

Research Article

Human-Mobility-Based Sensor Context-Aware Routing Protocol for Delay-Tolerant Data Gathering in Multi-Sink Cell-Phone-Based Sensor Networks

M. B. Shah, S. N. Merchant, and U. B. Desai

Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

Correspondence should be addressed to M. B. Shah, mbshah@iitb.ac.in

Received 17 February 2012; Revised 20 June 2012; Accepted 25 July 2012

Academic Editor: Sherali Zeadally

Copyright © 2012 M. B. Shah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ubiquitous use of cell phones encourages development of novel applications with sensors embedded in cell phones. The collection of information generated by these devices is a challenging task considering volatile topologies and energy-based scarce resources. Further, the data delivery to the sink is delay tolerant. Mobility of cell phones is opportunistically exploited for forwarding sensor generated data towards the sink. Human mobility model shows truncated power law distribution of flight length, pause time, and intercontact time. The power law behavior of inter-contact time often discourages routing of data using naive forwarding schemes. This work exploits the flight length and the pause time distributions of human mobility to design a better and efficient routing strategy. We propose a Human-Mobility-based Sensor Context-Aware Routing protocol (HMSCAR), which exploits human mobility patterns to smartly forward data towards the sink basically comprised of wi-fi hot spots or cellular base stations. The simulation results show that HMSCAR significantly outperforms the SCAR, SFR, and GRAD-MOB on the aspects of delivery ratio and time delay. A multi-sink scenario and single-copy replication scheme is assumed.

1. Introduction

Cyber Physical Systems (CPSs) [1, 2] are prospective technologies of the next millennium, which encompasses the vision of merging communication, computing and control systems in such a manner that they become almost invisible to layman's life and still enhance their living quality. CPS, are a paradigm shift in the conventional object-oriented thought process. CPS enables many machine-machine, machine-human, and human-human interactions, both in physical and cyber world. CPS-based services such as traffic control, pollution control, automated highways, automatic chemical processes, and power grid control need extensive monitoring in real world. In order to capture physical reality into decision making and controlling aspect, CPS needs to interact with Wireless Sensor Networks (WSNs). However, lack of ubiquitousness of WSN prevents penetration of CPS systems. In such a scenario, cell phones that are now equipped with sensory functionalities such as Global Positioning System (GPS), audio processing, camera, pollution sensors, and

accelerometers can be exploited. With the recent transition from the feature to the smart phone, the mobile phone has now extensive sensing capabilities and can work as a mobile sensing device [3]. Moreover, the number of smart phones has been rapidly increasing making cell phones an ideal ubiquitous platform.

Mobile sensing with smart phones has been explored extensively [4–9]. The pollution monitoring application using Cell-Phone-based Sensor Network (CPSN) developed in [10] uses short-range communication outlets such as Wi-Fi or Bluetooth. In this paper, we consider the same CPSN architecture as of [10] shown in Figure 1.

CPSN has many challenges such as volatile topology, limited buffer space and loose connectivity with neighboring nodes. Data gathering protocols used for traditional WSN are ineffective due to aforementioned issues. Albeit the cell phone user's mobility, enables a CPSN node to come in contact with other node and create opportunities to encounter any useful forwarder which can get their messages closer to its intended destination even in the absence of

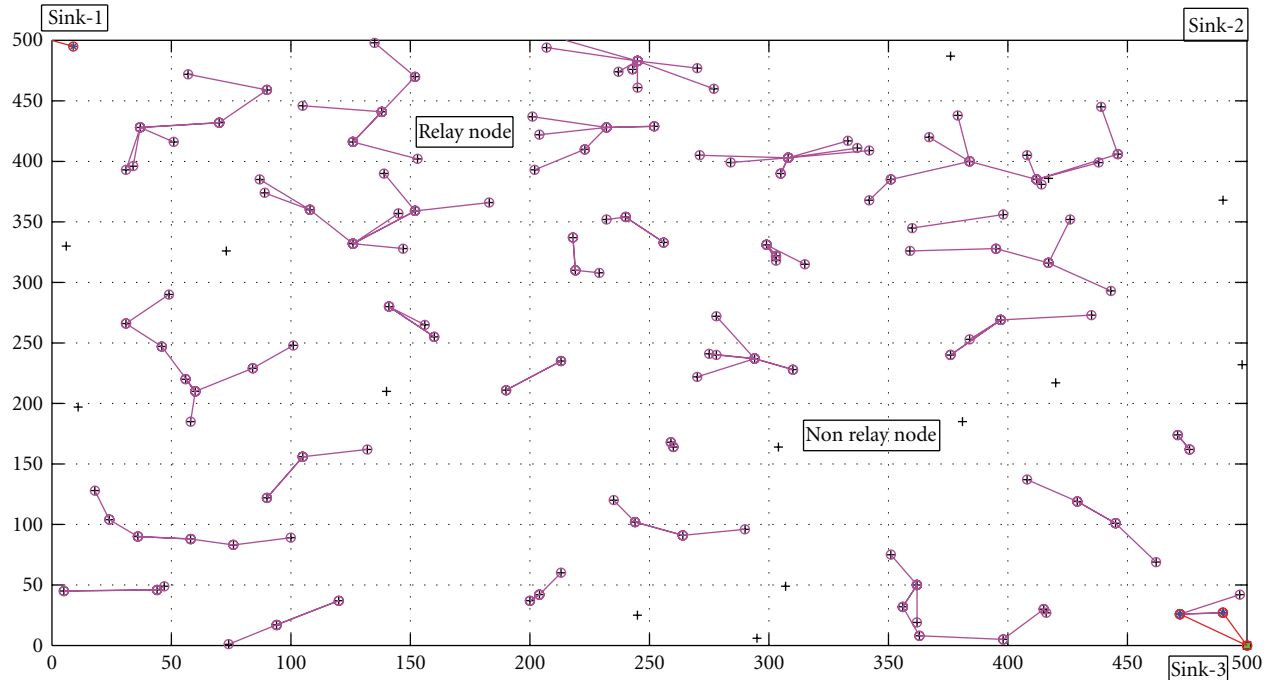


FIGURE 1: Cell-phone-based sensor network.

an end-to-end path [11]. CPSN can exploit these human interactions opportunistically to route the data to the sink or Base Station (BS). We assume CPSN as a type of delay-tolerant, intermittently connected, or opportunistic network [12–14] where end-end path is not available.

The common approach for data gathering in opportunistic sensor networks is directed transmission, where data is delivered to the sink when it is in its direct proximity. To be effective this scheme needs multiple sinks [15]. Another is flooding where messages are forwarded in multiple copies hoping to reach at the sink as early as possible. The third scheme is delay and fault-tolerant data delivery scheme (DFT-MSN) [16], where selected nodes are used as a forwarder of data messages. For CPSN environment, it is preferable to use this approach as energy consumption needs to be minimized.

The opportunistic data forwarding approaches [17, 18] identifies certain lead nodes that act as the relay point of communication between nodes and to be effective, they need good connectivity. These routing protocols select their relay nodes in such a way that the data packets are delivered using minimum hops to the destination. Connection and disconnection with destination node or sink, due to mobile nature of cell phone users, decide the protocol's performance. The mobility of cell phone users cannot be controlled, so the need is to select or predict an appropriate relay node, otherwise the overhead of forwarding data to wrong relay nodes will result in high energy expenditure and negate the benefits of opportunism.

Routing is an NP-complete problem, and a significant amount of research focuses on optimal methods for routing nodes in delay-tolerant Networks (DTNs) [19–24]. This

DTN network scenario is of unicast transmission where message data are sent between pair of source and destination node, while the CPSN scenario is of convergecast transmission where all message data has common destination. Compared with DTN, new opportunities exist in CPSN for relay selection due to increased node density. Furthermore, the human mobility patterns show specific human walk characteristics; namely, truncated power law distributions of flight length and pause time [25]. Given this mobility pattern, it is possible to relay or route more intelligently and design a better routing method for cell phone environments.

Several methods were proposed for forwarding data in mobile sensor networks. Routing approach named Sensor Context Aware Routing (SCAR) [26, 27] uses prediction technique over the context of a mobile sensor node (such as previously encountered history with sink nodes, energy level of each node, and connectivity difference with neighboring nodes) to forecast the best relay node from the neighboring nodes. It uses Kalman-filter-based prediction approach. SCAR uses a replication scheme by which the master copy of the message is retained in the network till it reaches the sink. Secondary copies, sent to the relays, are deleted if buffer becomes full.

Recently authors of [28] have proposed the scheme named gradient-based routing approach incorporating node Mobility (GRAD-MOB). Performance is improved by predicting relay node based on their relative movement with respect to sink. The node moving away from sink is not preferred as a relay node.

The approach of Scale Free Routing (SFR) [29] improves performance with the use of ballistic nodes as relay nodes. The ballistic nodes exhibit sudden jumps of flight length

as per TLW mobility model. Use of these nodes as relays, allows message data to be forwarded from the low to high utility area near sink. Our approach is to prevent nodes from forwarding messages in low utility area with a newly defined utility function.

Motivated by the SCAR and the SFR, we propose a Human Mobility Sensor Context-Aware Routing (HMSCAR) protocol tailored for CPSN architecture with multiple sinks. HMSCAR incorporates two new metrics based on human walk characteristics to predict best relays. The new parameters introduced in our algorithm is based on human walk characteristics; namely, truncated power law distributions of the flight length and the pause time. Use of these parameters is justified for cell phone-based wireless sensor networks as cell phone users exhibit human walk characteristics. SCAR assigns weights to mobile nodes based on metric-like change of connectivity, battery level, and collocation with sink. The newly introduced parameters operate on decision of super flight length and super pause time. A given flight length is super flight or super pause if the flight length or pause time is above certain threshold. The proposed algorithm is compared with SCAR, GRAD-MOB, and SFR using greedy prediction approach where node with highest weight value is preferred as the next relay. Moreover, we evaluate the performance of our algorithm with single copy replication scheme assuming single sink and multisink presence. Single copy replication schemes are basis of multicopy replication schemes. Relay node discard packets if its buffer becomes full.

In each sink or base station initiated round of data collection, the selected relays receive data samples from their surrounding members by 802.11-based Request to Send and Clear to Send (RTS-CTS) mechanism and store them in buffers. The stored data is again forwarded to the selected relays in next round of operation. The relay node uses a bloom filter to compress the data into L bits before it forwards to another relay in the path. Nodes that are in direct connection with the sink will finally send their stored data to it. Use of routing table at each node enables forwarding of message data from lowest utility node to highest utility nodes in each round of data collection.

We have presented the related work in Section 2. In Section 3, we discuss the system model. In Section 4, we introduce a definition of Super Pause Time and Super flight length and propose the HMSCAR algorithm. The performance metrics and simulation setup is described in Sections 5 and 6, followed by the results and conclusion in Sections 7 and 8, respectively.

2. Related Work

There are many approaches proposed for routing in Delay Tolerant Networks (DTN). A detailed survey of all these techniques has been done in [23, 24]. Forwarding strategies of DTN are either epidemic [21, 22] or probabilistic [19]. Epidemic protocols use varieties of replication schemes for reducing packet delay in the network at the cost of energy consumption. The probabilistic methods make smart decisions of packet forwarding so that the packet delivery

ratio is increased. It is very difficult to achieve both high delivery ratio and low delivery delay for a given energy and storage constraints [23]. Probabilistic methods leverage on a utility function, which is either destination dependent, destination independent, or hybrid. The SCAR algorithm proposed in [26] is a probabilistic one and based on hybrid utility function. In [30] a content- and context-aware routing protocol (CCBR) for multi-sink scenario is proposed. CCBR provides good delivery ratio with low power consumption. In [31] a history-based routing approach named CHARON is presented, which estimates delivery delay (EDD) at each node and routes the messages towards the sink along decreasing delay gradient. CHARON utilizes EDD as a context parameter, instead of traditional metrics such as relative mobility or sink encounter frequency. It performs better with low-density network scenario. In [32] a DTN protocol called SMITE is proposed for efficient data delivery of data packets to a mobile sink. SMITE outperforms SCAR, DFT-MSN and sidewinder [33] on the aspect of delivery ratio, transmission overhead, and time delay. SMITE uses cluster-based aggregation mechanism coupled with angle-based transmission mechanism to deliver data to the mobile sink.

