



HAL
open science

Game balanced multi-factor multicast routing in sensor grid networks

Qingfeng Fan, Naixue Xiong, Karine Zeitouni, Qiongli Wu, Athanasios Vasilakos, Yu-Chu Tian

► **To cite this version:**

Qingfeng Fan, Naixue Xiong, Karine Zeitouni, Qiongli Wu, Athanasios Vasilakos, et al.. Game balanced multi-factor multicast routing in sensor grid networks. *Information Sciences*, 2016, 367-368, pp.550-572. 10.1016/j.ins.2016.06.049 . hal-04371711

HAL Id: hal-04371711

<https://hal.science/hal-04371711>

Submitted on 2 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License



Queensland University of Technology
Brisbane Australia

This may be the author's version of a work that was submitted/accepted for publication in the following source:

Fan, Qingfeng, Xiong, Naixue, Zeitouni, Karine, Wu, Qiongli, Vasilakos, Athanasios, & [Tian, Glen](#)
(2016)
Game balanced multi-factor multicast routing in sensor grid networks.
Information Sciences, 367 - 368, pp. 550-572.

This file was downloaded from: <https://eprints.qut.edu.au/221471/>

© Consult author(s) regarding copyright matters

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

License: Creative Commons: Attribution-Noncommercial-No Derivative Works 2.5

Notice: *Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.*

<https://doi.org/10.1016/j.ins.2016.06.049>

Information Sciences, vols. 367-368, pp. 550-572, 1 Nov 2016.

DOI: 10.1016/j.ins.2016.06.049.

Game Balanced Multi-factor Multicast Routing in Sensor Grid Networks

Qingfeng Fan^{a,*}, Naixue Xiong^b, Karine Zeitouni^a, Qiongli Wu^c, Athanasios Vasilakos^d, Yu-Chu Tian^e

^aLaboratoire DAVID, University of Versailles-Saint-Quentin, 78035, Versailles Cedex
France

^bDept of Business and Computer Science, Southwestern Oklahoma State University, 73096,
USA

^cLaboratory Applied Mathematics and Systems, Ecole Centrale de Paris, Grande Voie des
Vignes, 92295, Chatenay-Mâlabry Cedex, France

^dDept of Computer Science, Electrical and Space Engineering, Lulea University of
Technology, 97187, Lulea, Sweden

^eSchool of Electrical Engineering and Computer Science, Queensland University of
Technology, GPO Box 2434, Brisbane QLD 4001, Australia.

Abstract

In increasingly important sensor grid networks, multicast routing is widely used in data aggregation and distributed query processing. It requires multicast trees for efficient data transmissions. However, sensor nodes in such networks typically have limited resources and computing power. Efforts have been made to consider the space, energy and data factors separately to optimize the network performance. Considering these factors simultaneously, this paper presents a game balance based multi-factor multicast routing approach for sensor grid networks. It integrates the three factors into a unified model through a linear combination. The model is standardized and then solved theoretically by using the concept of game balance from game theory. The solution gives Nash equilibrium, implying a well balanced result for all the three factors. The theoretic results are implemented in algorithms for cluster formation, cluster core selection, cluster tree construction, and multicast routing. Extensive simulation experiments show that the presented approach gives mostly better overall performance than benchmark methods.

Keywords: Game balance, multicast, query, semantic cache, sensor grid networks

*Corresponding author.

Email address: qingfeng.fan@prism.uvsq.fr; lyqingfeng@gmail.com (Qingfeng Fan)

1. Introduction

With the rapid development of wireless communication technologies, sensor grid networks are becoming more and more popular and increasingly important. They gather, distribute, and act on, the information about the behaviour of all participants, e.g., suppliers and consumers [1, 2]. They are widely used in various applications. Among those applications are smart power grids, environmental monitoring, smart transportation, and habitat monitoring [3].

A sensor grid network consists of hundreds of thousands of wireless sensor nodes. In general, these sensor nodes have limited resources and computing power. Thus, the computational tasks that are resource demanding and/or computationally intensive have to be partially or mostly offloaded to somewhere else from the sensor devices for prompt processing. Also, data gathered or generated by the sensor devices need to be transmitted over the sensor grid network [4]. All these requirements cause severe issues in wireless communications. Particularly, a challenges in sensor grid networks is to support efficient multicast routing for data aggregation and distributed query processing [5].

In sensor grid networks, efficient multicast routing typically use data aggregation. For data aggregation, the technique of data aggregation tree is generally employed. More specifically, a base station or sink node gradually collects data from distributed sensor nodes by using a reverse multicast tree [6]. Thus, multicast becomes a key concept in data aggregation for traffic routing and distributed query optimization.

