the probability that zero messages of type i is arriving is given by $1 - r_i$, where $i = 1, 2 \dots 5$.

Considering that the message arrival is at the start of a time unit and it joins the queue at pigeon j with probability $t_{i,j}$ where $t_{i,j} > 0$, for $j = 1, 2 \dots 5$

and

$$t_{i,1} + t_{i,2} + t_{i,3} + t_{i,4} + t_{i,5} = 1 \tag{7}$$

Also for each pigeon, the average number of messages arriving per time unit is assumed to be less than 1.

$$\therefore r_1 t_{1,j} + r_2 t_{2,j} + r_3 t_{3,j} + r_4 t_{4,j} + r_5 t_{5,j} < 1 \text{ for } j = 1, 2 \dots 5 \tag{8}$$

The above condition means that only one message of either type 1 or 2 or … 5 can arrive at a server at one time. This feature attributes to the ergodic nature of the system.

$$\therefore r_1 = r_2 = r_3 = r_4 = r_5 = 1 \tag{9}$$

is not possible according to our assumption as then it would mean that all five types of job have 100% probability to arrive at the same instant, which is not possible. Also a new message will get assigned to only one of the servers and not more than one at a time. The $5X5$ switch model is pictorially explained in Fig. 5.

The interarrival times of messages is considered to be exponential with mean $\frac{1}{\lambda}$ (λ is the arrival rate), along with an exponential service time with mean $\frac{1}{\mu}$ (μ is the service rate of one server). The number of servers (pigeons) is 5 but as each pigeon can service s messages at most, $c = 5s$. The occupation rate per server is supposed to be ρ, where

$$\rho = \frac{\lambda}{c\mu} \text{ is less than 1.} \tag{10}$$

Let there be n customers in the system, the equilibrium probability of which is denoted by p_n.

From [25], we get:

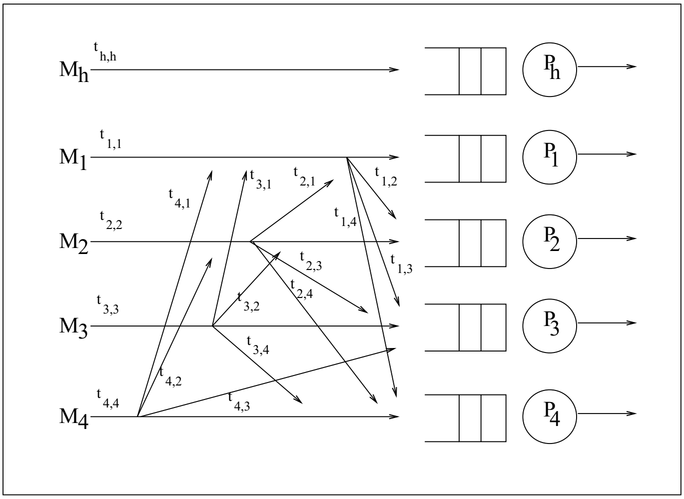


Fig. 5 The 5 × 5 queueing system model of ZHoP-DTN

$$p_n = \frac{(c\rho)^n}{n\,!}p_0, \text{ for } n = 0, \dots, c \tag{11}$$

and

$$p_{c+n} = \rho^n p_c \tag{12}$$

$$= \rho^n \frac{(c\rho)^c}{c\,!}p_0, \text{ for } n = 0, 1, 2 \dots \tag{13}$$

where

$$p_0 = (\sum_{n=0}^{c-1} \frac{c\rho^n}{n\,!} + \frac{c\rho^c}{c\,!}\frac{1}{1-\rho})^{-1} \tag{14}$$

Let Π_w denote the probability of the amount of time that a message has to wait before it is assigned to a pigeon.

$$\Pi_w = p_c + p_{c+1} + p_{c+2} + \cdots \tag{15}$$

$$= \frac{p_c}{1-\rho} \tag{16}$$

$$= (\frac{(c\rho)^c}{c\,!})((1-\rho)\sum_{n=0}^{c-1} \frac{(c\rho)^n}{n\,!} + \frac{(c\rho)^c}{c\,!}) \tag{17}$$

6.1 Mean Queue Length and Mean Waiting Time

The mean queue length is provided by,

$$E(L^q) = \sum_{n=0}^{\infty} n p_{c+n} \tag{18}$$

$$= \frac{p_c}{1-\rho} \sum_{n=0}^{\infty} n(1-\rho)\rho^n \tag{19}$$

$$= \Pi_w \frac{\rho}{1-\rho} \tag{20}$$

The mean waiting time can be derived from Little's Law,

$$E(W) = \Pi_w \frac{1}{1-\rho} \frac{1}{c\mu} \tag{21}$$

6.2 Distribution of the Waiting Time and the Sojourn Time

If D_k is the kth interdeparture time, where D_k is a random variable independently and exponentially distributed with mean $1/c\mu$, then by conditioning on the state seen on arrival we get,

$$P(W > t) = \sum_{n=0}^{\infty} P(\sum_{k=1}^{n+1} D_k > t) p_{c+n} \tag{22}$$

Proof of the above equation can be obtained from [25]. Also from [25] we find,

$$P(W > t) = \sum_{n=0}^{\infty} \sum_{k=0}^{n} \frac{(c\mu t)^k}{k\,!} e^{-c\mu t} p_c \rho^n \tag{23}$$

$$= \sum_{k=0}^{\infty} \sum_{n=k}^{\infty} \frac{(c\mu t)^k}{k\,!} e^{-c\mu t} p_c \rho^n \tag{24}$$

$$= \frac{p_c}{1-\rho} \sum_{k=0}^{\infty} \frac{(c\mu\rho t)^k}{k\,!} e^{-c\mu t} \tag{25}$$

$$= \Pi_w e^{-c\mu(1-\rho)t} \quad , t >= 0 \tag{26}$$

Therefore the conditional waiting time is given by,

$$P(W > t \| W > 0) = \frac{P(W > t)}{P(W > 0)} \tag{27}$$

$$=e^{-c\mu(1-\rho)t}, t >= 0 \tag{28}$$

The distribution of the sojourn time can be derived by conditioning on the length of the service time i.e.,

$$P(S > t) = P(W + B > t) \tag{29}$$

$$= \int_{x=0}^{\infty} P(W + x > t)\mu e^{-\mu x} dx \tag{30}$$

$$= \int_{x=0}^{t} P(W > t - x)\mu e^{-\mu x} dx + \int_{x=t}^{\infty} \mu e^{-\mu x} dx \tag{31}$$

$$= \int_{x=0}^{t} \Pi_w e^{-c\mu(1-\rho)(t-x)} \mu e^{-\mu x} dx + e^{-\mu t} \tag{32}$$

$$=\frac{\Pi_w}{1 - c(1-\rho)} e^{-c\mu(1-\rho)t} + (1 - \frac{\Pi_w}{1 - c(1-\rho)})e^{-\mu t}, \tag{33}$$

such that $c(1 - \rho) \neq 1$.

If however $c(1 - \rho) = 1$, then

$$P(S > t) = (\mu \Pi_w t + 1)e^{-\mu t} \tag{34}$$

All the pigeons (servers) in our system are identical in terms of their capacities and capabilities. For systems in statistical equilibrium, the average number of busy servers, L_s , is: $L_s = \lambda E(s) = \lambda/\mu$. The long-run average server utilization is: $\rho = \frac{L_s}{c} = \frac{\lambda}{c\mu}$, where $\lambda < c\mu$ for stable systems.

6.3 Costs Related to this Queueing System

Costs can be associated with various aspects of the waiting line or servers: System incurs a cost for each customer in the queue, say at a rate of ξ_c per hour per customer.