The authors of [34] propose a mechanism to calculate delivery probability based on node's moving direction, speed, and distance between node and sink. Similar approach called as GRAD-MOB is applied in [28], where in addition to gradient-based routing metric another metric, based on node's moving direction from the sink node, has been added. This algorithm has novelty of incorporating node mobility for DTN networks. All above approaches use either single copy, multi-copy, or hybrid replication scheme. Contrary to this, the authors in [35] has proposed a DTN routing protocol, which uses repetitive contacts and their time sequence. The protocol does not replicate message and hence can work for lower buffer size of mobile nodes.

Human-carried mobile devices have some specific movement patterns in real life. Some of the movement patterns have been studied in [36, 37]. In [11], the effect of human mobility was evaluated on two-hop relaying algorithm, which was proposed in [38]. The authors of [11] have analyzed the performance of two-hop relaying protocol in a network of nodes following power law intercontact time. Result indicates infinite delay if power exponent is greater than two, which is the case of light tail. The delay becomes finite if multicopy replication approach is used.

Authors of [25] have proposed Truncated Levy Walk (TLW) mobility model. The mobility traces generated by TLW with various combinations of levy exponent α and β fit real human walk traces for various outdoor settings. In [29] a Scale-Free Routing (SFR) scheme is proposed based on observation of flight length characteristics of TLW. Some special nodes called ballistic nodes or super nodes are identified and messages are forwarded, if such nodes are available as relays. The SFR claims to achieve lower delivery delay compared to traditional gradient-based utility schemes, which may lead to local minima. SFR approach is generalized and can be applied with any gradient-based scheme. However, authors of [29] have made assumptions

of knowing mobility information of ballistic node, which is a strong one, and have proposed secondary means to predict it. Our work is also based on observation of TLW mobility model. We leverage both pause time and flight length distributions of human walk without using secondary means to predict it. Our proposed scheme HMSCAR is based on properties of heavy-tailed data called declining hazard rate and mass-count disparity [39]. We use a threshold-based approach to predict about super flight length or super pause time. Prediction of super flight length is based on declining hazard rate property of heavy-tailed data and that of super pause time is based on mass-count disparity property of heavy-tailed data.

The use of mobile phone for sensing was advocated in [40, 41]. Authors of [42] have used a layered approach for data monitoring application in multihop cell-phone based Sensor Network (MSCN). We propose a cluster-based data-gathering protocol for MCSN and Single-hop Cell-phone-based Sensor Network (SCSN) [43, 44], where the data gathering schemes are designed for nondelay tolerant monitoring application.

3. System Model

Mobility models emulate the behavior of real life mobile user as closely as possible. Truncated Levy Walk (TLW) mobility model proposed in [25] is based on, about one thousand hours of GPS traces, involving 44 volunteers in various outdoor settings. The TLW model captures movement of the people in various outdoor scenarios such as campus area, theme park, fair, and metropolitan area. As the cell phone users exhibit human walk characteristics, we have used TLW model as the mobility model for performance evaluation of HMSCAR.

3.1. Truncated Levy Walk Mobility Model. Recent findings of human walk characteristics show truncated power law distribution of flights (straight line trip without directional change), pause times, and intercontact times (the time elapsed between two successive contacts of the same pair of nodes). Figure 2 shows mobility traces of TLW mobility model.

- (i) Flight lengths follow a truncated power law

$$P(l) \propto |l|^{-(1+\alpha)}, \quad 0 < l < l_{\max}, \quad (1)$$

where l_{\max} is the maximum flight length.

- (ii) Pause times follow a truncated power law

$$\phi(t) \propto t^{-(1+\beta)}, \quad 0 < t < t_{\max}, \quad (2)$$

where t_{\max} is the maximum pause time.

- (iii) Turning angles follow a uniform distribution.
 (iv) Velocity increases as flight lengths increase.

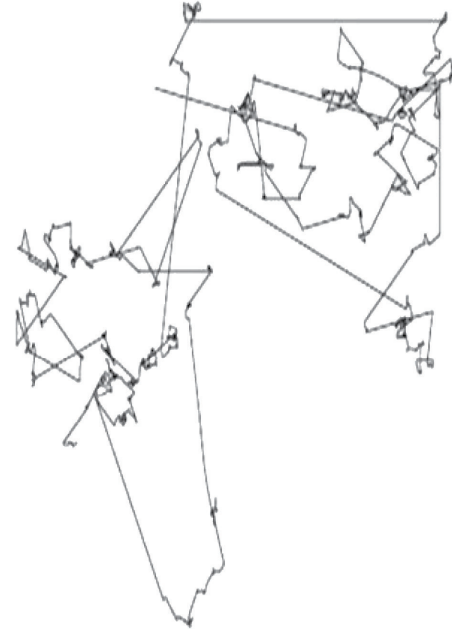


FIGURE 2: Truncated Levy Walk mobility model's waypoints.

3.2. Observations

3.2.1. For Flight Length Distribution

- (i) Power law distribution of flight length implies large number of frequent long flights as compared to Gaussian or an exponential distribution. For humans the duration of moves or the distance moved in one "step" (i.e., flight length) is limited owing to physical constraints. This results in truncated power law distribution for human flight length.
- (ii) Heavy tail of flight length implies the declining hazard rate and is captured in terms of conditional expectation [39]

$$E[X/X > k] \propto k. \quad (3)$$

This is also referred as an expectation paradox [45], which means if the observations of heavy-tailed interarrivals are made, then the longer a node has waited, the longer it should expect to wait. This is contrasted with exponential tails where one always gets to the point where the longer one waits, the less time one should expect to continue waiting.

- (iii) The tail of the flight length distribution may be long or short and is related to the context of the location. Small and/or highly crowded area encourage shorter flights and discourage longer flights; thus the flight length distribution results in a short-tailed distribution.
- (iv) However, with a really large area the truncations will not be excessive. When truncations have less impact on flight lengths, the mobility of the nodes has a stronger power-law tendency [25].

From opportunistic forwarding point of view, node exhibiting large flight length should be chosen as a relay. Based on (3), the concept of “super length parameter” is incorporated in HMSCAR algorithm.

3.2.2. For Pause Time Distribution

- (i) Heavy tail of pause time implies the mass-count disparity

$$\lim_{x \rightarrow \infty} \frac{P[X_1 + \dots + X_n > x]}{P[\max(X_1 + \dots + X_n) > x]} = 1, \quad n \geq 2. \quad (4)$$

This means that when considering collections of observations of a heavy-tailed pause time random variable, the aggregated mass contained in the small observations of pause times is negligible compared to the largest observation in determining the likelihood of large values of the sum. In practice, this means that the majority of mass in a set of observations of pause times is concentrated in a very small subset of the observations. To opportunistically exploit pause time, the node pauses for a longer time duration (corresponds to small subset of the observations) should not be chosen as a relay. Based on (4), a new parameter called “Super pause time” has been incorporated in HMSCAR algorithm to predict node, exhibiting the majority of mass in a set of pause time observations.

- (ii) For small or crowded area, the truncation will be short tailed. High traffic will prevent the walker from halting at one location for a long time.
- (iii) For really large areas, truncation will be heavy tailed or even long tailed.

The main assumptions made for the system are as follows.

- (i) All N nodes are uniformly distributed. The Base Stations (BSs) are located at the edge of the region.
- (ii) Nodes can communicate with neighboring nodes, either using cellular bandwidth or short-range communication outlets such as WI-FI or Bluetooth.
- (iii) Nodes have finite buffer size and operate on limited power supply.

4. Human-Mobility-Based Sensor Context-Aware Routing (HMSCAR) Algorithm

Traditionally the SCAR algorithm is used for Mobile Sensor Networks (MSN). We have modified the SCAR protocol by adding two human walk context metrics called H_u and P_u . The basic mechanism for relay node selection remains the same. The parameters H_u and P_u are based on our understanding of human walk characteristics as discussed in previous section. The calculation of H_u at each node is done with the help of GPS and is based on the decision whether

a given flight length should be considered as a super flight length or not. Similarly the calculation of P_u is based on the decision that whether a recent pause time is the super pause time or not. The definitions of super flight length, super pause time, and corresponding calculation of H_u and P_u are discussed next.

4.1. Flight Length Parameter H_u . The determination of parameter H_u depends on determination of whether a given flight, which a node takes, is a super flight length or not. Hence we will define super flight length next.

4.1.1. Super Flight Length. Given a set U of mobile nodes following the flight length distribution as a truncated power law, then there exists a value ζ for all nodes u of such a subset, that if a flight length is above ζ , then based on (3) all nodes of such a subset again travels that flight length with high probability. Let $S_{f(u)}$ denote the flight length above ζ taken by node u and is defined as super flight length. The actual physical distance corresponds to super flight length is which $S_{\text{len}(u)}$. Then

$$S_{\text{len}(u)} = \alpha' S_{f(u)}, \quad (5)$$

where α' is a constant and is a large percentage of the simulation area. It also depends on geographical conditions and mobility environment but most of the time it is the characteristics of human walk [25]. The $S_{\text{len}(u)}$ will not hold for small simulation areas (approx. 200 meters across) [25]. $S_{\text{count}(u)}$ in (6) is incremented every time, when a node makes a flight of length more than $S_{\text{len}(u)}$. The parameter H_u in (6) is directly proportional to $S_{\text{count}(u)}$

$$H_{(u)} = \frac{1}{1 + e^{-2(S_{\text{count}(u)}-1)}}, \quad \text{if } S_{\text{count}(u)} > 0, \quad (6)$$

$$H_{(u)} = 0, \quad \text{if } S_{\text{count}(u)} = 0.$$

A plot of parameter H_u is shown in Figure 3. The parameter value rises from initial value 0 to 1 as nodes take super lengths.

4.2. Pause Time Parameter P_u . The determination of parameter P_u depends on determination of whether a given pause is a super pause or not. Hence we will define super pause time next.

4.2.1. Super Pause Time. Given a set U of mobile nodes following the pause time distribution as a truncated power law, then there exists a value ξ for all nodes u of such a subset, that if the pause time is above ξ , then based on (4), that subset of node u pauses much more than ξ . We call all those pause above ξ as super pause $P_{s(u)}$ of node u . The actual physical time corresponding to super pause time is $P_{\text{stime}(u)}$. Then

$$P_{\text{stime}(u)} = \beta' P_{s(u)}, \quad (7)$$

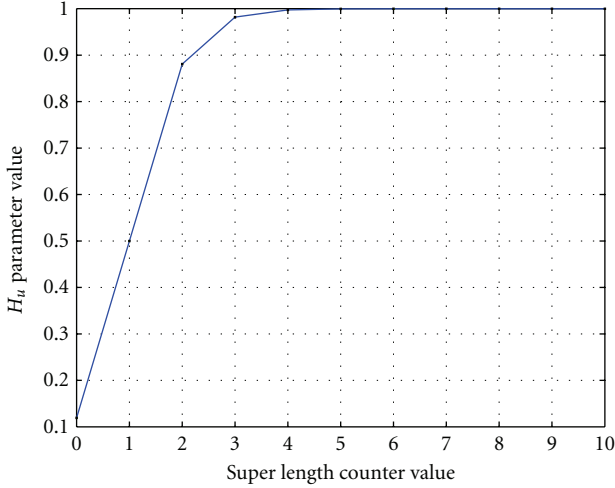
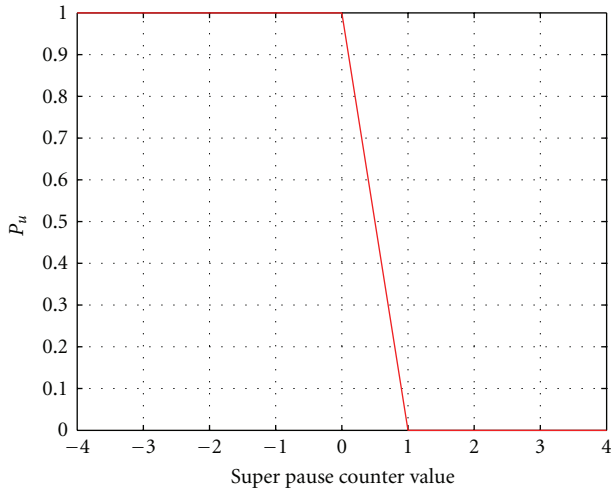
FIGURE 3: H_u parameter with respect to super length count value.