While research on multicast has been extensive for data aggregation, existing multicast schemes have mainly considered the shortest transmission distance from the geographical factor perspective. When a hierarchical multicast tree is constructed, the geographical center is often chosen as a core node at which the data is aggregated. This reduces the transmission distance of the data [7]. However, in addition to the geographical space factor, energy consumption and data generation volume are also significant factors in sensor grid networks [8]. For a longer lifetime of the network, sensor nodes with a higher energy residual should be assigned more communication tasks. This requires to change the core node dynamically in multicast routing. Also, the volume of data that each sensor node generates or collects is quite different from each other. This has a significant impact on the performance of the data communications through data aggregation. While space, energy and data factors have been considered separately in existing methods, simultaneous considerations of all these three factors have not yet been reported except our preliminary work [9, 10]. This motivates our research in this paper on efficient routing in sensor grid networks.

This paper presents a game-balanced multi-factor multicast routing approach for sensor grid networks. It makes three main contributions: 1) A unified model is established with simultaneous considerations of the three factors of space, energy and data through a linear combination with unknown coefficients; 2) After standardization, the unified model is solved theoretically for all unknown coefficients by using the concept of game balance, giving Nash equilibrium with well-balanced result among the three factors; and 3) the theoretical

46 results are implemented in five algorithms. The presented approach is evaluated
47 through simulation experiments against benchmark methods.

48 The remainder of this paper is organized as follows. Section 2 reviews the
49 related work and motivates our research. Section 3 describes the multi-factor
50 problem. The problem is solved theoretically in Section 4 through game balance
51 theory. The theoretical results are implemented in algorithms in Section 5. Sec-
52 tion 6 demonstrates the performance of the presented approach in comparison
53 with benchmark methods. Finally, Section 7 concludes the paper.

54 **2. Background, Related Work and Motivations**

55 Multicast routing in sensor grid networks relies on data aggregation and dis-
56 tributed query processing. This section reviews the related work of data aggre-
57 gation, distributed query processing, and multicast routing. Then, it discusses
58 the motivations of our research in this paper.

59 *2.1. Data Aggregation*

60 In sensor grid networks, data acquisition typically utilizes data aggregation
61 through a structure called “data aggregation tree”. More specifically, a base
62 station or sink node collects data from distributed sensor nodes by using a
63 reverse multicast tree. The collected data are aggregated and then sent out [6].

64 In the network layer, there are two categories of routing strategies through
65 data aggregation: address-centric (AC) and data-centric (DC). For AC routing,
66 each source node sends data along the shortest path in the intermediate nodes
67 to the sink node. In comparison, DC routing considers the content of the data
68 to be transmitted. During the data forwarding process, the intermediate sensor
69 nodes aggregate data from multiple data sources according to the content of the
70 data. They do not necessarily follow the shortest path for traffic routing.

71 Energy is one of the major issues in wireless and mobile sensor grid networks.
72 It has been considered in data aggregation design. Two popularly used methods
73 for data aggregation and traffic routing with consideration of energy are and
74 TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [11] and
75 TDMA-based LEACH (Low-Energy Adaptive Clustering Hierarchy) [12]. While
76 LEACH is a good approximation of a proactive network protocol, with some
77 minor differences, TEEN is targeted at reactive networks.

78 Both LEACH and TEEN use periodic clustering. They experience two oper-
79 ational phases in each round: cluster establishment and data communications.
80 In clustering, adjacent nodes form a dynamic cluster and generate its core. To
81 achieve a balanced network energy consumption, each node in the cluster needs
82 to rotate the cluster core. Nodes that have been cluster cores cannot become
83 cluster cores again for a certain number of rounds. In data communications,
84 cluster nodes send data to the cluster core. Then, the cluster core aggregates
85 the data and sends the aggregated data to the sink nodes.

86 *2.2. Distributed Query Processing*

87 Distributed query processing in sensor grid networks can use data aggrega-
88 tion for efficient data collection from multiple data sources. It also disseminates
89 queries from sink nodes to other sensor nodes over the network. Then, the
90 sensor data converge toward the sink nodes in a reverse multicast manner.

91 Existing studies on sensor databases, e.g., TinyDB, have limitations in dis-
92 tributed query processing. Firstly, they consider query processing as a basic
93 operation. However, queries from sensor nodes should be processed collabo-
94 ratively with other sensors nodes rather than exclusively by a central site for
95 sensor grid networks [13]. Secondly, in most existing methods, queries are usu-
96 ally processed in a static environment. But mobile sensor nodes and the network
97 environment are highly dynamic. While there exist considerations of dynamic
98 networks, a usual implicit assumption is instant query processing during which
99 the network topology and connections do not change.