If W_j^Q denotes the time customer j spends in queue Q then,

$$\text{Average cost per customer } = \xi_c \frac{1}{N} \frac{1}{Q} \sum_{j=1}^{N} \sum_{Q=1}^{5} W_j^Q \tag{35}$$

Servers also do generate costs to the system. In our analogy, there are c servers which run parallel and have utilization ρ. If the cost of a server per hour it is busy is denoted by ξ_s then,

$$\text{Total server cost is } = \xi_s c\rho \tag{36}$$

$$= \xi_s 5s\rho \tag{37}$$

7 Simulation Analysis

In this penultimate section we elaborate the simulation scenario and the comparison yardstick chosen for the fair evaluation of the schemes. After that we discuss and analyse the results obtained from the extensive simulations that we have conducted, complete with stress tests to test the robustness of our scheme.

7.1 Simulation Scenario

To test the serviceability of our proposed scheme and compare it to other existing ones, we have built a simulated scenario using the *C* programming language. There are 100 nodes spread uniformly across a deployment area of 100 unit square with 4 zones, each containing 25 nodes (CH). Each pigeon has the capability to carry bulk messages from home node to various destination nodes taking the help of a distance adjacency matrix and using the Branch and Bound algorithm as the basic TSP solution. The message delay caused in pigeon buffer is the waiting time of the message from its inception into the buffer to that pigeon's take-off. All pigeons have flying speed of 1 unit of distance per unit of time (for simulation 1 unit of time is equivalent to 1 second). All other simulation parameters are shown in Table 2. Energy usage by Pigeon during its travail is 1 unit of energy per unit of distance. When Blockchain Sharing of Energy is employed, a CH maintains a threshold value of its own remaining energy level. If its energy is more than the *NodeShare_Threshold* it takes part in the Blockchain energy sharing scheme. In the simulations, we have used the service incentive.

7.2 Comparison Yardstick

This work can be divided into three major contribution areas: Scheduling, Routing and Energy Efficiency. For comparison purposes, the protocols that we have chosen from existing literature are the TTI HoP [21], PriorityB HoP [20] and the StorageB HoP (since we are working with storage based pigeons that can perform bulk delivery, we considered only the most popular storage based pigeon scheduling schemes). Two levels of message priority has been considered: High and Low (for ZHoP and PriorityB). For routing we denote the existing HoP-DTN routing scheme as the Vanilla HoP. As no separate scheme exists for energy efficiency and routing in HoP-DTN we use the basic model for comparisons. Apart from these, we have also created two variants of ZHoP called Pure ZHoP and AllClubbed ZHoP. In Pure ZHoP, all the pigeons are packed with messages for the same zone (i.e. no

Table 2 Simulation parameters

Parameters	Value/Range
Pigeon buffer size (s)	3 to 30
No. of pigeons (NoP)	5 to 45
No. of messages generated (NoM)	5 to 200
Simulation time (SimTime)	5 to 300 unit of time
Initial pigeon energy (Pmax)	10 to 30 unit of energy
Message expiration time (MeT)	20 to 110 unit of time
Node energy share threshold % (NodeShare_Threshold)	10 to 90%

Clubbing is required), which is an ideal case for our scheme (only the zone is same for all messages and not their destination CH). Similarly to show the full picture and add credible variety we have shown results of the worst case scenario of ZHoP, the AllClubbed ZHoP wherein all the pigeons are clubbed (i.e. all of them have to travel more than one zone to complete their delivery assignments). Nevertheless Clubbing is an important technique, and as stated in the previous sections we have highlighted its necessity with simulating the complete ZHoP scheme with one without clubbing in Fig. 11. Comparisons with these ideal and worst case ZHoP is only possible for energy and routing simulations and not scheduling as in scheduling we examine the behaviour of the scheduling strategy and not where the pigeon has to travel after take-off. When rate of message generation is low or nodes have very few messages to pass amongst themselves over large periods of time causing severe delay to all messages waiting with pigeons, we call them the less talkative networks. Besides applying Blockchain for energy efficiency, for an enhanced analysis we have additionally presented results for the system without employing Blockchain to show that by itself our scheme is a big improver on energy efficiency. The performance criteria on which all the schemes have been compared are average delay of messages in pigeon buffer before take-off, average worst case delay (the first message in a pigeon buffer is always to receive the maximum delay before take-off in that pigeon), number of pigeons that took-off, number of messages that remain unsent, percentage of dead pigeons, average remaining energy of pigeons after returning home, message delivery rate of schemes and number of messages expired before delivery.

7.3 Result Discussion

In this section we discuss, analyse and infer behavioural characteristics of the schemes from the results. Apart from discussing the scheduling, energy efficiency and routing issues via extensive simulations, we also do Stress Tests and Sensitivity Analysis of the system.

7.3.1 Impact of Scheduling

Figure 6 compares the proposed ZHoP to the existing TTI, PriorityB and StorageB (in HoP-DTN) while varying the simulation time and rate of message generation on their average delay caused to all messages due to waiting in the pigeon buffer before it takes-off. The results show that ZHoP incurs much less delay than PriorityB and StorageB but on an average has higher delay than TTI. Likewise Figs. 7, 8 shows the average delay caused to the first message in every pigeon. In this case ZHoP performs better than PriorityB and StorageB in all cases and on an average performs better than TTI. The number of pigeons that take-off according to the respective scheduling strategies is depicted in Figs. 9, 10, 11; this is heavily dependent on the delay between message arrivals for TTI and ZHoP (as they use delay thresholds). PriorityB and StorageB would not let its pigeon fly until it is full, so they might incur more delay but the number of pigeons that they use is less. ZHoP requires slightly more pigeons than TTI in some cases but in a less talkative network TTI uses much more than ZHoP. A similar metric is the number of messages that remain with the home node, assigned to pigeons that did not take-off due to end of *SimTime* and pigeon take-off criteria of its respective scheduling scheme. This is shown in Fig. 8 from which we see that ZHoP performs better than PriorityB and StorageB but lags a little behind TTI. The plot of Figs. 12 and 13 shows how *NoM* and *SimTime*, affect the average message (all and first) delay. All these results are compared and a summary is presented in Table 3 where we can

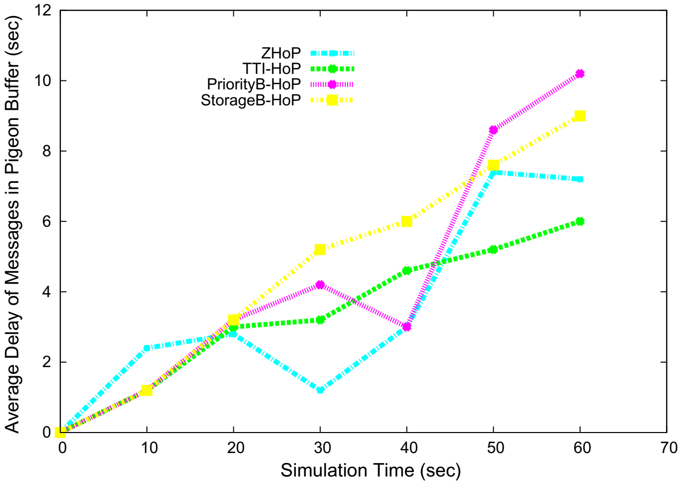


Fig. 6 Average delay of messages in pigeon buffer before take-off in a less talkative network ($s = 7$, $NoM = 10$)

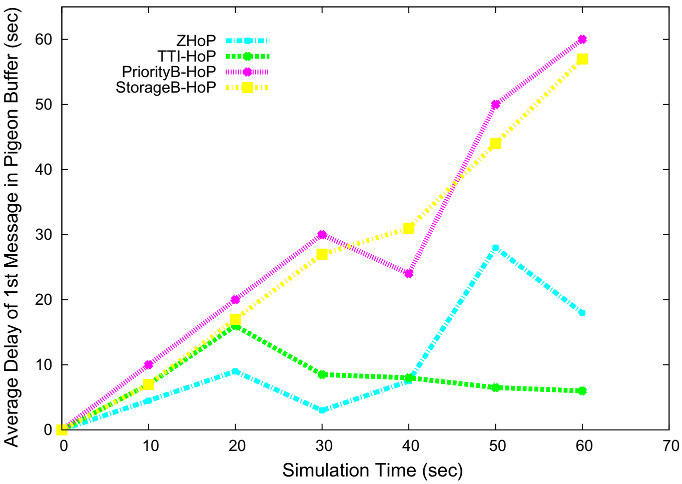


Fig. 7 Average delay of all first messages in pigeon buffer in a less talkative network ($s = 7$, $NoM = 10$)

see the percentage of improvement ZHoP has over the other schemes. In Fig. 10 we plotted simulation results vis-s-vis the theoretical results derived from the Equation 1. The comparison shows that even for a less talkative network under stress test the delay of the first message never crosses the theoretical upper bound. In fact, on an average, the theoretical worst case delay upper bound is 294.173% more than the simulated average message delay.