FIGURE 4: Count function with respect to super pause counter.

where β' is again a constant. Super pause counter $P_{\text{scount}(u)}$ in (8) is changed from 0 to 1 if a node pauses for more than $P_{\text{stime}(u)}$ time.

$$\begin{aligned} P_{(u)} &= 0, & \text{if } P_{\text{scount}(u)} > 0, \\ P_{(u)} &= 1, & \text{if } P_{\text{scount}(u)} < 0, \end{aligned} \quad (8)$$

A plot of parameter P_u is shown in Figure 4. The difference in (6) and (8) is due to the complementary nature of human mobility context parameters and also due to the fact that the distribution for pause time is less emphatic than that of flight length. The HMSCAR algorithm consists of four different phases and for its implementation clock synchronization among the nodes are required. The different phases of operation are as follows.

4.2.2. Relay Weight Calculation. The calculation of the delivery probability or relay weight is local, and it does not involve any distributed computation. Nodes exchange

information about their current delivery probability and available buffer space, periodically with their neighboring node. This also includes routing information.

- (1) At the start of data collection round, each node finds all its neighbors (within transmission range R_{tx}). Each node periodically broadcasts NBR_{dis} (Neighbor Discovery) message containing its own ID.
- (2) After a certain time $T_{\text{bootstrap}}$, each node receives all NBR_{dis} message from its neighbors. Upon receiving NBR_{dis} message from neighbors, each node can estimate the change degree of connectivity by Cdc_u by

$$\text{Cdc}_{(u)} = \frac{|\text{Nu}_{t-1} \cup \text{Nu}_t| - |\text{Nu}_{t-1} \cap \text{Nu}_t|}{|\text{Nu}_{t-1} \cup \text{Nu}_t|}, \quad (9)$$

where Nu_t is the number of nodes in reach of the sensors u at time t and Nu_{t-1} is the number of nodes in reach of the sensors at time $t - 1$.

The parameter Cdc_u corresponds to the number of nodes that have transitioned from the in-reach to out-of-reach status or vice versa in the time interval $[t - 1, t]$, normalized by dividing it for the total number of nodes met in the same time interval.

- (1) Each node calculates the remaining battery parameter Bat_u . The value 1 corresponds to full battery and 0 corresponds to an empty battery.
- (2) Each node u summarizes its history of collocation with a sink with parameter Colloc_u where

$$\text{Colloc}_{(u)} = \frac{1}{d}, \quad (10)$$

where d is number of hops from the closest sink. Colloc_u parameter has decreasing value as the distance from a sink increases. If a path does not exist between the sink and the host, its value is set to 0. This information is extracted from the routing table, which is periodically updated depending upon, routing message exchange done with neighboring nodes.

- (3) Each node finds H_u as per (6). Each node finds its super length by estimating the distance from received signal strength in some time interval.
- (4) Each node finds P_u as per (8).
- (5) Each host calculates its delivery probability locally, based on observations related to the various context attributes. Our aim is the optimization of the bundle delivery process. To solve this problem, we apply the so-called weights method [26]. The values of these weights are same for every node.

All the nodes calculate their weight value W_u given by

$$w_u = w_1 \text{Cdc}_u + w_2 \text{Bat}_u + w_3 \text{Colloc}_u + w_4 H_u + w_5 P_u, \quad (11)$$

where $w_1, w_2, w_3, w_4,$ and w_5 are arbitrarily chosen weighing factors satisfying $\sum_{i=1}^5 w_i = 1$. The higher the weight value (W_u), the more probable that the node will be a relay node. The BS transmits information about the maximum value of the above parameters attained in the network periodically to all the nodes. All the nodes normalize their parameters with respect to this value.

- (1) A timer is used to transmit “ WT_{declare} ” routing beacon message at a certain time interval, containing the node id, routing table, current delivery probability, and buffer size of the node. This procedure terminates after T_{stop} .

4.2.3. Data Transmission

- (9) After T_{stop} time each node receives “ WT_{declare} ” message from neighboring nodes. Another timer T_1 is used to send data messages periodically. For single copy replication scheme, the compressed data will be sent to the chosen relay node with the highest weight value. If no such neighbor is available, then the message is stored in the buffer and transmitted later. Each message is uniquely identified by the host name and a message number, generated using a counter that is incremented by one for each message sent.

4.2.4. Data Reception

- (10) The nodes are also waiting for incoming data. If it is a routing message, then it is processed and corresponding routing table entry is updated. If it is a data message, then an ACK is sent and the message is stored in the buffer. Messages are not deleted from the buffer unless they are acknowledged by some relay node or until the buffer becomes full.

4.2.5. Data Collection Phase at Sinks

- (11) Nodes will send their entire data to the sink when they arrive within the transmission range of the sink. Sink aggregates the data packets and sends them to the data repository through a wired network.

5. Performance Metrics

We have evaluated the performance of opportunistic routing algorithm based on received data messages or data delivery ratio and message delay. Moreover, the energy consumption and efficiency of nodes are also evaluated.

- (1) Number of Received Packets: it is a measure of the reliability, effectiveness, and efficiency of routing protocols. We mainly consider packet drop only due to nonavailability of end-end path. Other sources of packet drop due to collision, queue overflow, channel unavailability, and so forth are ignored.
- (2) Delay: it is the time between packet generation and packet reception at the sink node or the BS.

- (3) Energy Consumption: our energy model for the CPSN nodes is based on the first-order radio model described in [46]. A sensor consumes $E_{\text{elec}} = 50$ nJ/bit to run the transmitter or receiver circuitry and $E_{\text{amp}} = 100$ pJ/bit/m² for the transmitter amplifier. Thus, the energy consumed by a sensor i in receiving a k -bit data packet is given by

$$E_{Rx} = (E_{\text{elec}} \cdot k) \quad (12)$$

while the energy consumed in transmitting a data packet to sensor j is given by

$$E_{Tx} = E_{\text{elec}} \cdot k + E_{\text{amp}} \cdot R_{tx}^2 \cdot k, \quad (13)$$

where “ R_{tx} ” is the transmission range. One round of operation will consume E_d units of energy due to routing message exchange, where

$$E_d = \sum_{i=1}^N (E_{\text{elec}} \cdot k') + (E_{\text{amp}} \cdot R_{tx})^2 \cdot k', \quad (14)$$

k' is size of “Hello” packet.

- (4) Network Lifetime (Energy Efficiency): the common definitions include the time until the first or the last node in the network depletes its energy. For CPSN, it is defined as the time until an α percent of sensors consume β percent of initial energy, where α is taken as 50 percent and beta is taken as 10 percent.

These parameters are studied for varying buffer size, transmission range, node density, and mobility patterns for single copy replication scheme.

6. Simulation Setup

We measured the performance of our proposed HMSCAR algorithm with SCAR, SFR, and GRAD-MOB on a MATLAB simulator developed to compare the performance of the algorithms. The simulator only simulates the core network characteristics, such as the nodes’ connectivity and hello and data message transmissions as these are integral to the understanding of algorithms’ behavior. Other details, such as the MAC level protocol, signal propagation, collisions, and realistic channel conditions, are ignored as they have lesser effects. All the algorithms are compared within the same simulation environment making the conclusions unbiased. Further we also assume that there are no obstacles that could limit node mobility or signal propagation.

Due to the importance of connectivity, the weight w_1 associated with Cdc_u was chosen high. The weight values for SCAR algorithm are $w_1 = 0.6, w_2 = 0.2,$ and $w_3 = 0.2$. The weight values for HMSCAR algorithm are $w_1 = 0.2, w_2 = 0.2, w_3 = 0.2, w_4 = 0.2,$ and $w_5 = 0.2$. HMSCAR-sln is a variant of HMSCAR algorithm having weight value 0 associated with the pause time parameter. HMSCAR-sln is also SFR, when gradient scheme according to SCAR is considered. The weight value for HMSCAR-sln or SFR is $w_1 = 0.2, w_2 = 0.2, w_3 = 0.2, w_4 = 0.4,$ and $w_5 = 0.0$. While HMSCAR-sps is

TABLE 1: Default simulation parameters.

Parameters	Values
Sensor field	500 × 500 rectangle area
Number of nodes (N)	150
Transmission range	50
Buffer or queue size	[300, 450, 600, 750]
BS location	(250,250), (0,500), (500,0)
Routing retrans. Interval	60 seconds
Message retrans. Interval	60 seconds
Sampling time	1 minute
Total simulation time	60 minutes
Contextual constant α'	425 meters
Contextual constant β'	20 minutes
Packet header size	25 bytes
Initial energy	2 J/battery
Data packet size	500 bytes
Routing packet size	25 bytes
Scale parameter for flight length	10
Scale parameter for pause time	1

an another variant of HMSCAR, which has weight value 0 associated with flight length parameter. The weight value for HMSCAR-sps is $w_1 = 0.2$, $w_2 = 0.2$, $w_3 = 0.2$, $w_4 = 0.0$, and $w_5 = 0.4$. The GRAD-MOB algorithm is the one which has a different metric, depending upon the decision of whether a node is moving towards the sink or away from the sink. Based on this decision, a value 1 or 0 is assigned, respectively. The first three metric are same as that of SCAR algorithm. The weight value for GRAD-MOB is $w_1 = 0.4$, $w_2 = 0.2$, $w_3 = 0.2$, and $w_4 = 0.2$. These values are used throughout the first four and sixth simulation scenarios. Montecarlo simulations are performed for calculating various performance metrics. We apply SCAR, SFR, GRAD-MOB, HMSCAR, and HMSCAR-sps to the TLW mobility model. Table 1 presents a summary of the already listed simulation parameters. These default parameters are used in all simulations, except where otherwise noted.

- (1) Simulation-1: a network of 150 nodes, which move in a rectangular area of 500 m × 500 m, are considered. This is a high density scenario. Simulation was carried out for a total duration of 3600 seconds, and buffer sizes are varied from 300 to 750. Monte Carlo simulations have been carried out for 10 random seeds and 50 meter transmission range. Multicopy replication scheme is implemented, and performance of SCAR, HMSCAR, SFR, and GRAD-MOB is compared.
- (2) Simulation-2: simulation scenario similar to part-1 is considered but instead of multicopy, the single copy replication scheme is implemented and performance of SCAR, HMSCAR, SFR, and GRAD-MOB is compared.
- (3) Simulation-3: rectangular area of 500 m × 500 m is considered. The node density is varied from 75 to 150

with fix buffer size of 1200. Simulation was carried out for a total duration of 3600 seconds. Monte Carlo simulations have been carried out for 10 random seeds and node communicate with transmission range of 50 meter. Single-copy replication scheme is implemented and performance of SCAR, HMSCAR, SFR, and GRAD-MOB is compared.

- (4) Simulation-4: here we vary the transmission range of the nodes from 20 meter to 50 meter for rectangular area of 500 m × 500 m and for 150 nodes having buffer size of 900. The simulation duration was 3600 seconds. Single-copy replication scheme is implemented and performance of SCAR, HMSCAR, SFR, and GRAD-MOB is compared.
- (5) Simulation-5: here we vary the weight values of HMSCAR algorithm to see its effect on performance metrics. The transmission range is kept at 50 meter, and 150 nodes moving in a rectangular area of 500 m × 500 m according to TLW mobility pattern are chosen. Buffer size is varied from 300 to 750 for total simulation time of 3600 seconds. Single-copy replication scheme is implemented with six different combinations of weight values. As a first part, the weight values of Cdc_u and P_u are relatively changed while other weight values are not changed. The weight combinations are $wt_1 = [0.1 \ 0.2 \ 0.0 \ 0.1 \ 0.6]$, $wt_2 = [0.3 \ 0.2 \ 0.0 \ 0.1 \ 0.4]$, and $wt_3 = [0.5 \ 0.2 \ 0.0 \ 0.1 \ 0.2]$. In second part, the weight values of $Colloc_u$ and P_u are relatively changed while other weights are unchanged. These weight combinations are $wt_4 = [0.1 \ 0.2 \ 0.1 \ 0.1 \ 0.6]$, $wt_5 = [0.1 \ 0.2 \ 0.3 \ 0.1 \ 0.4]$, and $wt_6 = [0.1 \ 0.2 \ 0.5 \ 0.1 \ 0.2]$.
- (6) Simulation-6: authors of [25] has shown that the realistic human walk traces can be well fitted using TLW mobility model with various combination of α and β . Here we take different combination of β and α for 150 nodes having fixed buffer size of 900 with 3600 seconds of simulation time. Single-copy replication scheme is implemented and performance of SCAR, HMSCAR, SFR, and GRAD-MOB is compared for single-sink and multisink case.