100 Depending on how query answers are delivered, a sensor grid query system is
101 designed and implemented differently. One type of sensor grid systems delivers
102 query answers to a sink node that generates the queries. In another type of
103 sensor grid systems, queries originate from arbitrary peers, to which the answers
104 must be delivered back.

105 Query processing in sensor grid networks is usually a broadcast or multicast
106 process from one peer, e.g., the sink, to multiple peers. Periodically, query
107 processing sends out a query list via broadcast or multicast. Then, it keeps
108 communicating with multiple peers for query questions and answers. In this
109 way, it gradually attains global query distribution knowledge, which can be
110 utilized to predict queries using former query information [13].

111 Generally, sensor nodes in sensor grid networks have limited resources, e.g.,
112 bandwidth, energy, memory, storage, and computing power. This will affect how
113 to build a multicast tree to propagate query messages. The factors of space,
114 energy and data have been considered separately in existing methods. However,
115 how to consider all these factors in an integrated model is still an open problem.
116 This will be addressed in this paper.

117 *2.3. Existing Multicast Protocols*

118 Efforts have been made to develop multicast protocols. NICE is an ex-
119 tendible multicast protocol. It is a hierarchical multicast tree technique. An-
120 other multicast protocol is Double-Channel XY Multicast Wormhole Routing
121 (DCXY). It uses an extension of the XY routing algorithm to set up the routing
122 scheme [14]. Dual-Path Multicast Routing (DPM) is a multicast protocol de-
123 veloped for 2-D MESH networks. For content-addressable networks (CANs) [7],
124 CAN-based Multicast Routing (CANM) is developed for communications [1].

125 Despite the progress mentioned above, existing multicast technologies con-
126 sidered only one factor or two that affect data transmission efficiency [15]. For
127 example, DCXY, DPM and AC routing considered the location (space) factor
128 only. LEACH and TEEN focused on the energy factor only. DC routing only
129 considered the data factor. Emphasizing a single factor only without considera-
130 tion of other factors largely limits the applications of existing multicast protocols

131 in sensor grid networks. In large-scale sensor grid networks, a balanced consider-
132 ation of all these important factors is essential for improved overall performance
133 in dynamic environments.

134 *2.4. Motivations and our Preliminary Work*

135 Existing techniques for multicast routing, data aggregation and distributed
136 query processing have mainly addressed constraint factors separately. Traditionally,
137 the geographical location is considered, leading to the shortest path
138 technique with fewest links. When a hierarchical multicast tree is constructed,
139 the geographical central node is chosen as the cluster core [16]. Later, the energy
140 factor is considered for energy-efficient multicast communications [17, 18]. But
141 the energy factor has not incorporated with the space factor. Furthermore, the
142 amounts of data from different sensor nodes differ significantly in sensor grid
143 networks. In general, more data require more data collection, processing, shar-
144 ing, and querying [19, 20]. This has been addressed in data-focused methods.

145 The requirement of improving the overall performance of a sensor grid net-
146 work in dynamic environment demands a unified approach that considers all
147 these factors simultaneously. However, such an approach has not been found in
148 the literature except our preliminary work [9, 10]. Among multiple factors, three
149 factors show particular significance: space, energy and data. To develop such
150 an approach with consideration of the three factors, one needs to integrate the
151 three factors into a unified model. Then, solve the model for cluster formation,
152 cluster core selection, determination of model parameters, and multicast rout-
153 ing. Furthermore, the model and its theoretical solution need to be extended
154 for the application scenarios [21]. All these requirements motivate our research
155 in the present paper. Particularly, the concept of game balance is adopted in
156 this paper for a balanced solution from multiple factors.

157 Our preliminary studies on this topic have addressed more than one factor [9,
158 10]. Our work in [9] considered two factors, space and data, in a unified model.
159 Our work in [10] moved one step further to consider the energy factor in addition
160 to the space and data factors. However, algorithm design was not given, and
161 limited experiments were conducted to demonstrate the approach.

162 The present paper extends our preliminary studies substantially in both
163 breadth and depth. Firstly, we have extended the two-factor model [9] to a
164 three-factor model. This extension is not straightforward, but requires new
165 mathematical treatment and also gives a new insight into the multicast routing
166 with consideration of multiple factors. The new insight directly leads to finite
167 M -factor scenarios. Secondly, we have refined the mathematical treatment of
168 the three-factor problem in our previous study [10]. Particularly, the math-
169 ematical treatment is clearly separated into several steps: linear combination
170 modelling, model standardization, factor weights at Nash equilibrium, and so-
171 lution at Nash equilibrium. Thus, the theoretic results become neat and more
172 compact. Moreover, we have designed detailed algorithms to implement the
173 theoretical results, and have conducted comprehensive experiments. Algorithm
174 design and comprehensive experiments are missing in our previous work [10].