7.3.2 Impact on Energy Efficiency

When it comes to energy efficiency, ZHoP combined with Blockchain completely outperforms the existing Vanilla HoP. As stated earlier we also compare for each criteria, the Pure

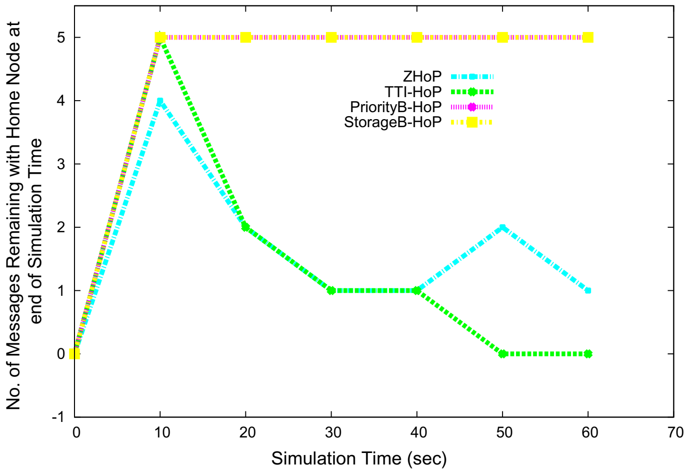


Fig. 8 Messages remaining with home node in a less talkative network ($s = 7$, $NoM = 10$)

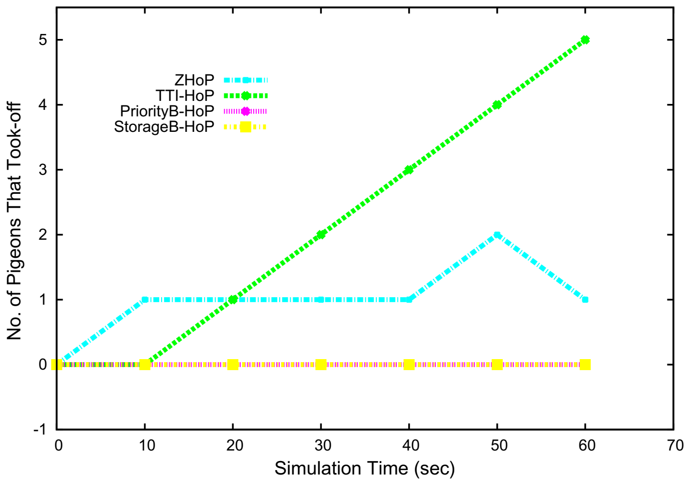


Fig. 9 Number of pigeons that take-off in a less talkative network ($s = 7$, $NoM = 10$)

and AllClubbed ZHoP, and in all circumstances (including the stress test) all versions of ZHoP perform stupendously better than Vanilla HoP. For a fair comparison, we applied Blockchain to Vanilla HoP too for comparisons. Figure 14 shows that no matter how large the *NodeShare_Threshold* be, no version of ZHoP produces even a single dead pigeon (a pigeon whose battery is dead) but Vanilla HoP produces many dead pigeons in the same scenario. Likewise when number of pigeons in the network is varied (see Figs. 15, 16, 17) or the initial pigeon energy is varied even under stress test, each time the results tell the same story. Figures 18 and 19 show that whatever the circumstances, all versions of ZHoP have higher remaining energy when pigeons are back home than the Vanilla HoP. Amongst themselves, quite predictably, Pure ZHoP performs the best. The plots of Figs. 22 and 23 give more insight into how the schemes behave with respect to (w.r.t.) varying the number of pigeons, initial energy of pigeons and energy share thresholds of nodes (two at

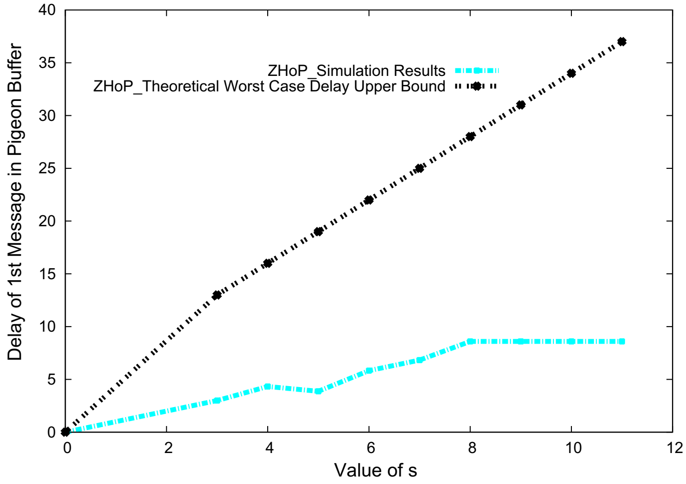


Fig. 10 Theoretical v/s simulation worst-case delay comparison (*NoM* = 30, rate of message generation = 50%)

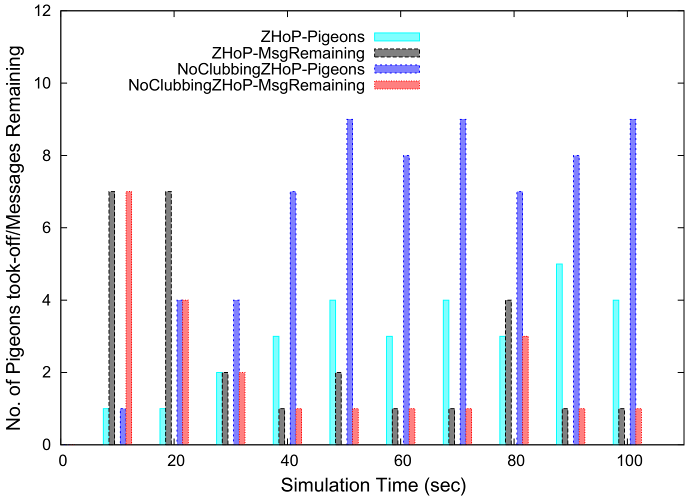


Fig. 11 Comparison of full ZHoP and ZHoP without clubbing ($s = 5$, $NoM = 10$): shows the necessity of clubbing

a time). Our proposed mechanism is more energy efficient than the Vanilla HoP even without Blockchain, but adding it increases its energy efficiency (see Fig. 25 for a side-by-side comparison of Pure ZHoP and Vanilla HoP with and without Blockchain). Figures 20 and 21 shows that when most of the pigeons have average remaining energy of zero in Vanilla HoP, all versions of ZHoP perform way more respectably. Same can be seen in their tendency to produce dead pigeons (see Figs. 16 and 17). The results also suggest that applying Blockchain improves the energy efficiency of Vanilla HoP but it still is way behind ZHoP. These plots and tables confirm the fact that when we use ZHoP+Blockchain, dead pigeons