7. Results and Discussions

7.1. Simulation-1. Figure 5 shows number of data packets received for SCAR, HMSCAR, SFR, GRAD-MOB, and HMSCAR-sps algorithms for single-sink and multisink case. The HMSCAR and HMSCAR-sps algorithm performs best among all the five algorithms. Data delivery is improved by 4% compared to SCAR and GRAD-MOB and 3% compared to SFR algorithm for multisink case. Improvement is negligible for single-sink case.

Figure 6 shows the number of packets received for SCAR, HMSCAR, SFR, GRAD-MOB, and HMSCAR-sps algorithms for various buffer sizes. The number of packets received is more for HMSCAR-sps and HMSCAR algorithms compared to other algorithms. HMSCAR has 5% more

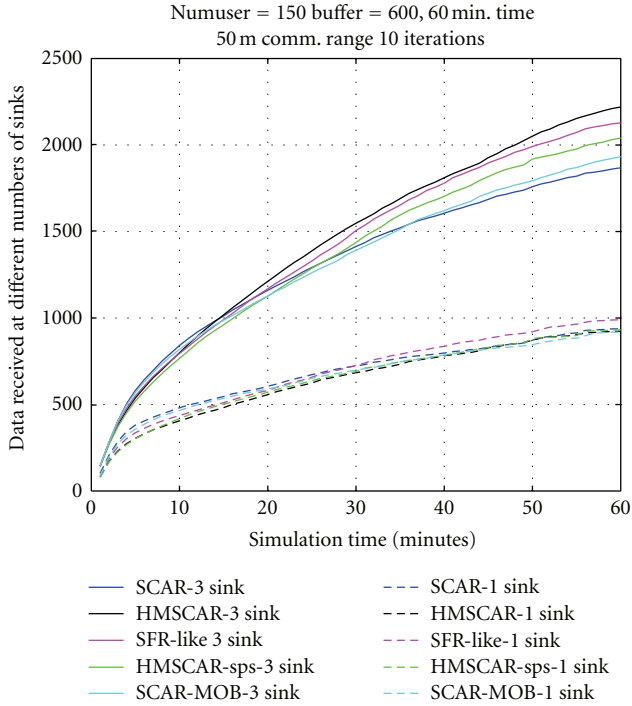


FIGURE 5: Data received at various sinks versus simulation time.

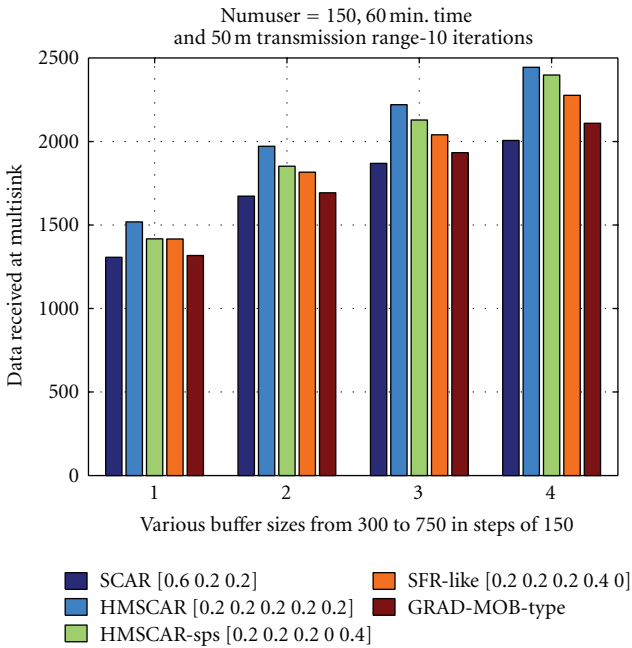


FIGURE 6: Data received at sinks versus buffer size.

message delivery ratio compared to SCAR for buffer size of 750 and has 4% more for buffer size of 600.

Figure 7 shows the number of packets received for various algorithms at single sink. The number of packets received is more for HMSCAR-sps algorithm for all buffer sizes. SCAR and GRAD-MOB have the least amount of received packets.

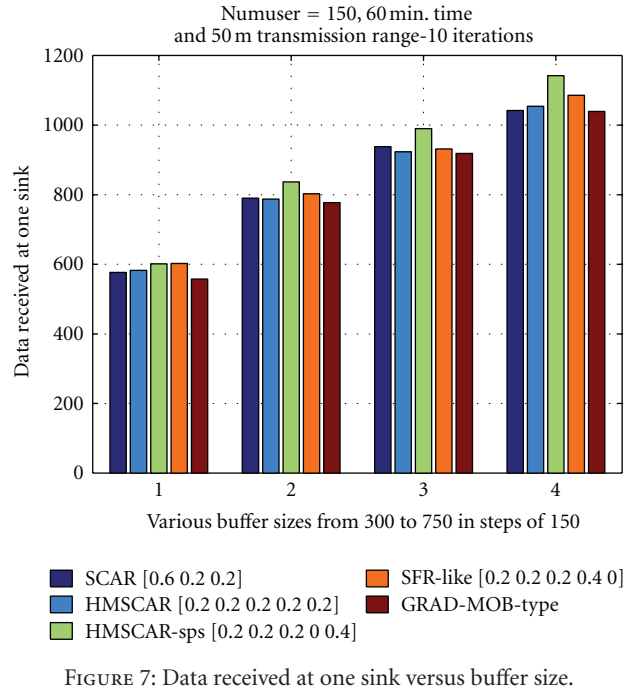


FIGURE 7: Data received at one sink versus buffer size.

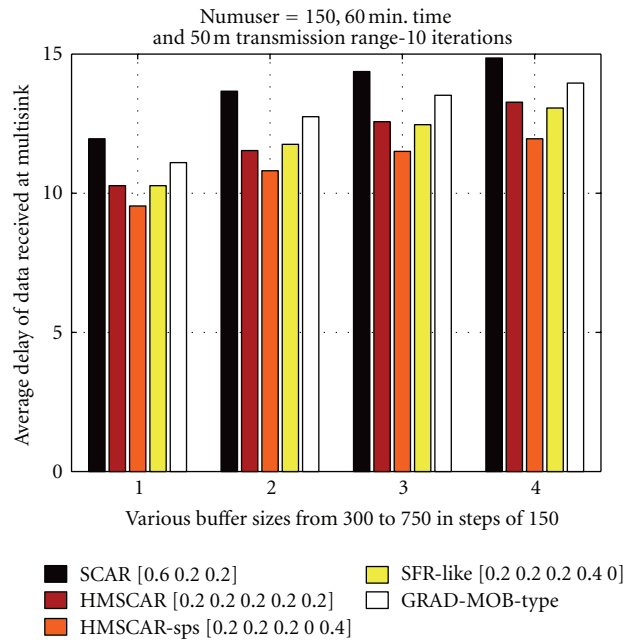


FIGURE 8: Average delay of packets at multisinks versus buffer size.

Figure 8 shows average data delivery delay of the packets for SCAR, HMSCAR, SFR, GRAD-MOB, and HMSCAR-sps algorithms. Average data delivery delay is least for HMSCAR-sps algorithm as compared to other four. SCAR performs worst across various buffer sizes. HMSCAR-sps's performance decreases at higher buffer sizes. This is due to multicopy approach, which stores redundant data in the buffer.

Figure 9 shows average data delivery delay of the packets for SCAR, HMSCAR, HMSCAR-sln, and HMSCAR-sps

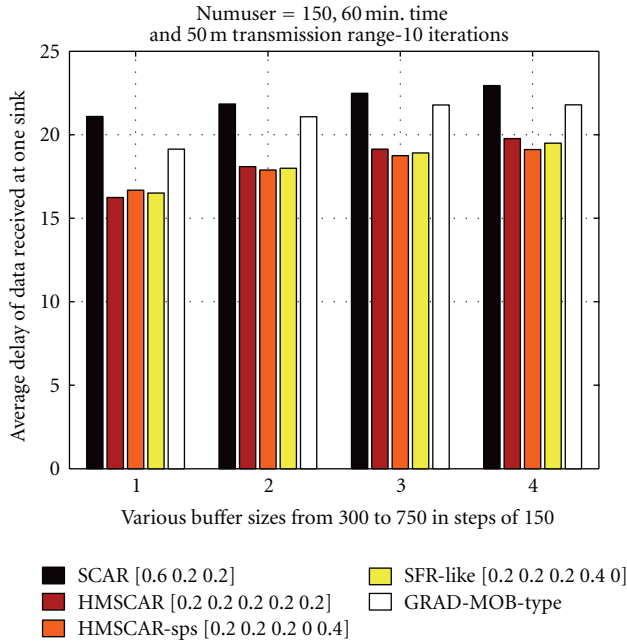


FIGURE 9: Average delay of packets at single sink versus buffer size.

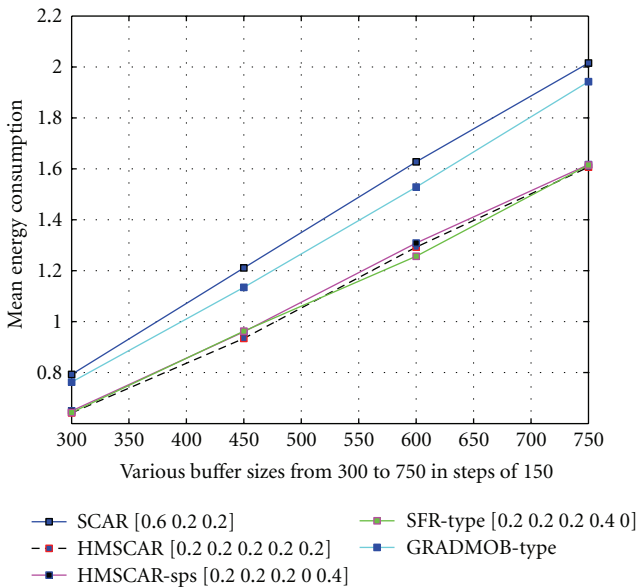


FIGURE 10: Average energy consumption of nodes versus buffer size.

algorithms. Average data delivery delay is least for HMSCAR, SFR, and HMSCAR-sps algorithms as compared to other two. HMSCAR performs the best across various buffer sizes. SCAR performs the worst among all algorithms.

Figure 10 shows energy consumption of all the nodes for various algorithms. Average energy consumption increases with increase in buffer size. HMSCAR and its variants have minimum energy consumption compared to SCAR and GRAD-MOB.

Figure 11 shows energy efficiency for various algorithms. All the algorithms perform more or less the same. No

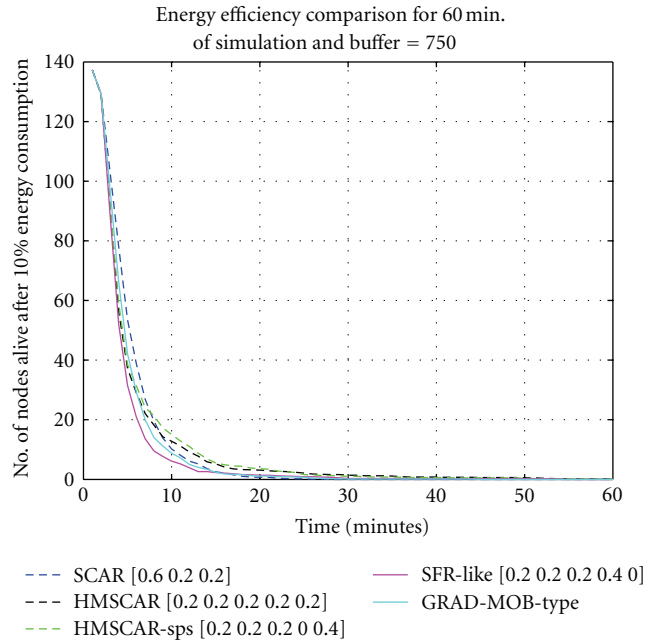


FIGURE 11: Number of nodes alive after 10% energy consumption.

algorithm has exceptional energy efficiency compared to others. Result shows that HMSCAR performs some what better compared to SCAR and others.

7.2. Simulation-2. Figure 12 shows the number of packets received at the sink for single-copy replication scheme of various algorithms, with different number of sinks. HMSCAR outperforms SCAR almost 22%, and 15% to GRAD-MOB for fixed buffer size of 600.