175 **3. Description of the Multi-factor Problem**

176 This section describes the problem of multicast routing for data aggregation
 177 and distributed query processing. Three factors are considered for each cluster of
 178 sensor nodes. Then, for each factor, a weight vector is defined for the cluster. It
 179 is used for selecting the cluster core and building the multicast tree. After that,
 180 a linear combination of the three weight vectors is designed to derive a general
 181 weigh vector. All notations used in this paper are summarized in Table 1.

182 *3.1. Multiple factors in the Multicast Problem*

183 Consider a cluster of sensor nodes in a sensor grid network as shown in
 184 Figure 1. A multicast group with l members is denoted as: $G = \{U_0, \dots, U_i, \dots,$
 185 $U_{l-1}\}$, $i = 0, \dots, l - 1$. Each multicast member is identified by m coordinates:
 186 $U_i = (u_{i,0}, \dots, u_{i,j}, \dots, u_{i,m-1})$, when $0 \leq j \leq m - 1$. For example, for member
 187 U_0 , its two-dimensional coordinates $(u_{0,0}, u_{0,1})$ are $(0, 0)$.

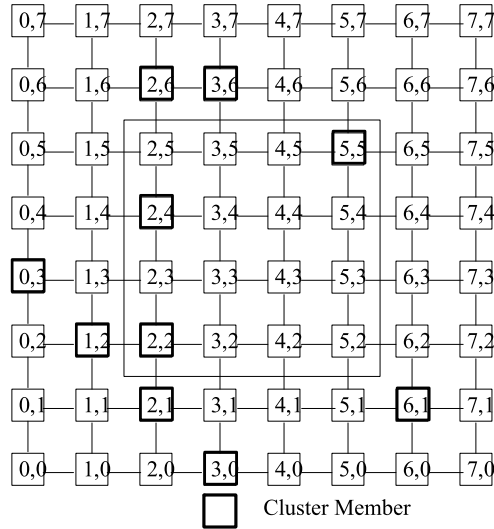


Figure 1: A cluster of sensor nodes within a 2 - D sensor grids network. Thick square boxes are cluster members. The spatial center node of the cluster should be in the area $[1, 1] \times [6, 6]$.

188 We define neighbour nodes as follows. Consider two sensor nodes $U_i =$
 189 $(u_{i,0}, \dots, u_{i,j}, \dots, u_{i,m-1})$, where $i \in [0, l - 1]$, and $U_{i'} = (u_{i',0}, \dots, u_{i',j}, \dots,$
 190 $u_{i',m-1})$, where $i' \in [0, l - 1]$ and $i' \neq i$. U_i and $U_{i'}$ are neighbours if and only
 191 if $u_{i,j} = u_{i',j}$ for all j , except that $u_{i,j'} = u_{i',j'} \pm 1$ along the 1-D j' . In an m -D
 192 sensor grid network, a node may have m to $2m$ neighbors [14].

193 We further define Manhattan distance between two nodes. In a 2-D sensor
 194 grid network, the static delay distance between (X_0, Y_0) and (X_1, Y_1) is
 195 $|X_1 - X_0| + |Y_1 - Y_0|$. The sum of the static delay distances from all other nodes,
 196 (X_i, Y_i) to (X_0, Y_0) , $i = 1, \dots, n - 1$, is $f(X_0, Y_0) = \sum_{i=1}^{n-1} (|X_i - X_0| + |Y_i - Y_0|)$.

Table 1: Nomenclature.