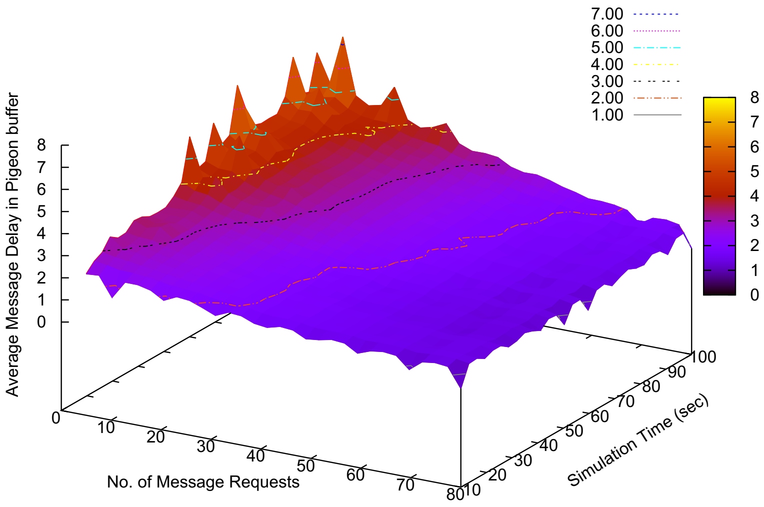


Fig. 12 Average message delay incurred by ZHoP

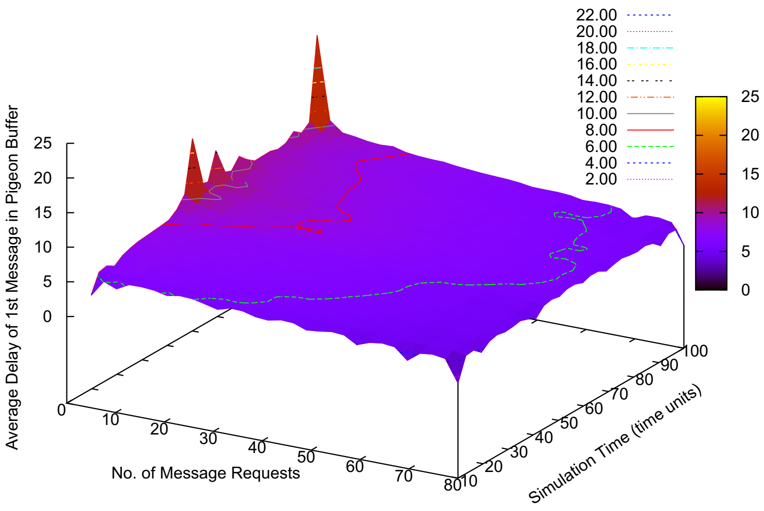


Fig. 13 Average worst-case message delay incurred by ZHoP

are almost never to occur. On the other hand Vanilla HoP might still produce dead pigeons (albeit improving in pigeon remaining energy) when accompanied by Blockchain. So for a full-proof energy efficient system meant for rural networks ZHoP+Blockchain is the way to go (Figs. 22, 23, 24, 25).

7.3.3 Impact on Routing

One of the primary efforts of this work is to improve the delivery rate of pigeons by trying to decrease the distance that each pigeon travels. The simulation results certify our

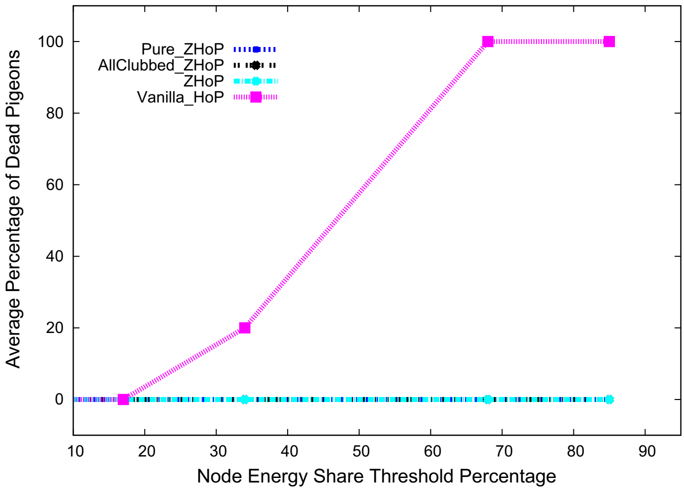


Fig. 14 No. of dead pigeons, blockchain applied ($s = 5$, $NoP = 5$, $Pmax = 10$)

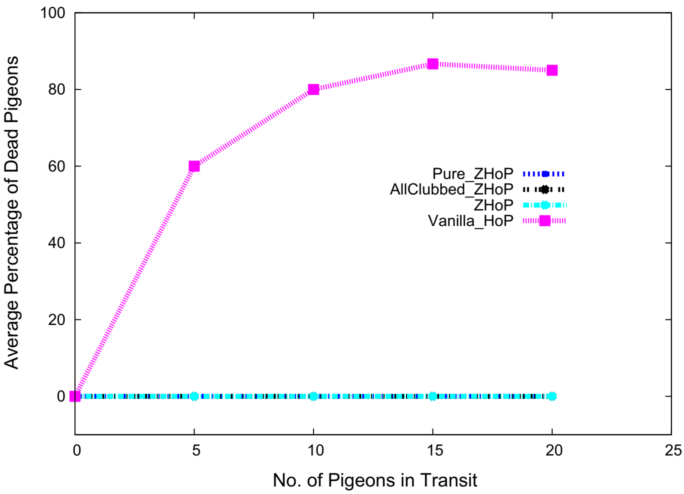


Fig. 15 No. of dead pigeons, blockchain applied ($s = 5$, $Pmax = 10$, $NodeShare\_Threshold = 50\%$)

success in achieving the theoretical assumptions set in Sect. 5. To better understand the behaviour of each version of ZHoP vis-a-vis varying s and MeT simultaneously, one can refer the Tables 3 and 4. In all cases Pure ZHoP, AllClubbed ZHoP and ZHoP perform much better than Vanilla HoP and their amount of improvement over Vanilla HoP. As the deployment area gets divided into zones and each pigeon travels only specific zones instead of destinations scattered around the network, it decreases the chance of messages expiring and hence increases delivery rate (Figs. 26, 27).

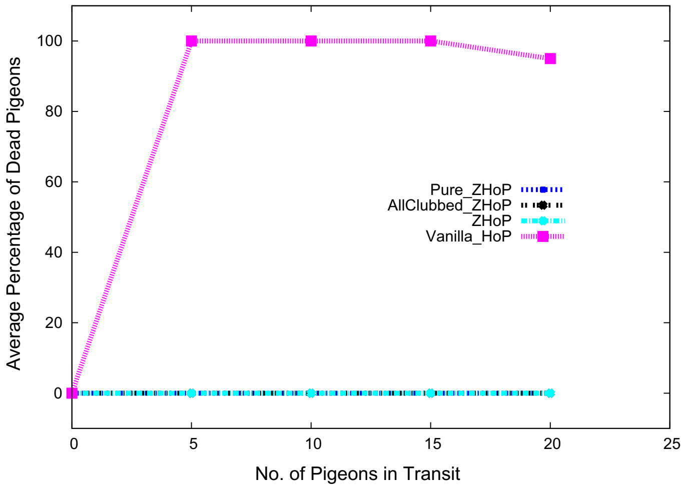


Fig. 16 No. of dead pigeons, blockchain NOT applied ($s = 5$, $Pmax = 10$)