Figure 13 shows the number of packets received for SCAR, HMSCAR, HMSCAR-sln, and HMSCAR-sps algorithms. The number of packets received is more for HMSCAR algorithm. Performance of all the algorithms increases as buffer size is increased. SCAR has the least amount of received packets amongst all the algorithms.

Figure 14 shows the number of packets received for SCAR, HMSCAR, SFR, GRAD-MOB, and HMSCAR-sps algorithms. The number of packets received is more for HMSCAR-sps algorithm for various buffer sizes. SCAR has least amount of packets received among all the algorithms. Performance of all the algorithms is improved with an increase in the buffer size.

Figure 15 shows average data delivery delay of the packets of various algorithms for single-copy replication case. Average data delivery delay of HMSCAR-sps algorithm is the least. HMSCAR and SFR have poor performance for various buffer sizes. Increasing buffer size also increases data delivery delay. Increasing buffer size enables more messages to be delivered, which were dropped earlier due to insufficient buffer space, but now they are able to reside in the buffer long enough to be delivered to the sink. This incurs a larger delay for these messages and, therefore, increases the average data delivery delay.

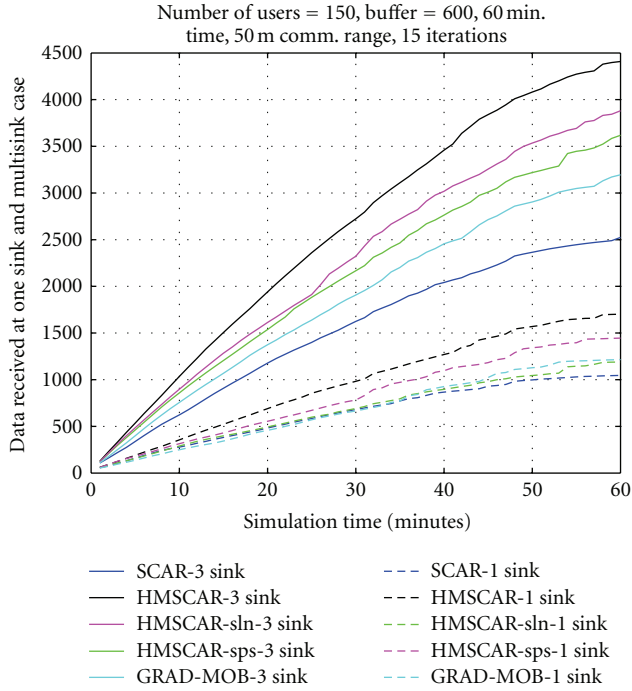


FIGURE 12: Data received versus simulation time for single copy case.

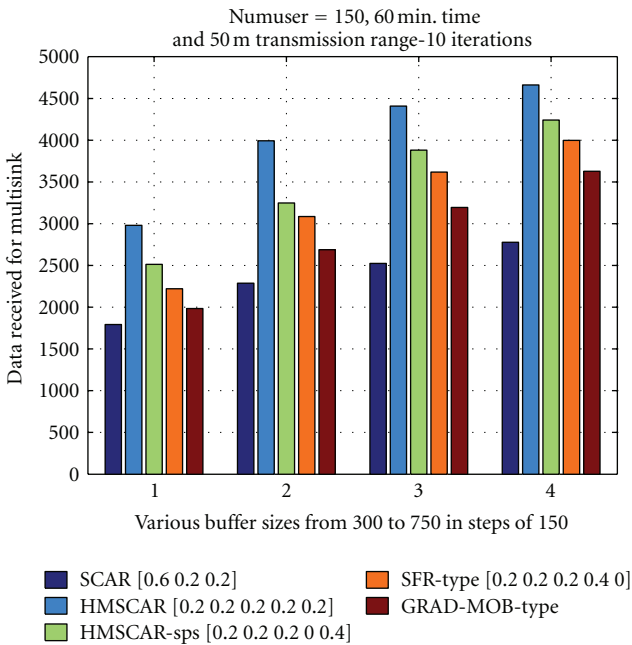


FIGURE 13: Data received at multisinks versus buffer size.

Figure 16 shows average data delivery delay of the packets for SCAR, HMSCAR, SFR, GRAD-MOB, and HMSCAR-sps algorithms for single-copy, single-sink case. The SFR performs worst and so is HMSCAR. Performance of SCAR is the best for various buffer size.

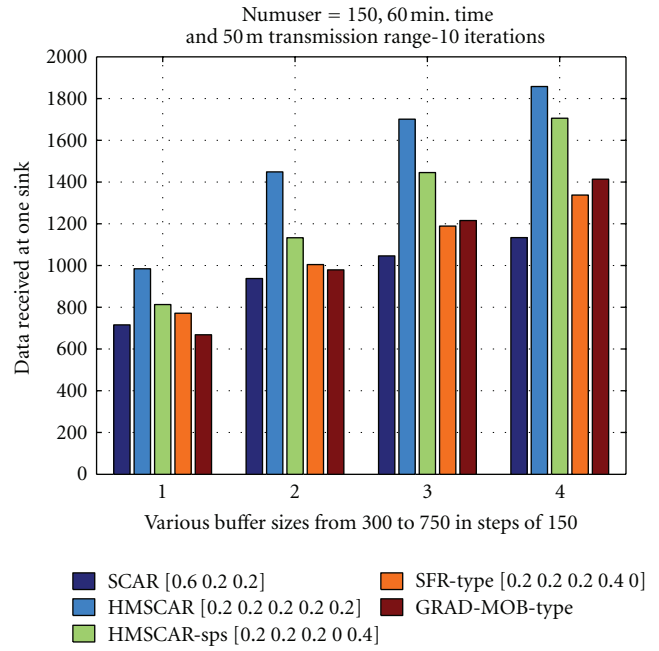


FIGURE 14: Data received at one of the sink versus buffer size.

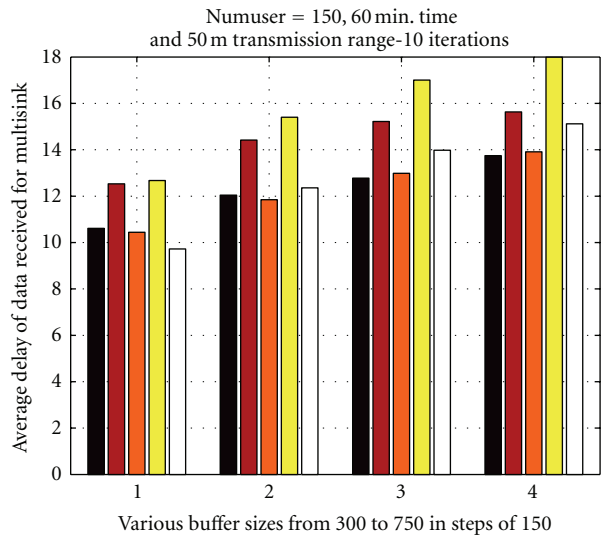


FIGURE 15: Average delay of packets received for various buffer size.

Figure 17 shows energy consumption of data received at all the sinks for various algorithms. The HMSCAR-sln algorithm consumes more energy while HMSCAR-sps consumes least. For HMSCAR-sln, algorithm the nodes making super flight length are preferred as a relay, so they carry most of the load of the network. While HMSCAR-sps algorithm prevents certain nodes as a relay node and thereby improves data delivery by not carrying network load and therefore its energy consumption is minimum at the expense of data delivery delay.

Figure 18 shows energy efficiency for various algorithms. HMSCAR is energy efficient compared to SCAR and GRAD-MOB. This is obvious as SFR and HMSCAR prefer nodes

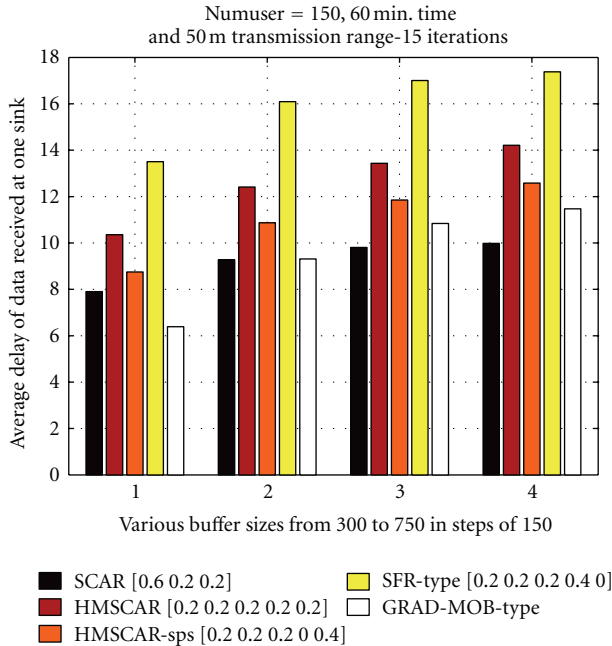


FIGURE 16: Average delay of packets received for various buffer size.

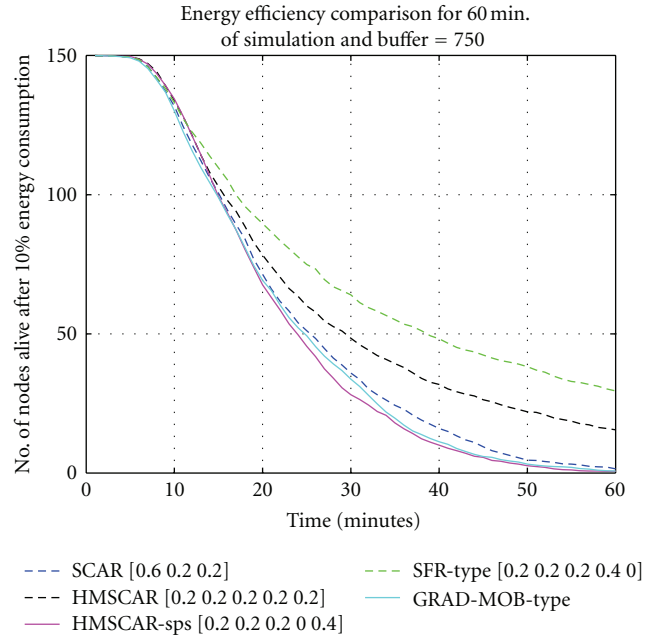


FIGURE 18: Number of nodes alive after 10% energy consumption.

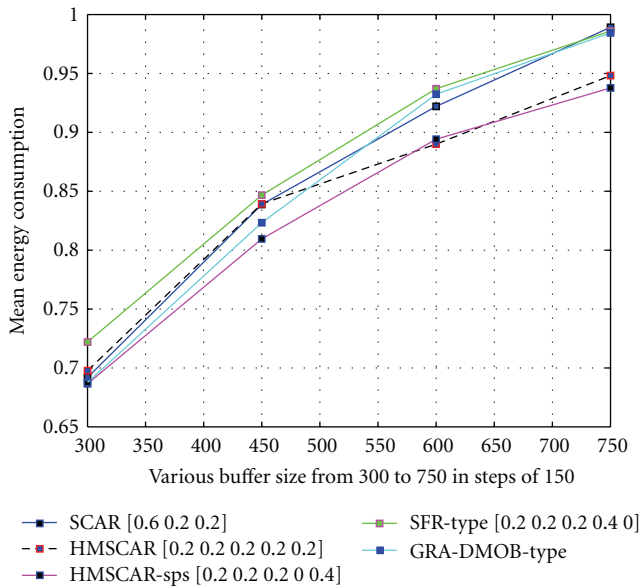


FIGURE 17: Average energy consumption of nodes for various buffer sizes.

with long jump as relay nodes and most of the load is carried by these nodes, so these nodes will deplete their energy very soon. So instead of 10% node death, if first node death is considered as energy efficiency, then HMSCAR performs the best.