Notation	Description
AD	Average multicast delay
C	The strategies of Collaboration
C^*	Cluster core, $C^* = (c_0^*, \dots, c_i^*, \dots, c_{m-1}^*)$
C_i	The cost for bringing object i into the cache
c_i	New general core
$c_{i,a}, c_{i,b}, c_{i,c}$	Space, energy and data cores, respectively
$d(s, u_i)$	Packet delay from source s to member u_i
F_i	Frequency usage of the object I
G	The set of the system
I	The object of a semantic cache item
K_i	Priority key in the GDSF semantic cache algorithm
$(k, 3k - 1)$	The range of a random constant, $k = 3$ is used in NICE
L	A running age factor
l	The number of members of the system
M	The strategies of monopolization
m	The number of coordinates
MN	Management node
n	The number of nodes in the cluster
$n_{<j}, n_{>j}, n_{=j}$	The number of cluster members with the J th coordinates $>$ (right nodes of J th row), $<$ (left nodes of J th row), and $= u_j$ (the nodes just on the J th row), respectively
S, S_i	The cluster size, and the size of the object I , respectively
$SPAN$	The Shortest-Path Area Nodes
TET	Time_Energy_Threshold
U_0, \dots, U_{l-1}	l members of system G
U_i	i th node identified by m coordinates, $U_i = (u_{i,0}, \dots, u_{i,m-1})$
$U_{i'}$	Node also identified by m coordinates, $U_{i'} = (u_{i',0}, \dots, u_{i',m-1})$
u_j	Finite strategy set, $j = 1, \dots, m$
W	General weight vector
W', W'', W'''	Space, energy and data weight vectors, respectively
W'_j, W''_j, W'''_j	J th cluster: space, energy & data weight vectors, respectively
$w_{j,i}, w_{j,i}, w_{j,i}$	J th cluster: space, energy & data weights of node i , respectively
$W_{i,j}$	General weights of the nodes
W_j	General weight vector of the J th cluster
$w_{j,i}$	The general weight of the node i within the J -th cluster
$W_{i,j}^{(1)}, W_{i,j}^{(l)}, W_{i,j}^{(m)}$	The first, l -th and m -th weight vectors, respectively
$(X_0, Y_0), (X_1, Y_1)$	Two coordinates of two nodes
$ X_1 - X_0 + Y_1 - Y_0 $	Delay distance of two nodes (X_0, Y_0) and (X_1, Y_1)
$[x_0, y_0] \times [x_1, y_1]$	Area of the shortest paths between (x_0, y_0) and (x_1, y_1)
$\alpha_i, \beta_i, \gamma_i$	Linear parameters
$\alpha_i^{(1)}, \dots, \alpha_i^{(m)}$	Linear modulus, $\alpha_i^{(1)}, \dots, \alpha_i^{(m)} \geq 0, \alpha_i^{(1)} + \dots + \alpha_i^{(m)} \neq 0$
$\theta_1, \theta_2, \theta_3$	Angles between W', W'' and W'' , respectively, and W

197 For every cluster of sensor nodes, three weight vectors are used to character-
 198 ize the three factors of space, energy and data, respectively. For the i -th cluster
 199 with n nodes, they are defined as follows:

- 200 • Space weight vector $W'_i = (w'_{i,0}, \dots, w'_{i,j}, \dots, w'_{i,n-1})$, $i = 0, \dots, n-1$,
 201 where $w'_{i,j}$ indicates the space weight of node i within the i th cluster.
- 202 • Energy weight vector $W''_i = (w''_{i,0}, \dots, w''_{i,j}, \dots, w''_{i,n-1})$, $i = 0, \dots, n-1$,
 203 where $w''_{i,j}$ indicates the data weight of node j within the i -th cluster;
- 204 • Data weight vector $W'''_i = (w'''_{i,0}, \dots, w'''_{i,j}, \dots, w'''_{i,n-1})$, $i = 0, \dots, n-1$,
 205 where $w'''_{i,j}$ indicates the data weight of the node j within the i th cluster.

206 These three weight vectors need to be determined. Then, a general weight vector
 207 is designed from these three vectors for the hole cluster.

208 3.2. The Space Weight Vector W'

209 The space weight of a node relates to its location at a specific time instant.
 210 It characterizes the node's degree of closeness to a location, typically a core
 211 node. To determine the space weight of the node, we need to find the central
 212 node of the cluster first. Then, derive the space weight from the shortest path
 213 principle. The greater the space weight, the closer to the cluster core the node.
 214 The node with the maximum weight is the space cluster core of the cluster [22].

215 As an example, the upper part of Table 2 shows space weight values for a
 216 sensor grid network. In this cluster, the node at cell (2, 2) has the maximum
 217 weight $w'_{(2,2)} = 10$. Thus, it is the cluster space core. Building a multicast tree
 218 by considering the space weight only gives the result in Fig. 2(a). Using this
 219 multicast tree, the system achieves the shortest communication distance.

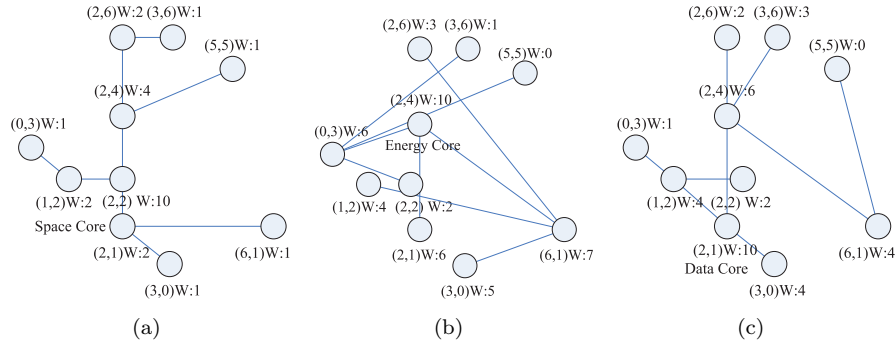


Figure 2: Multicast trees built from the weights shown in Table 3: (a) from space weights only; (b) from energy weights only; and (c) from data weights only.