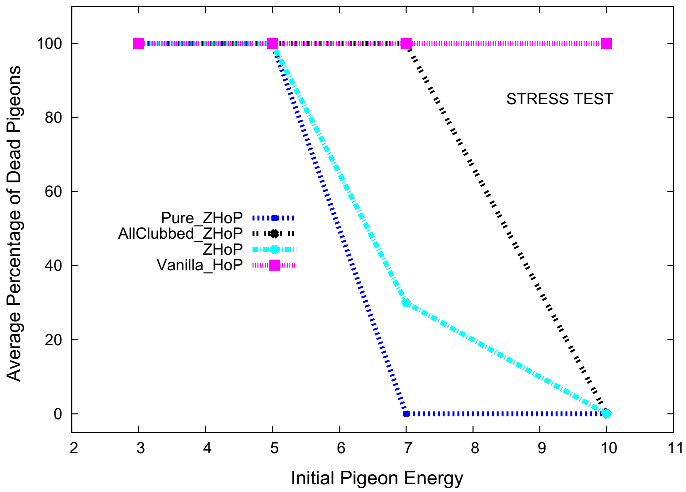


Fig. 17 Stress test: average percentage of dead pigeons, blockchain NOT applied ($s = 5$, $NoP = 10$)

7.3.4 Behaviour in a Less Talkative Network

When the inter-arrival message delay increases in a less talkative network, it has an adverse affect on the scheduling strategies. Scheduling strategies like TTI and ZHoP are capable to deal with it as they both have an upper limit on the maximum possible delay. The other pre-existing schemes PriorityB and StorageB are incapable of working around such a situation, as their pigeons cannot take-off until they are full. So in theory if the next message is to arrive days after the first, the first message might expire at the home node itself (for PriorityB and StorageB). PriorityB still has chance to perform better if

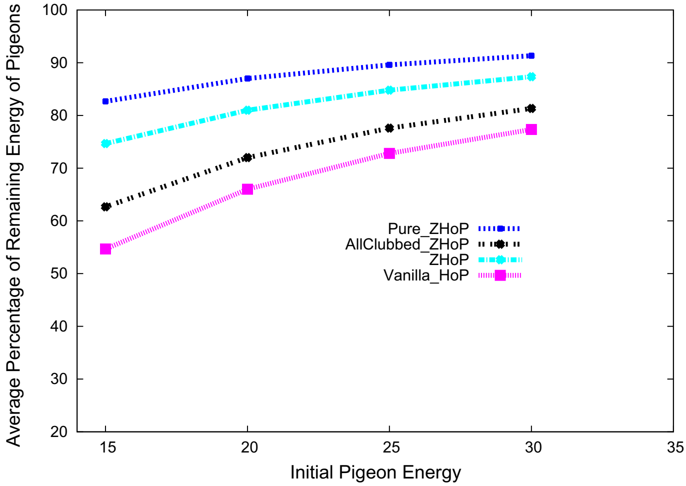


Fig. 18 Average remaining energy of pigeons, blockchain applied ($s = 5$, $NoP = 5$, $NodeShare\_Threshold = 50\%$)

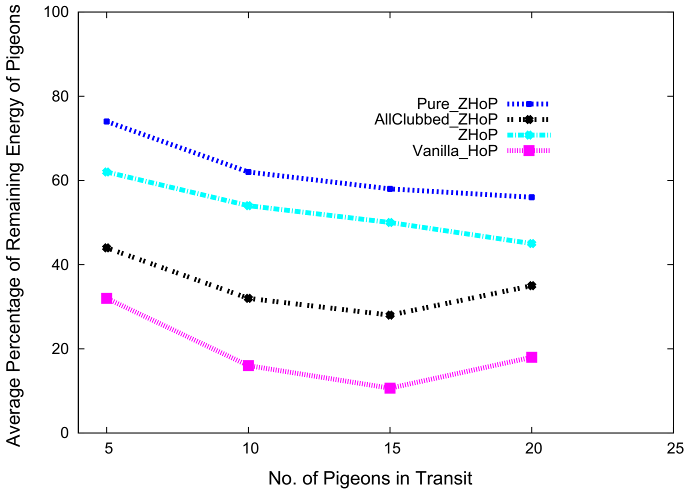


Fig. 19 Average remaining energy of pigeons, blockchain applied ($s = 5$, $Pmax = 10$, $NodeShare\_Threshold = 50\%$)

it gets high priority messages. Figures 6 and 8 show that as expected ZHoP performs better than PriorityB and StorageB in terms of delay and number of messages unsent. It though sometimes lags behind TTI in the delay department. Number of pigeons used by ZHoP and TTI is obviously more than that of the other two schemes. This is so because to lessen delay we have to use more pigeons which in a less talkative network might travel unfilled. Thing to be noted from the graph of Fig. 9 is that in a less talkative network ZHoP requires lesser number of pigeons than TTI. This is due to the clubbing feature of ZHoP which is absent from TTI. In this case ZHoP turns a disadvantage of

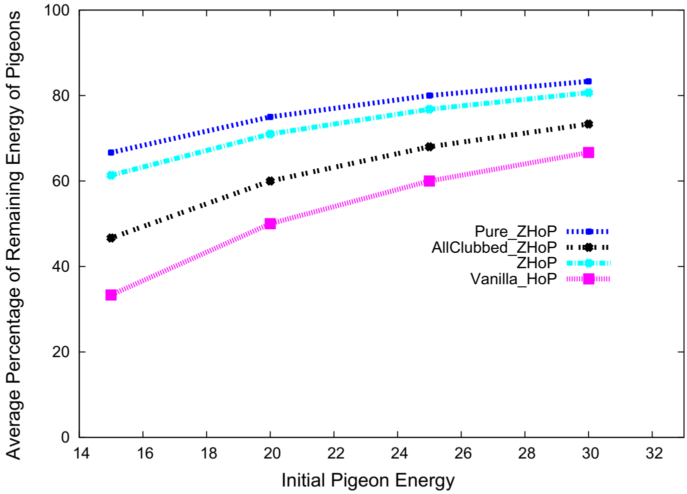


Fig. 20 Average remaining energy of pigeons when back home, blockchain NOT applied ($s = 5$, $NoP = 5$)

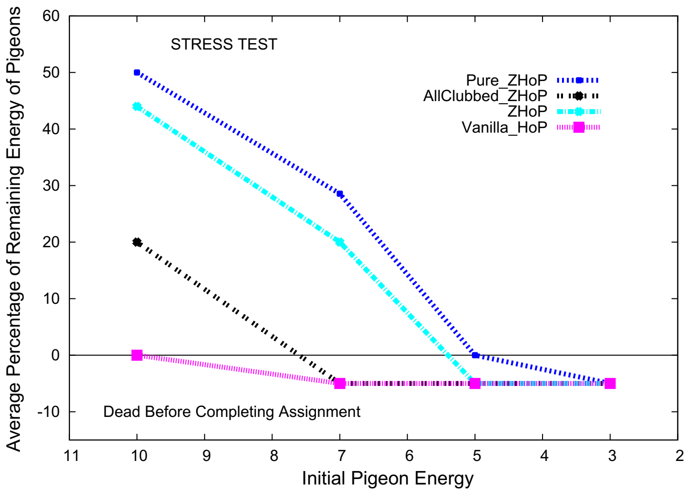


Fig. 21 Stress test: average remaining energy of pigeons w.r.t. very low initial pigeon energy, blockchain NOT applied ($s = 5$, $NoP = 10$)

its (having to maintain multiple pigeon queues) into its advantage. In Fig. 9 a fluctuating pattern can be seen in the number of pigeons that took-off with ZHoP and TTI. This phenomenon can be attributed to the fact that both of these schemes use delay thresholds which govern the take-offs of the pigeons. ZHoP in addition to it employs clubbing which makes the plots even more fluctuating. This is so because, although *SimTime* is increasing, *NoM* remains the same, hence inter-arrival delay of messages is increasing. Therefore thresholds *t*/*T* crosses for the pigeons, and they are ready to fly, but due to clubbing the *NoP* does not increase each time but the messages remaining with home node decreases in that curve (see Fig. 8). At other times the message inter-arrival delay