7.3. Simulation-3. Figure 19 shows the number of packets received of various algorithms for various node densities with single-copy replication scheme. HMSCAR has the highest amount of packets received among all the algorithms

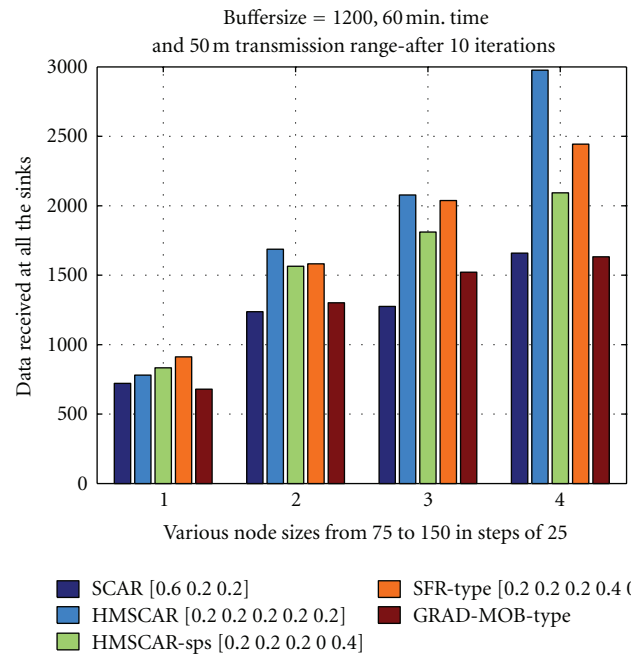


FIGURE 19: Data received at multisinks versus various number of nodes.

at higher node densities. The number of packets received for HMSCAR is 15% more compared to SCAR for 150 number of nodes and higher by 5% at 100 number of nodes. Performance of all the algorithms improves as node density is increased.

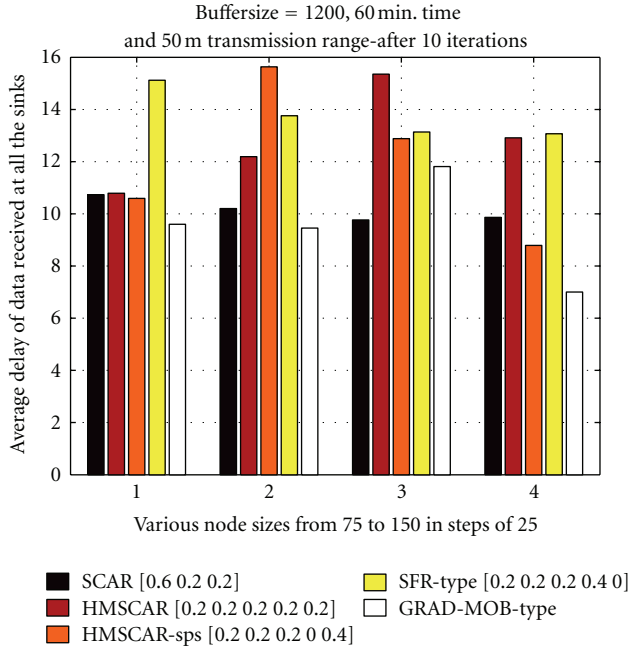


FIGURE 20: Average delay of packets received for various no. of nodes.

Figure 20 shows average data delivery delay of the packets for SCAR, GRAD-MOB, HMSCAR, SFR, and HMSCAR-sps algorithms with various node densities with the single-copy replication scheme. Average data delivery delay is worst for HMSCAR and its variants at lower node densities. Performance of HMSCAR is improved as node density is increased. It is important to note that, although the HMSCAR performs poor, than GRAD-MOB at node density of 150, the delivery ratio of HMSCAR is 15% more compared to GRAD-MOB at this node density.

Figure 21 shows the average energy consumption of all the nodes for various algorithms. Mean energy consumption is reduced as node density increases. This is due to the decrease in internode distance, that a packet has to travel with relative increase in node density. HMSCAR has least amount of energy consumption among all the algorithms for various node densities.

7.4. *Simulation-4.* Figure 22 shows the number of packets received of various algorithms for various transmission range with single-copy replication case. The number of packets received is more for HMSCAR-sps algorithm for higher transmission range. HMSCAR-sps has highest packet received amongst all the algorithms for transmission range above 30. At low transmission range, packet delivery is very poor. This is due to unavailability of super nodes as relays at lower transmission range.

Figure 23 shows average data delivery delay of various algorithms for various transmission range with single copy replication case. Average data delivery delay is worst for SFR and HMSCAR algorithms when compared to other three at

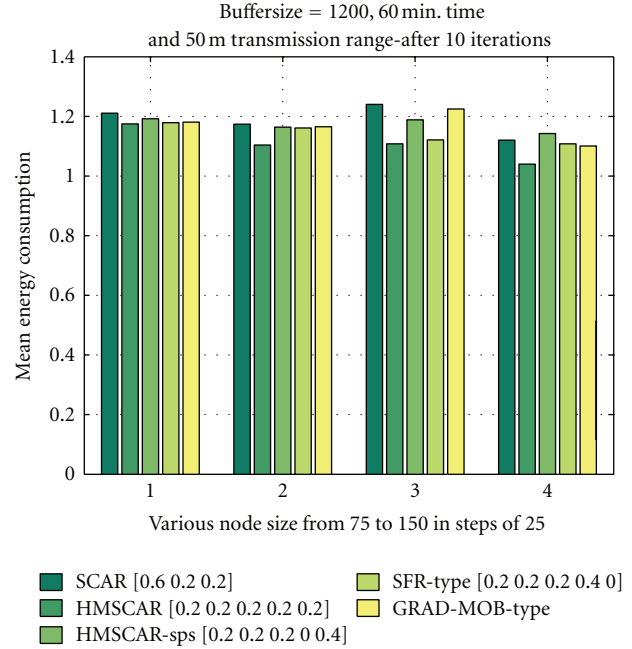


FIGURE 21: Average energy consumption of nodes for various no. of nodes.

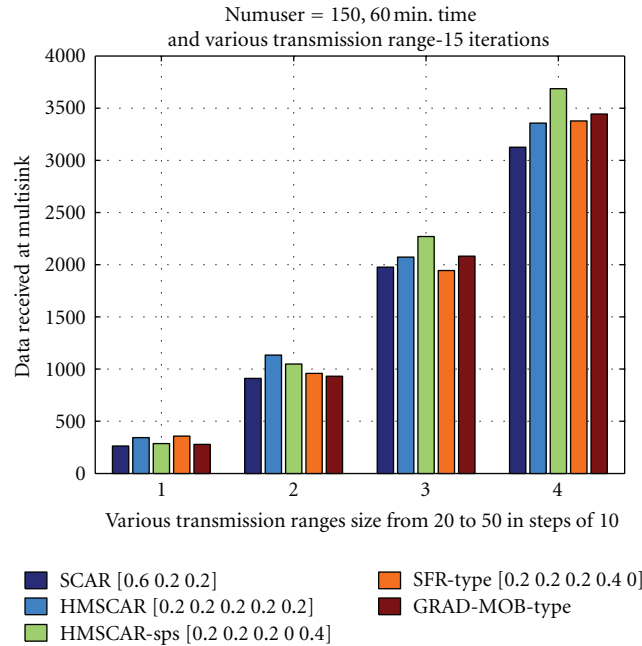


FIGURE 22: Data received at multi sinks versus various trans. Range.

lower transmission range. Performance is almost equal at higher transmission range.

Figure 24 shows energy consumption of data received at all the sinks for various algorithms. The SFR algorithm consumes more power while HMSCAR-sps consumes the least. Energy consumption increases for change in transmission range from 20 meter to 40 meter. For transmission range of 50 meter, the energy consumption is reduced. Connectivity

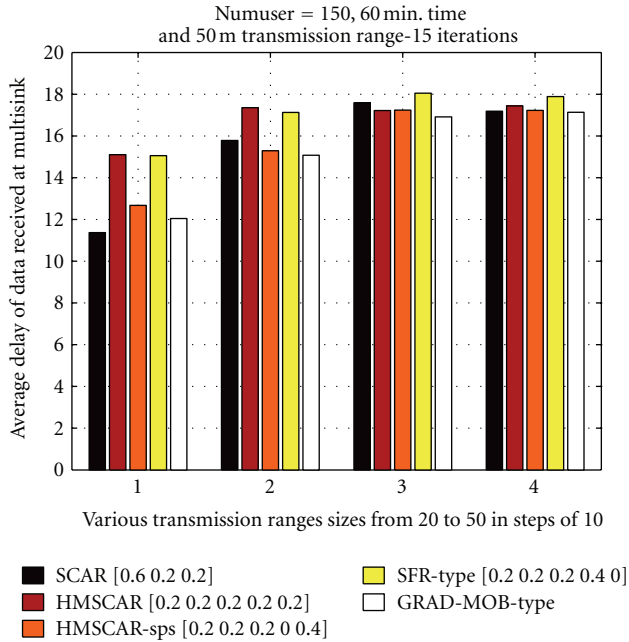


FIGURE 23: Average delay of packets for various trans. Range.

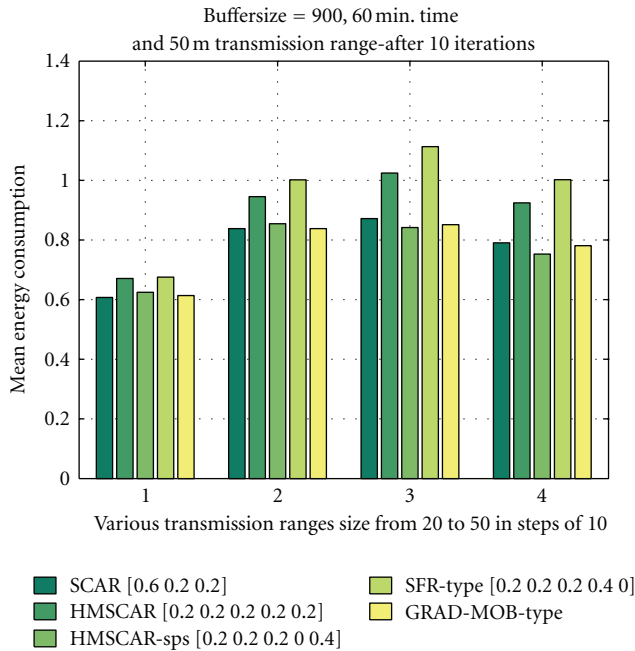


FIGURE 24: Average energy consumption of nodes for various trans. Range.

with sink node is increased due to increase in transmission range, and therefore the relative internode distance which a packet has to travel from source to destination is decreased.

7.5. *Simulation-5.* Figure 25 shows the number of packets received for HMSCAR algorithm with various weight combinations of weight w_1 and w_5 . The number of packets received

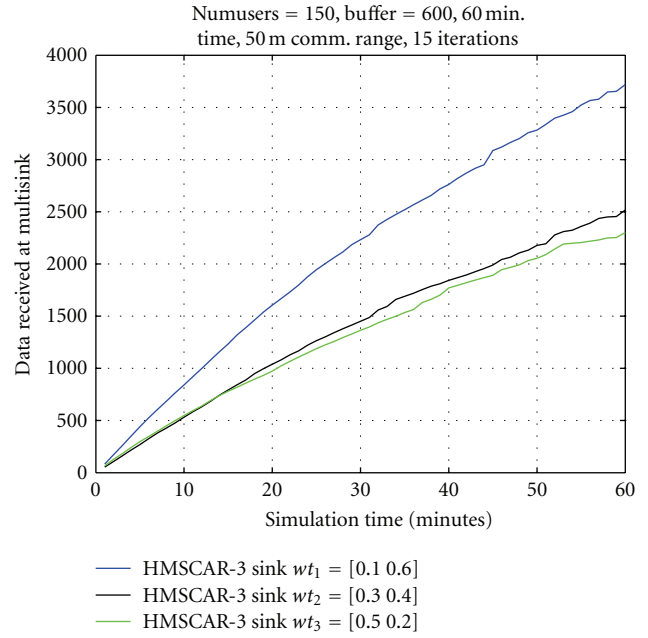


FIGURE 25: Data received versus simulation time for various w_1 and w_5 .

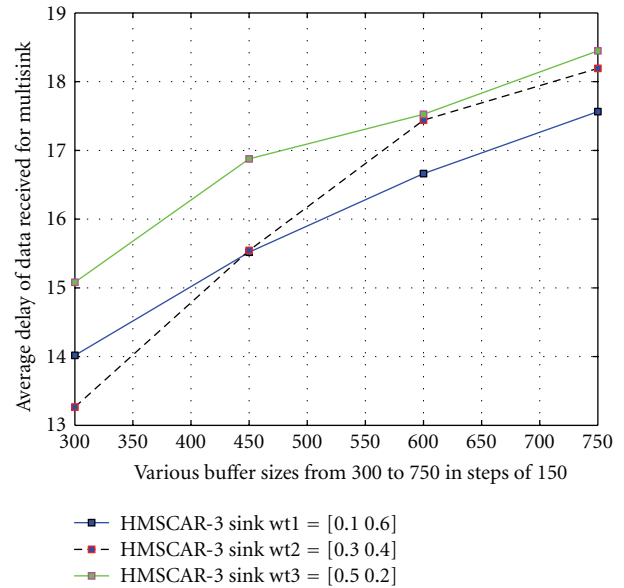


FIGURE 26: Average delay of packets received for various w_1 and w_5 .

is more for higher value of w_5 compared to w_1 . This shows the effectiveness of our proposed scheme.