Table 2: Weights of a sensor grid network. The upper, middle and lower parts are space, energy and data weights, respectively. The boxed cells belong to the same cluster.

		X=0	X=1	X=2	X=3	X=4	X=5	X=6
Space weights	Y=6	0	0	2*	1*	0	0	0
	Y=5	0	0	3	2	1	1*	0
	Y=4	0	0	4*	2	1	1	0
	Y=3	1*	1	5	2	1	1	0
	Y=2	1	2*	10*	4	2	2	0
	Y=1	0	0	2*	1	0	0	1*
	Y=0	0	0	1	1*	0	0	0
Energy weights	Y=6	0	0	3*	1*	3	0	0
	Y=5	0	1	1	4	1	0*	0
	Y=4	0	0	10*	6	5	1	0
	Y=3	6*	1	3	4	3	3	0
	Y=2	2	4*	2*	3	5	7	3
	Y=1	1	3	6*	3	0	0	7*
	Y=0	0	1	4	5*	0	0	0
Data weights	Y=6	0	0	2*	3*	3	0	0
	Y=5	0	1	3	4	1	0*	0
	Y=4	0	0	6*	6	5	1	0
	Y=3	1*	1	3	4	3	3	0
	Y=2	2	4*	2*	3	5	7	1
	Y=1	1	3	10*	3	0	0	4*
	Y=0	0	1	4	4*	0	0	0

220 3.3. The Energy Weight Vector W''

221 In a cluster of sensor nodes, some nodes consume more energy and some
222 others consume less. The nodes with a higher power residual may be able to
223 transmit data at a higher transmission rate. To achieve a balanced energy
224 consumption among the cluster members, each node needs to rotate the cluster
225 core based on its energy status. A randomized rotation of cluster heads is
226 used in LEACH. Similar to LEACH, TEEN is also a routing protocol based
227 on clustering. However, in comparison with LEACH, it is more adaptive to a
228 reactive sensor network.

229 In our modelling of the energy weight vector, the traditional energy algo-
230 rithms LEACH and TEEN are adopted. The energy weight of a node is defined
231 according to the remaining lifetime of the node. The node with the maximum
232 value of energy weight is selected as the cluster core. As an example, the middle
233 part of Table 2 lists the energy weights for the same cluster of sensor nodes

234 discussed previously. The cluster core is the node at cell (2,4) with energy
 235 weight $w''_{(2,4)} = 10$. After the cluster core is selected, a multicast tree can be
 236 established from the energy weights only. It is shown in Fig. 2(b). With this
 237 multicast tree, the system achieves the highest energy efficiency.

238 3.4. The Data Weight Vector W'''

239 A sensor grid network generally transmits a huge amount of data. There-
 240 fore, the cost of having, using or transmitting the data becomes an important
 241 factor [23, 24]. A data weight can be defined to characterize the consumption of
 242 network resources. For example, the average delay and the number of links used
 243 for transmissions of the data are indicators of the cost of the data. To simplify
 244 the problem, the data weight of a node is described according to the amount
 245 of data on the node [25, 26]. This is based on the observation that more data
 246 generally means more data processing and queries [5, 27].

247 In real network scenarios, the cost of data is a complicated variable. It may
 248 be a function of multiple and dynamic parameters [28]. For instance, the query
 249 hot degree, which means query frequency, is a good indicator of the data weight.
 250 In a sensor grid database, different tables or data items have very different query
 251 frequencies, sometimes at the ratio of 1:10:100, even for the same data quantity.
 252 When a node with a highest query hot degree is chosen as the cluster core, the
 253 query message propagation can be reduced and consequently the query efficiency
 254 can be improved. This will naturally reduce fading and shadowing [29].

255 Furthermore, the semantic cache technique has been employed in smart sensor
 256 databases [30]. It stores query results and query semantics in order to re-
 257 spond to future queries. When a new query result must be stored in a saturated
 258 cache, the most irrelevant queries must be evicted. When deciding which cached
 259 queries are replaced, the cost and frequency of access to the cached objects need
 260 to be considered in addition to the size constraint.