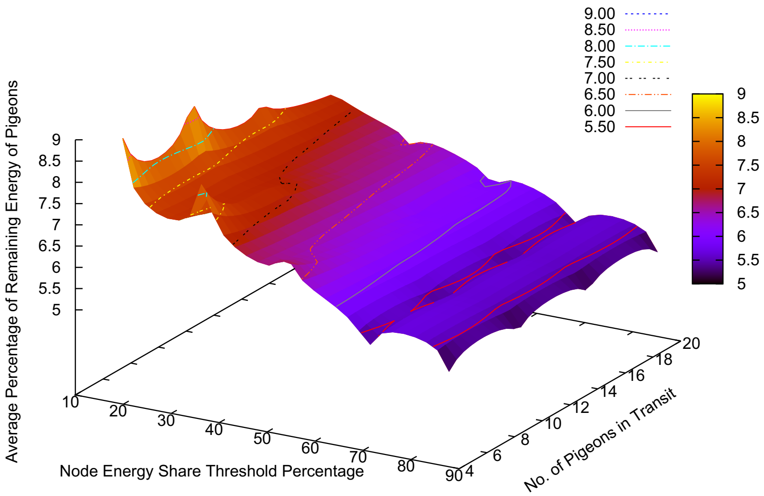


Fig. 22 Average remaining energy of pigeons in pure ZHoP, blockchain applied ($s = 5$, $Pmax = 10$)

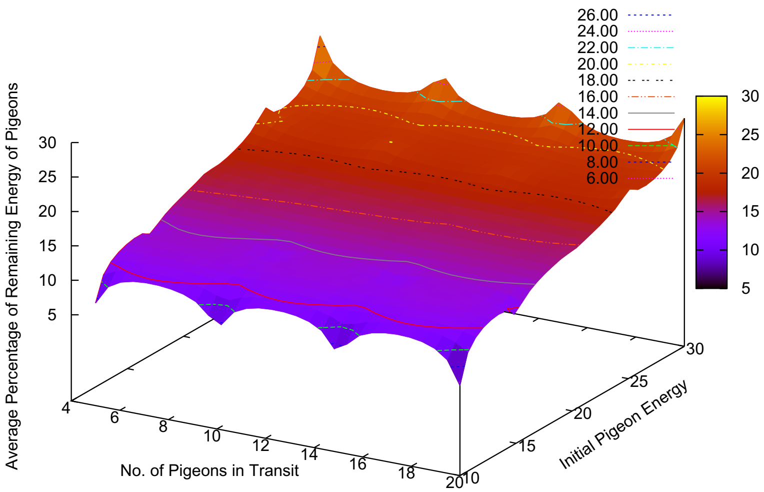


Fig. 23 Average remaining energy of pigeons in pure ZHoP, blockchain applied ($s = 5$, $NodeShare\_Threshold = 50\%$)

is such that a pigeon's t/T crosses only after another pigeon has taken-off, so this time there is no opportunity for clubbing. Hence the zig-zag curve.

7.3.5 Behaviour under Stress Test

Stress Tests uncover the hidden flaws in every system. We put our proposed scheme along with the existing one under the scanner of a few stress test to do the same. The stress test in Fig. 17 shows the behaviour when very low initial energy is assigned to

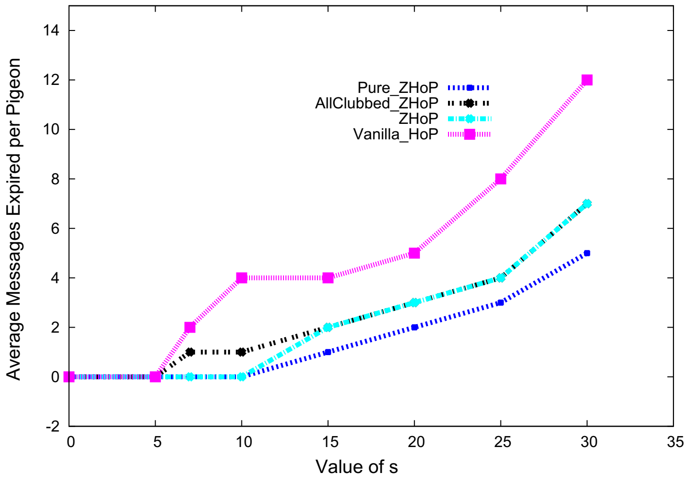


Fig. 24 Messages expired per pigeon with varying pigeon buffer size *s* (MeT=70): stress test when *s* gets higher

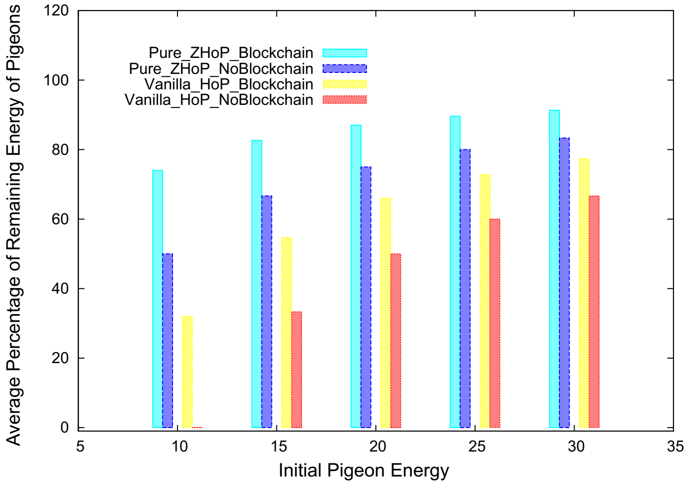


Fig. 25 Average remaining energy of pigeons employing pure ZHoP & vanilla HoP with and without blockchain sharing of energy ($s = 5$, $NoP = 5$, $NodeShare\_Threshold = 50\%$)

pigeons and no Blockchain sharing of energy is done. Vanilla HoP has all its pigeons dead in all of the energy levels whereas ZHoP starts to produce dead pigeons on decreasing *Pmax* and reaches 100% dead pigeons at *Pmax* of 5 onwards. Such low *Pmax* is almost never going to be assigned in real life, and also when Blockchain is applied ZHoP (and all its versions) never has dead pigeons as they keep on recharging themselves on the way. In the same settings when we record the average remaining energy of pigeons, we see that (see Fig. 21) due to total battery drainage, pigeons die before accomplishing their message delivery tasks. Even in this case Pure ZHoP, AllClubbed ZHoP and ZHoP perform much better than Vanilla HoP. The solution again is to apply