Figure 26 shows average data delivery delay of the packets for HMSCAR algorithm with various weight combinations of weights w_1 and w_5 . Average data delivery delay is least for higher value of w_5 . Increasing buffer size also increases delay of messages.

Figure 27 shows energy consumption of data received at all the sinks for HMSCAR algorithm with various weight

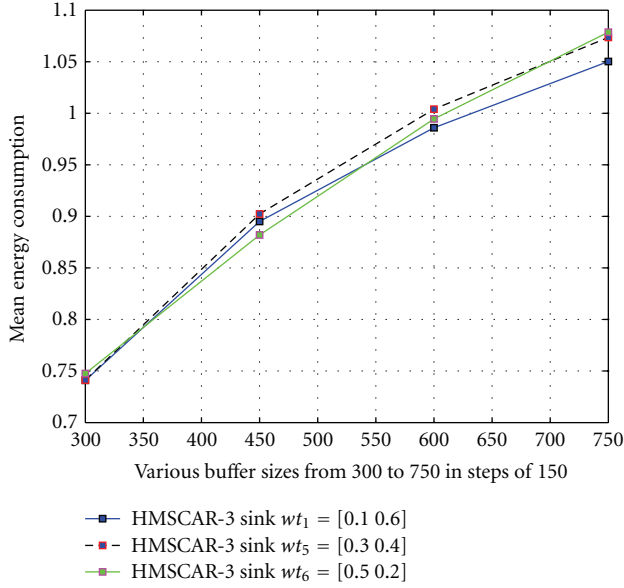


FIGURE 27: Average energy consumption of nodes for various w_1 and w_5 .

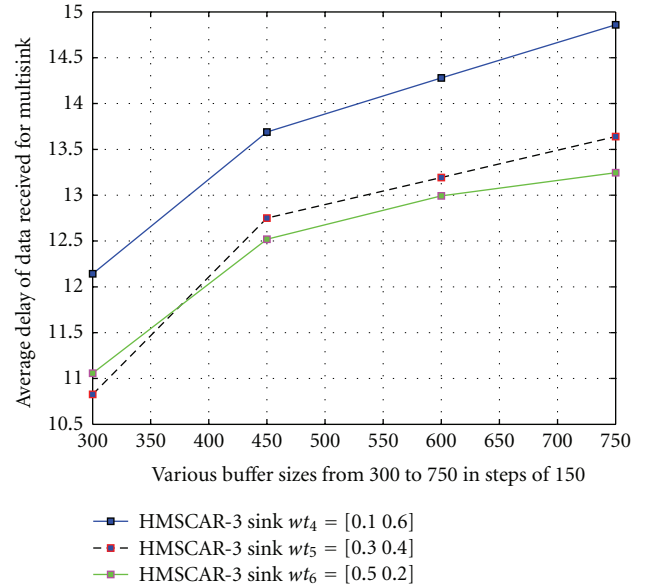


FIGURE 29: Average Delay of packets received for various w_3 and w_5 .

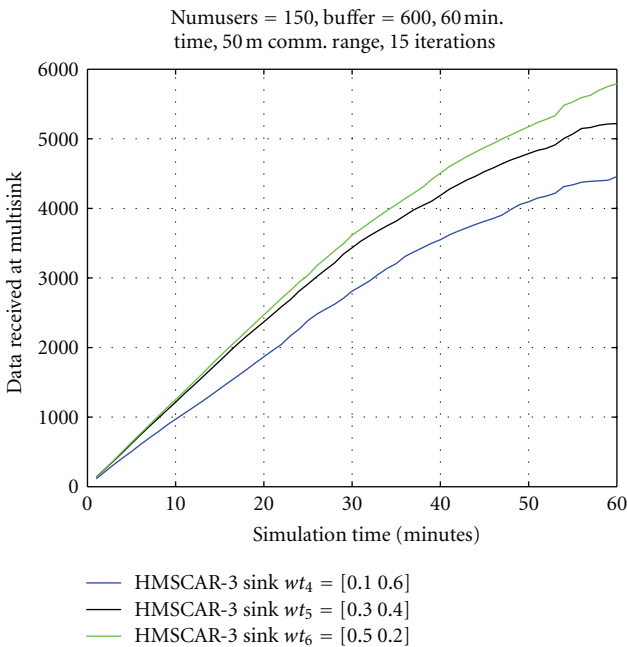


FIGURE 28: Data received versus simulation time for various w_3 and w_5 .

combinations of weights w_1 and w_5 . The power consumption is almost same for all different combination of w_5 and w_1 for buffer size less than 600. At buffer size 750, the HMSCAR algorithm with higher weight value of w_5 consumes less energy. The energy consumption increases as buffer size increases for all the three different weight combinations.

Figure 28 shows the number of packets received for HMSCAR algorithm with various weight combinations of

weights w_3 and w_5 . The number of packets received is more for higher value of weight w_3 compared to w_5 . This suggests inefficiency of proposed P_u parameter compared to $Colloc_u$, when packet delivery ratio is considered as a performance metric.

Figure 29 shows average data delivery delay of the packets for HMSCAR algorithm with various weight combinations of weights w_3 and w_5 . Average data delivery delay is worst for higher value of w_5 . This suggests the inefficiency of proposed P_u parameter compared to $Colloc_u$, when packet delay metric is considered. However, it should be noted that calculation of $Colloc_u$ needs routing message exchanges and routing table maintenance at each node, while calculation of P_u does not need message exchanges. It only needs the on-board GPS information, which is easily available on smart phones.

Figure 30 shows energy consumption of data received at all the sinks for HMSCAR algorithm with various weight combinations of weights w_3 and w_5 . The power consumption decreases for higher value of w_3 parameter compared to w_5 . The energy consumption increases as buffer size increases for all the three different weight combinations.

7.6. *Simulation-6.* Figure 31 shows the number of packets received for various algorithms with various combinations of levy exponents α and β . For various combinations of α and β , our proposed algorithm outperforms SCAR, SFR and GRAD-MOB. However, it can be observed that the performance of HMSCAR and HMSCAR-sps is best, for smaller value of α which is 0.5. This value corresponds to heavy tail distribution of flight length.

Figure 32 shows the number of packets received of various algorithms with single-sink replication scheme. Result shows that our proposed algorithms HMSCAR and HMSCAR-sps outperform SCAR, SFR, and GRAD-MOB only for α and β equal to $[0.5 \ 1.5]$ and $[1, 1]$, as compared to α

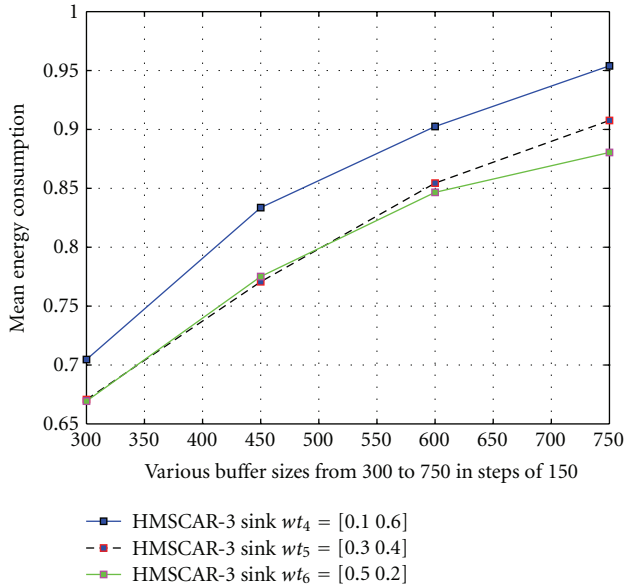


FIGURE 30: Average energy consumption of nodes for various w_3 and w_5 .

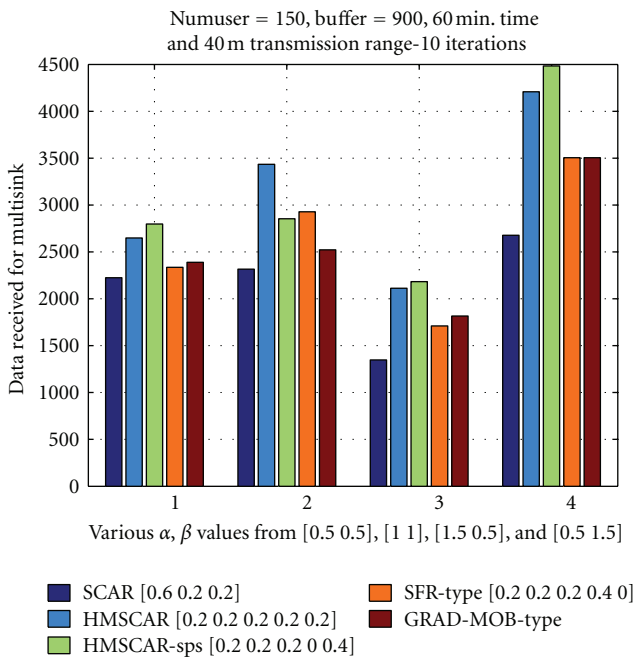


FIGURE 31: Data received versus time for various alpha and beta.

and β equal to [1.5 0.5] and [0.5 0.5]. This shows inability of our proposed scheme to perform better compared to SCAR, SFR, and GRAD-MOB in single-sink scenario. The nature of our newly proposed parameters are such that they perform the best for multisink scenario.

Figure 33 shows average data delivery delay of various algorithms with various combinations of levy exponents α and β . Average data delivery delay of HMSCAR-sps is the lowest amongst all the algorithms. This suggests efficiency of

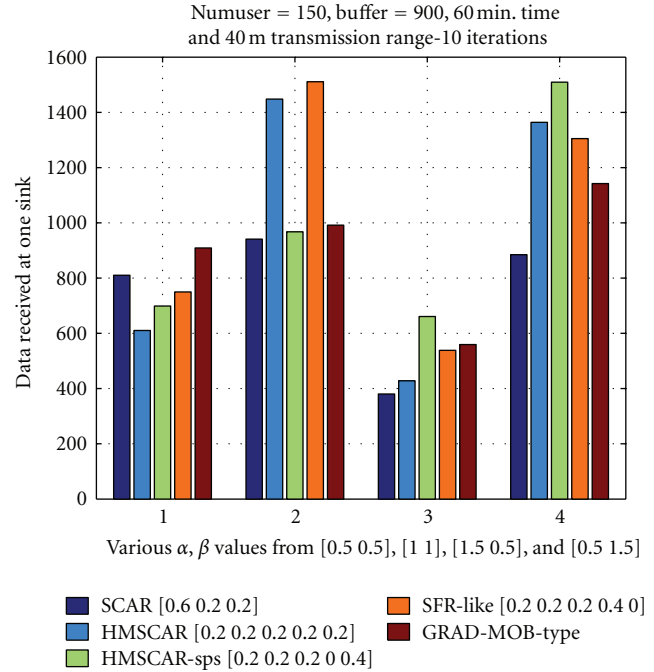


FIGURE 32: Data received versus time for for various alpha and beta.

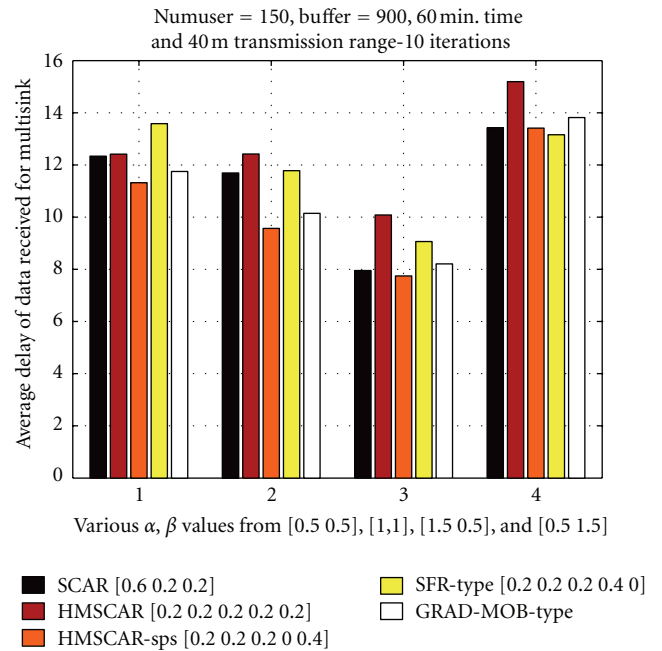


FIGURE 33: Average delay of packets received for various alpha and beta.

proposed P_u parameter for better delay performance under multisink case.