The Greedy Dual-Size Frequency (GDSF) semantic cache algorithm replaces
 the object with the smallest key for a semantic cache value function [24]:

$$K_i = F_i * C_i / S_i + L, \quad (1)$$

261 where K_i is the priority key, F_i is the frequency of using the object I , C_i is the
 262 cost associated with bringing the object i into the cache, S_i is the size of the
 263 object I , and L is a running age factor. The semantic cache value of a node is
 264 the sum of the values for different semantic cache items. The larger the semantic
 265 cache value is, the greater the data communications, query and propagation are.

266 Once the data weights are calculated for all nodes in a cluster, the cluster
 267 core is selected with the highest data weight among all cluster members. For
 268 the example shown in the lower part of Table 2, the cluster core is the cell at
 269 (2, 1) which has the highest data weight $w'''_{(2,1)} = 10$. Then, a multicast tree can
 270 be established based on the data weights only, as shown in Fig. 2(c).

271 3.5. Integration of the Three Weight Vectors

272 Understandably, focusing on space, energy and data separately results in dif-
 273 ferent multicast topology for data aggregation and traffic routing in a sensor grid

274 network, as clearly demonstrated in the examples in Table 2 and Fig. 2. Each
 275 of the three sets of results only shows the best interest in its own factor. Using
 276 any of them will ignore the requirements and constraints from the other two.
 277 This raises a question: how to consider all these three factors simultaneously in
 278 a unified framework to construct a multicast tree?

To answer this question, a general weight vector is derived from the three weight vectors. For the i -th cluster, the general weight vector is denoted by $W_{ij} = (w_{i,0}, \dots, w_{i,j}, \dots, w_{i,n-1})$, $i = 0, \dots, n - 1$. This paper defines the general weight vector as a weighted sum of the space weight vector W'_{ij} , energy weight vector W''_{ij} and data weight vector W'''_{ij} :

$$W_{ij} = \alpha_i^* W'_{ij} + \beta_i^* W''_{ij} + \gamma_i^* W'''_{ij}, \alpha_i^*, \beta_i^*, \gamma_i^* \geq 0, \alpha_i^* + \beta_i^* + \gamma_i^* \neq 0, \quad (2)$$

279 where α_i^* , β_i^* and γ_i^* are coefficients to be determined.

280 Then, the problem of our research is how to derive the coefficients α_i^* , β_i^*
 281 and γ_i^* to maximize the benefits for all three factors. This will be solved in the
 282 next section from game balance theory.

283 4. Theoretic Results for Solving the Multi-factor Problem

284 In this paper, each of the three factors is considered as a game player. Three
 285 factors as game players enter a game. Each player tries to achieve its best
 286 interest. The concept of collaborative games has been shown to be useful for
 287 constructing multicast trees for data communications [31, 32]. In the following,
 288 the multi-factor problem is also investigated from game theory. The concept of
 289 Nash equilibrium is introduced first. Then, the linear combination of the three
 290 weight vectors shown in Eq. (2) is standardized. After that, the coefficients
 291 of the standardized linear combination model are derived theoretically at Nash
 292 equilibrium. Furthermore, some extensions are made for different scenarios.
 293 Multicast trees will be established from the general weight vector.

294 4.1. Game Balance Analysis

295 Let us start with discussions of a complete information game of two players
 296 (factors): space and energy weight vectors W' and W'' . In this game, each
 297 player has two strategies: Monopolization (M) or Collaboration (C). If both
 298 choose M strategy, both will get the minimal benefits. If one player choose
 299 M while the other choose C, the player who has chosen M will maximize the
 300 benefits while the other will get minimal benefits. This means when one player
 301 knows the other one has already chosen M, it does not make sense for the player
 302 to choose C. The only win-win strategy is C for both players.

303 In geometry, the gain for players 1 and 2 are $\cos \theta_1$ and $\cos \theta_2$, respectively,
 304 where θ_1 is the angles of the weight vector W' and the general weight vector
 305 W , and θ_2 is the angle of W'' and W . The best result for both game players is
 306 $\cos \theta_1 = \cos \theta_2$, implying that $\theta_1 = \theta_2$, which is Nash equilibrium.

In this paper, three weight vectors are considered simultaneously, i.e., W' for space, W'' for energy and W''' for data. This three-player game achieves the best results for all three players at Nash equilibrium

$$\cos\theta_1 = \cos\theta_2 = \cos\theta_3, \quad (3)$$

where θ_1 , θ_2 and θ_3 are the angles between W' and W , W'' and W , and W''' and W , respectively.

4.2. Standardization of the System Model

The system model in Eq. (2) for the general weight vector is a linear combination of the three weight vectors. However, depending on the values of the three coefficients α_i , β_i and γ_i , there would be infinite number of such linear combinations. This complicates the problem solving. To simplify the problem solving, the following Theorem gives a standard form for the system model.