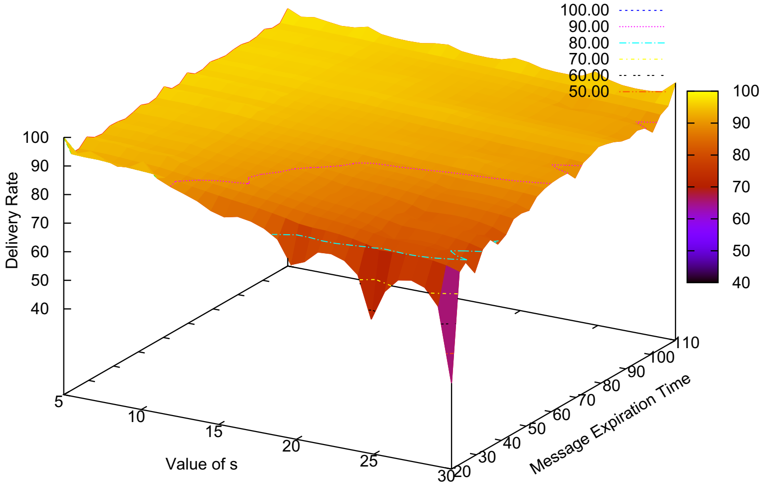


Fig. 26 Delivery rate of pure ZHoP

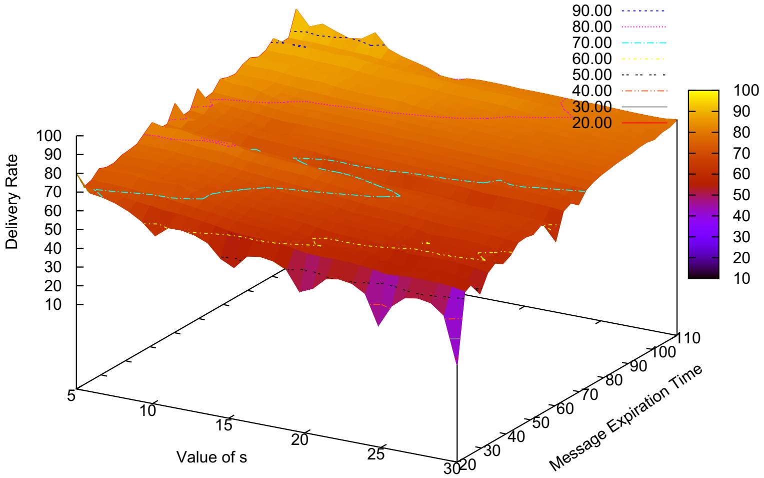


Fig. 27 Delivery rate of vanilla HoP

Blockchain. Figure 24 depicts an account of the number of messages that expire in the pigeon buffer before they are delivered w.r.t. the value of s. This gives an interesting insight into the fact that if we increase the pigeon buffer size s, we might increase the chances of message expiry. That is so because when number of pigeon destinations rises, it has to travel more distances and individual message delay increases (especially those belonging to the last destination). Since ZHoP pigeons have to travel less and they incur less delay before take-off, their chances of having expired messages is pretty less than Vanilla HoP. Also high priority messages are treated with emergency in ZHoP. PriorityB too treats high priority messages with emergency but ZHoP has an advantage

Table 3 A complete summary of average percentage of improvement of ZHoP over qther protocols in HoP-DTN

Performance metric	Average Percentage of Improvement of ZHoP over Protocols		
	TTI HoP (%)	PriorityB HoP (%)	StorageB HoP (%)
Average remaining energy of pigeons_Blockchain	157.425	157.425	157.425
Rate of dead pigeons_Blockchain	453.333	453.333	453.333
Average remaining energy of pigeons_NOBlockchain	233.877	233.877	233.877
Rate of dead pigeons_NOBlockchain	544.444	544.444	544.444
Pigeons required	96.223	− 28.116	− 30.999
Messages unsent	− 59.722	23.735	18.056
Average delay of messages	− 20.537	50.829	243.717
Delay of 1st message	6.503	170.309	181.18
Delivery rate	29.063	29.063	29.063
Messages expired per pigeon	54.524	54.524	54.524

Table 4 A Complete summary of average percentage of improvement of pure (Ideal) and AllClubbed (Worst) ZHoP over vanilla (TTI/PriorityB/StorageB) HoP

Performance Metric	Average percentage of improvement over vanilla HoP	
	Pure ZHoP (%)	All-Clubbed ZHoP (%)
Average remaining energy of pigeons_Blockchain	194.259	71.591
Rate of dead pigeons_Blockchain	453.333	453.333
Average remaining energy of pigeons_NOBlockchain	275.397	112.659
Rate of dead pigeons_NOBlockchain	608.333	525
Delivery rate	37.863	26.443
Messages expired per pigeon	65.119	43.81

over it as it clubs messages (low) belonging to the same destination as the high priority messages with its High Pigeon.

7.3.6 Sensitivity Analysis

Every output is dependent on the inputs and the manner in which they are varied. To understand the impact of each input (and the extent of it) on the performance of the system is to be able to have a better control over the behaviour of the system. As such for an even more in-depth analysis of the scheme, we have done an extensive sensitivity analysis. The results have been presented in Tables 5, 6 and 7. We have given three different tables to show all dependabilities of the three main outcomes of this research work: Scheduling, Routing and Energy Efficiency in ZHoP/HoP-DTN. As these are governed by different parameters, their influence

Table 5 Sensitivity analysis of ZHoP w.r.t. varying input parameters on various performance metrics

Performance Metric	Sensitivity of input parameter		
	s	No. of messages generated	Rate of message generation
No. of pigeons that Took-off	0.644	1.016	0.119
Messages remaining at home node	3.2666	1.288	0.071
Delay of 1st message in pigeon buffer	0.965	0.052	0.242
Average delay of messages	0.359	0.08	0.253

Table 6 Sensitivity analysis of different schemes w.r.t. input parameters on delivery rate

Protocols	Sensitivity on delivery rate	
	s	Message expiration time
Pure ZHoP	0.029	0.044
AllClubbed ZHoP	0.06	0.079
ZHoP	0.044	0.065
Vanilla HoP	0.125	0.116

Table 7 Sensitivity analysis of different schemes w.r.t. input parameters

Protocols	Sensitivity analysis of different schemes w.r.t. input parameters		
	Pmax	Node energy share threshold	No. of pigeons in transit
No. of dead pigeons			
Pure ZHoP	~ 0	~ 0	~ 0
AllClubbed ZHoP	~ 0	~ 0	~ 0
ZHoP	~ 0	~ 0	~ 0
Vanilla HoP	0.8	1.375	0.885
Average remaining energy of pigeons			
Pure ZHoP	1.351	0.111	0.104
AllClubbed ZHoP	2.273	0.207	0.14
ZHoP	1.613	0.134	0.099
Vanilla HoP	3.125	0.276	0.267

have been separately shown for clarity. The sensitivity analysis given in Tables 5, 6 and 7 has been performed using the below formula [26]:

$$\text{Sensitivity for input } x = \frac{\%\text{ of change in output}}{\%\text{ of change in input}}$$

From all the three tables we can deduce that some input parameters have especially more effect on some outputs than the others. Notable is the high dependence of the messages unsent on the value of s in case of ZHoP. Quite understandably the number of messages generated too has a big effect on the number of messages unsent. The rate of message generation has the most potent effect on average and worst case delays of a message which is very probable since this rate dictates whether a network is a talkative one or a less talkative one. More the talkative the network lesser is the delay caused to every message waiting in the pigeon for its take-off. The delay of first (worst) message is also sensitive to the value of s as more the size, more the delay. In terms of delivery rate, Vanilla HoP is more sensitive towards the input parameters than all the versions of ZHoP (see Table 6). All the versions of ZHoP seem to have no sensitivity to input parameters in case of number of dead pigeons, as already discussed, they produce almost none due to their strategy (see Table 7). Vanilla HoP on the other hand weighs heavily on each input parameter. In case of remaining energy levels of pigeons, the parameter to which the performance of all schemes is most sensitive to is the initial pigeon energy (*Pmax*). Among all the schemes, the Vanilla HoP is most sensitive towards the input parameters even in this case. All in all the analysis proves that our proposed scheme, ZHoP is a much stabler scheme than the existing Vanilla HoP-DTN.