Figure 34 shows average data delivery delay for various algorithms with various combinations of levy exponents α and β . Average data delivery delay of various algorithms is worst for higher value of β equal to 1.5. Average data delivery delay of HMSCAR-sps is the lowest among all algorithms

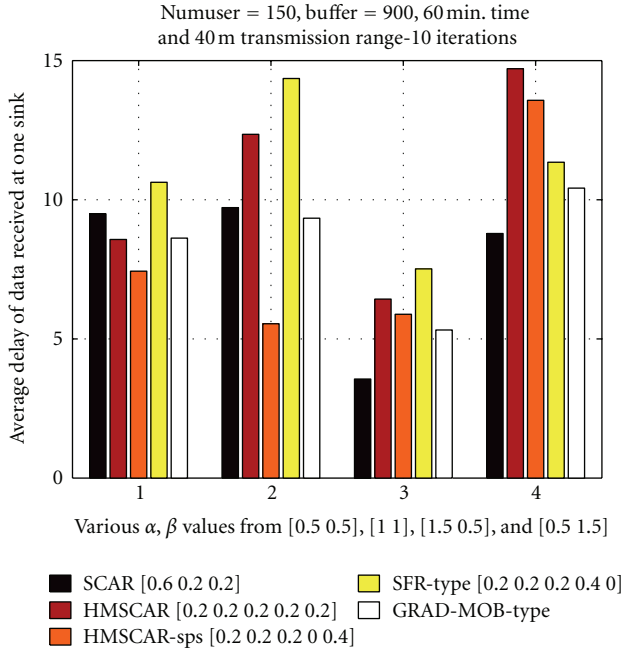


FIGURE 34: Average delay of packets received for various alpha and beta.

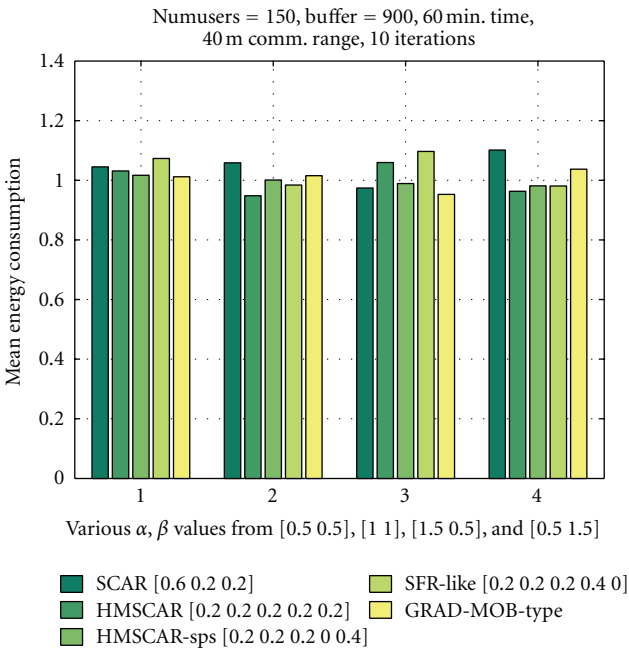


FIGURE 35: Average energy consumption of nodes for for various alpha and beta.

except for β equals to 1.5. When value of levy walk exponent is near 2, it closely matches Brownian motion. The threshold that is set for deciding super pause time is excessive for this value of levy exponent.

Figure 35 shows energy consumption of data received at all the sinks of various algorithms, with various combinations of levy exponents α and β . Results show that

our proposed algorithms, HMSCAR and HMSCAR-sps, outperform the SCAR, SFR, and GRAD-MOB except when α and β equal [1.5 0.5]. The number of super flight lengths decreases as levy exponent α is increased.

8. Conclusion

This paper presents an opportunistic routing algorithm, which exploits the truncated power law distribution of flight length and pause time of human mobility. The proposed algorithm’s performance depends upon availability of nodes, which exhibit super flight length and super pause time. This behavior is observed in tail part of the distribution and is a rare event. The probability of occurrence of this rare events is more for dense networks like CPSN. The majority of DTN routing schemes are aimed towards random mobility and are unrealistic for CPSN scenario. Proposed algorithm considers realistic mobility of cell phone users and performs better with realistic human mobility. The algorithms design is such that it selects nodes with super flight length as relay nodes and avoids nodes likely to observe super pause time, as the relay node. This is quite a natural observation for human mobility. Simulation results indicate HMSCAR’s best performance is under 3-sink presence compared to 1. Due to nature of newly introduced utility parameters, it performs better for multi-sink scenario. We have evaluated algorithm for single copy replication scheme, and data delivery and delay are improved simultaneously. This shows the power of newly introduced context parameters, which incorporates human walk characteristics into its design. With multisink presence HMSCAR and HMSCAR-sps perform best for all types of human walk (various combinations of levy exponent α and β). Since single-replication scheme is basis of all other replication schemes, the design of HMSCAR will provide a new insight for routing under human carried mobile devices.

Acknowledgments

This work has been supported by the Department of Science and Technology (DST), Government of India, and the India-UK Advanced Technology Center of Excellence in Next Generation Networks (IUATC) Project.

References

- [1] J. Shi, J. Wan, H. Yan, H. Suo, and C. Datong, “A survey of cyber-physical systems,” in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP’11)*, pp. 1–6, 2011.
- [2] F. J. Wu, Y. F. Kao, and Y. C. Tseng, “From wireless sensor networks towards cyber physical systems,” *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, 2011.
- [3] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, “A survey of mobile phone sensing,” *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
- [4] T. Abdelzaher, Y. Anokwa, P. Boda et al., “Mobiscopes for human spaces,” *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 20–29, 2007.

- [5] S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, G. Ahn, and A. Campbell, "Metrosense project: people-centric sensing at scale," in *Proceedings of the 1st Workshop on World-Sensor-Web (WSW'06)*, Citeseer, Boulder, Colo, USA, 2006.
- [6] S. C. Hu, Y. C. Wang, C. Y. Huang, and Y. C. Tseng, "A vehicular wireless sensor network for CO₂ monitoring," in *Proceedings of the IEEE Sensors Conference (SENSORS'09)*, pp. 1498–1501, Christchurch, New Zealand, October 2009.
- [7] B. Hull, V. Bychkovsky, Y. Zhang et al., "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06:)*, pp. 125–138, November 2006.
- [8] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 52–57, 2006.
- [9] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 323–336, 2008.
- [10] D. Chander, B. Jagyasi, U. B. Desai, and S. N. Merchant, "Spatio-temporally adaptive waiting time for cell phone sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 962476, 21 pages, 2011.
- [11] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007.
- [12] M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 126–139, 2010.
- [13] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications (HotMobile'08)*, pp. 11–16, February 2008.
- [14] S. B. Eisenman, N. D. Lane, and A. T. Campbell, "Techniques for improving opportunistic sensor networking performance," *Lecture Notes in Computer Science*, vol. 5067, pp. 157–175, 2008.
- [15] T. Liu, C. Sadler, P. Zhang, and M. Martonosi, "Implementing software on resource-constrained mobile sensors: experiences with impala and zebrantet," in *Proceedings of the 2nd international Conference on Mobile Systems, Applications, and Services*, pp. 256–269, ACM, 2004.
- [16] Y. Wang and H. Wu, "DFT-MSN: the delay/fault-tolerant mobile sensor network for pervasive information gathering," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, Barcelona, Spain, April 2006.
- [17] H. A. Nguyen and S. Giordano, "Routing in opportunistic networks," *International Journal of Ambient Computing and Intelligence*, vol. 1, no. 3, pp. 19–38, 2009.
- [18] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.
- [19] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.
- [20] L. Song and D. Kotz, "Evaluating opportunistic routing protocols with large realistic contact traces," in *Proceedings of the 2nd ACM Workshop on Challenged Networks*, pp. 35–42, ACM, 2007.
- [21] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking*, pp. 252–259, ACM, 2005.
- [22] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-200006, Duke University, 2000.
- [23] I. Psaras, L. Wood, and R. Tafazolli, "Delay-/disruption-tolerant networking: state of the art and future challenges," Tech. Rep., University of Surrey, UK, 2010.
- [24] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.
- [25] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 630–643, 2011.
- [26] C. Mascolo and M. Musolesi, "SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC'06)*, pp. 533–538, July 2006.
- [27] B. Pásztor, M. Musolesi, and C. Mascolo, "Opportunistic mobile sensor data collection with SCAR," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'07)*, pp. 1–12, October 2007.
- [28] H. Kanai, Y. Koizumi, H. Ohsaki, and M. Imase, "Gradient-based routing in delay tolerant mobile sensor networks incorporating node mobility," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC'12)*, pp. 235–239, Las Vegas, Nev, USA, 2012.
- [29] M. Shin, S. Hong, and I. Rhee, "DTN routing strategies using optimal search patterns," in *Proceedings of the 3rd ACM Workshop on Challenged Networks (CHANTS'08)*, pp. 27–32, September 2008.
- [30] G. Cugola and M. Migliavacca, "A context and content-based routing protocol for mobile sensor networks," *Wireless Sensor Networks*, vol. 5432, pp. 69–85, 2009.
- [31] J. M. Soares, M. Franceschinis, R. M. Rocha, W. Zhang, and M. A. Spirito, "Opportunistic data collection in sparse wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2011, Article ID 401802, 2011.
- [32] L. Guo, R. Beyah, and Y. Li, "SMITE: a stochastic compressive data collection protocol for Mobile Wireless Sensor Networks," in *Proceedings of the IEEE INFOCOM*, pp. 1611–1619, Shanghai, China, April 2011.
- [33] M. Keally, G. Zhou, and G. Xing, "Sidewinder: a predictive data forwarding protocol for mobile wireless sensor networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'09)*, Rome, Italy, June 2009.
- [34] J. Zhu, J. Cao, M. Liu, Y. Zheng, H. Gong, and G. Chen, "A mobility prediction-based adaptive data gathering protocol for delay tolerant mobile sensor network," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'08)*, pp. 730–734, New Orleans, La, USA, December 2008.

- [35] R. Jathar and A. Gupta, "Probabilistic routing using contact sequencing in delay tolerant networks," in *Proceedings of the 2nd International Conference on COMMunication Systems and NETWORKS (COMSNETS'10)*, Bangalore, India, January 2010.
- [36] M. C. González, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [37] K. Minkyong, D. Kotz, and K. Songkuk, "Extracting a mobility model from real user traces," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, Barcelona, Spain, April 2006.
- [38] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1360–1369, April 2001.
- [39] M. Crovella, "Performance evaluation with heavy tailed distributions," *Computer Performance Evaluation. Modelling Techniques and Tools*, pp. 1–9, 2000.
- [40] A. Kansal, M. Goraczko, and F. Zhao, "Building a sensor network of mobile phones," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN'07)*, pp. 547–548, April 2007.
- [41] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
- [42] D. Chander, B. Jagyasi, U. B. Desai, and S. N. Merchant, "Layered data aggregation in cell-phone based wireless sensor networks," in *Proceedings of the International Conference on Telecommunications (ICT'08)*, St. Petersburg, Russia, June 2008.
- [43] M. Shah, P. Verma, S. Merchant, U. Desai et al., "Leveraging pause time distributions for stable cluster-based data gathering in multihop cell phone-based sensor network," *IETE Technical Review*, vol. 28, no. 6, p. 470, 2011.
- [44] M. B. Shah, P. P. Verma, S. N. Merchant, and U. B. Desai, "Human mobility based stable clustering for data aggregation in singlehop cell phone based wireless sensor network," in *Proceedings of the 25th IEEE International Conference on Advanced Information Networking and Applications (AINA'11)*, pp. 427–434, Biopolis, Singapore, March 2011.
- [45] B. Mandelbrot, *The Fractal Geometry of Nature*, New York, NY, USA, 1983.
- [46] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