8 Conclusion

The primary objective of this research work was to design an energy efficient model of HoP-DTN that has a smarter routing technique and a scheduling strategy to complement it. In response, we proposed the Zonified HoP-DTN using Blockchain Sharing of Energy. ZHoP is a 4-step mechanism that includes routing and scheduling mechanisms that make the system more energy efficient. The 4 steps are namely, Pizza Boy Strategy, Friend Zone, High-Low Pigeons and Conditional Pickup. The scheduling scheme that we chose provided an upper limit on the maximum delay possible to a message waiting in the pigeon buffer before take-off. Applying Blockchain further enhanced energy efficiency. We also derived a queueing model for ZHoP which is a 5×5 switch model and a $M/G/5s/\infty/\infty/FIFO$ queueing model according to Kendall's Notation. Accordingly we found out the occupation rate per pigeon, mean queue length, mean waiting time, distribution of the waiting and sojourn time (the queue length and wait time is the one faced by messages while waiting for pigeon allocation), average cost per customer and total server (pigeon) cost from the standard equations. Mathematical verification alone is not enough and so we did an extensive simulation analysis complete with stress tests and sensitivity analysis. On simulation we found that our scheme performs better in almost every aspect than the existing ones and that too in high percentage. The proof of this can be seen from the various plots and more specifically from Tables 3 and 4. In fact apart from very successfully achieving all the objectives, it outperforms the existing HoP-DTN mechanisms in almost all other important network metrics. To certify the above conclusion we have additional results from stress tests which confirm the same. From the sensitivity analysis we conclude that the system is much less sensitive to individual input parameters than other existing schemes which makes ZHoP-DTN a stable system.

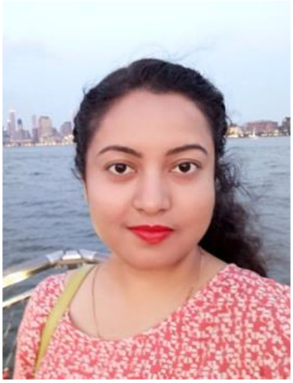
In future we would like to experiment with pragmatic clustering techniques to create zones for situations where the nodes are not uniformly distributed. Future work will also encompass physical deployment of this model, releasing the simulator that we developed for this work custom made for research in HoP-DTN and considering mobile cluster heads.

References

1. Guo, H., Li, J., & Qian, Y. (2010). Hop-dtn:modeling and evaluation of homing-pigeon based delay tolerant networks. *IEEE Transactions on Vehicular Technology, 59*(2), 133.
2. Hui, G. J. L., & Qian, Y. (2009). Modeling and evaluation of homing-pigeon based delay tolerant networks with periodic scheduling. In *Proceedings of the IEEE international conference on communications* (pp. 4988–4992), (IEEE Press, 2009).
3. Project W. (2018). Wizzy digital courier project. http://www.wizzy.org.za/. (Accessed 10 February 2018).
4. Vahdat, A., & Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. Technical report, CS-200006, Duke University.
5. Spyropoulos, T., Psounis, K., & Raghavendra, C.S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings SIGCOMM workshop on Delay-tolerant networking* (pp. 252–259), (ACM, 2005).
6. Zhao, W., & Ammar, M. H. (2003). Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Proceedings of the ninth IEEE workshop on future trends of distributed computing systems* (pp. 308–314). FTDCS 2003. https://doi.org/10.1109/FTDCS.2003.1204352.
7. Zhao, W., Ammar, M., & Zegura, E. (2005). Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proceedings of of the 24th annual joint conference of the IEEE computer and communications societies, INFOCOM* (Vol. 2, pp. 1407–1418), (IEEE Press, 2005).
8. Shah, R. C., Roy, S., Jain, S., & Brunette, W. (2003). Data mules: Modeling a three–tier architecture for sparse sensor networks. In *IEEE SNPA* (pp. 30–41), (IEEE Press, 2003).
9. Medjiah, S., Taleb, T., & Ahmed, T. (2014). Sailing over data mules in delay-tolerant networks. *IEEE Transactions on Wireless Communications, 13*(1), 5. https://doi.org/10.1109/TW.2013.123013.120398.
10. Banerjee, N., Corner, M. D., & Levine, B. N. (2007). An energy-efficient architecture for dtn throwboxes. In *Proceedings of 26th IEEE international conference on computer communications (INFOCOM 2007)* (pp. 776–784).
11. Trullols-Cruces, O., Morillo-Pozo, J., Barcelo-Ordinas, J. M., & Garcia-Vidal, J. (2011). Power saving trade-offs in delay/disruptive tolerant networks. In *Proceedings of 2011 IEEE international symposium on a world of wireless, mobile and multimedia networks* (pp. 1–9), (IEEE, 2011).
12. Choi, J. (2018). Virtual machine placement algorithm for energy saving and reliability of servers in cloud data centers. *Journal of Network and Systems Management*. https://doi.org/10.1007/s10922-018-9462-3.
13. Li, Y., Jiang, Y., Jin, D., Su, L., Zeng, L., & Wu, D. O. (2010). Energy-efficient optimal opportunistic forwarding for delay-tolerant networks. *IEEE Transactions on Vehicular Technology, 59,* 4500.
14. Rodrigues-Silva, D., Costa, A., & Macedo, J. (2012). Energy impact analysis on dtn routing protocols. In *Proceedings of of ACM ExtremeCom* (pp. 1–6), (ACM, 2012).
15. Guo, H., Li, J., & Qian, Y. (2008). Hop: Pigeon-assisted forwarding in partitioned wireless networks. In *Proceedigs of the 3rd international conference on wireless algorithms, systems, and applications* (pp. 72–83), (Springer-Verlag, 2008).
16. Zhou, J., Li, J., & III, L. L. B. (2010). Efficient scheduling of pigeons for a constrained delay tolerant application. *EURASIP Journal on Wireless Communications and Networking*. https://doi.org/10.1155/2010/142921.
17. Hui, G., Jiang, L., Qingyang, H. R., & Yi, Q. (2011). Homing pigeon based messaging: Multiple pigeon assisted delivery in delay tolerant networks. *Wireless Communications and Mobile Computing, 13*(8), 719. https://doi.org/10.1002/wcm.1133.
18. Guo, H., Li, J., Qian, Y., & Tian, Y. (2008). A practical routing strategy in delay tolerant networks using multiple pigeons. In *MILCOM 2008-2008 IEEE military communications conference* (pp. 1–7). https://doi.org/10.1109/MILCOM.2008.4753116.
19. Zhou, J., Li, J., & Mitchell, K. (2010). Adaptive scheduling of message carrying in a pigeon network. *Journal of Ubiquitous Systems and Pervasive Networks, 1*(1), 29.
20. Das, P., Dubey, K., & De, T. (2013). Priority aided scheduling of pigeons in homing-pigeon-based delay tolerant networks. In *2013 3rd IEEE international advance computing conference (IACC)* (pp. 212–217).

21. Das, P., Dubey, K., & De, T. (2012). Threshold triplet incorporated scheduling of storage based pigeons in homing-pigeon-based delay tolerant networks. In *5th international conference on computers and devices for communication (CODEC-2012)* (pp. 1–4).
22. Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., et al. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews, 100,* 143. https://doi.org/10.1016/j.rser.2018.10.014.
23. Contributors, W. (2018). Kendall's notation—wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Kendall (2018). (Accessed 3 January 2018).
24. Adan, I. J., Boxma, O. J., & Resing, J. A. C. (2001). Queueing models with multiple waiting lines. *Queueing Systems, 37,* 65.
25. Adan, I., & Resing, J. (2015). Queueing systems. Technical report, Department of Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands (2015).
26. EduPristine. Sensitivity analysis: Meaning, uses, methods of measurement. https://www.edupristine.com/blog/all-about-sensitivity-analysis (2018). (Accessed 12 March 2018).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Priyanka Das has been a researcher in the Department of Computer Science and Engineering, National Institute of Technology Durgapur. She had also done her M.Tech (Gold Medalist) from the same university. She did her B.Tech from Asansol Engineering College. Her research interests include Delay Tolerant Networking, Optical networks, Cloud Computing among others. She also enjoys dabbling in humanity and social sciences.

Tanmay De is an Associate Professor in the Department of Computer Science and Engineering, National Institute of Technology Durgapur. He did his PhD from the prestigious Indian Institute of Technology Kharagpur, India. After completing his B.Tech from Calcutta University, Kolkata, India, he received his M.E. from Jadavpur University, Kolkata, India. His research interests are Optical Networks, Delay Tolerant Networks, Wireless Sensor Networks and the like. He is a member of IEEE, USA.