Theorem 1. *A linear combination of three vectors W'_i , W''_i and W'''_i through three non-negative coefficients that are not all zero can be standardized as*

$$W_{ij} = \alpha_i W'_{ij} + \beta_i W''_{ij} + \gamma_i W'''_{ij}, \quad \alpha_i, \beta_i, \gamma_i \in [0, 1], \quad \alpha_i + \beta_i + \gamma_i = 1. \quad (4)$$

315

Proof. Consider a linear combination of three weight vectors through any values of three non-negative coefficients α_i^* , β_i^* and γ_i^* that are not all zero. This gives the system model in Eq. (2). It follows from Eq. (2) that

$$\frac{W_{i,j}^*}{\alpha_i^* + \beta_i^* + \gamma_i^*} = \frac{\alpha_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*} W'_{i,j} + \frac{\beta_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*} W''_{i,j} + \frac{\gamma_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*} W'''_{i,j}. \quad (5)$$

Define

$$W_{i,j} = \frac{W_{i,j}^*}{\alpha_i^* + \beta_i^* + \gamma_i^*}, \quad \alpha_i = \frac{\alpha_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*}, \quad \beta_i = \frac{\beta_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*}, \quad \gamma_i = \frac{\gamma_i^*}{\alpha_i^* + \beta_i^* + \gamma_i^*} \quad (6)$$

Substituting Eq. (6) into Eq. (5) gives the results in Eq. (4). \square

4.3. Nash Equilibrium of the Three Weight Vectors

After the system model is standardized, the game balance can be discussed based on the standardized system model. We have the following theorem on Nash equilibrium for the standardized system model.

Theorem 2. *For the standardized model in Eq. (4), its Nash equilibrium is*

$$\frac{W'_i \cdot W_i}{\|W'_i\|} = \frac{W''_i \cdot W_i}{\|W''_i\|} = \frac{W'''_i \cdot W_i}{\|W'''_i\|}. \quad (7)$$

321

Proof. The angles between the general weight vector W_i and the three weight vectors W'_i , W''_i and W'''_i are respectively given by

$$\cos\theta_1 = \frac{W_i \cdot W'_i}{\|W_i\| \cdot \|W'_i\|}, \quad \cos\theta_2 = \frac{W_i \cdot W''_i}{\|W_i\| \cdot \|W''_i\|}, \quad \cos\theta_3 = \frac{W_i \cdot W'''_i}{\|W_i\| \cdot \|W'''_i\|}. \quad (8)$$

The Nash equilibrium is the point that satisfies Eq. (3), i.e., $\cos\theta_1 = \cos\theta_2 = \cos\theta_3$. From Eqs. (8) and (3), we have the results in Eq. (7). \square

322
323

324 *4.4. Coefficients of the Linear Combination at Nash Equilibrium*

325 With Theorem 2, we are now ready to derive the coefficients for the stan-
 326 dardized linear combination given in Theorem 1 to achieve Nash equilibrium.

Theorem 3. Consider the general weight vector $W_i = (w_{i,0}, \dots, w_{i,j}, \dots, w_{i,m-1})$ and the three weight vectors $W'_i = (w'_{i,0}, \dots, w'_{i,j}, \dots, w'_{i,m-1})$, $W''_i = (w''_{i,0}, \dots, w''_{i,j}, \dots, w''_{i,m-1})$, $W'''_i = (w'''_{i,0}, \dots, w'''_{i,j}, \dots, w'''_{i,m-1})$. When W_i is derived from the linear combination of W'_i , W''_i and W'''_i through Eq. (4), then Nash equilibrium is achieved for the three weight vectors when the three coefficients α_i , β_i and γ_i are set as follows

$$\begin{aligned}\alpha_i &= \frac{1}{2} \cdot \frac{\sqrt{\sum_{j=0}^{m-1} w''_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w'''_{i,j}{}^2}}{\sqrt{\sum_{j=0}^{m-1} w'_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w''_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w'''_{i,j}{}^2}}, \\ \beta_i &= \frac{1}{2} \cdot \frac{\sqrt{\sum_{j=0}^{m-1} w'_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w'''_{i,j}{}^2}}{\sqrt{\sum_{j=0}^{m-1} w'_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w''_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w'''_{i,j}{}^2}}, \\ \gamma_i &= \frac{1}{2} \cdot \frac{\sqrt{\sum_{j=0}^{m-1} w'_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w''_{i,j}{}^2}}{\sqrt{\sum_{j=0}^{m-1} w'_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w''_{i,j}{}^2} + \sqrt{\sum_{j=0}^{m-1} w'''_{i,j}{}^2}}.\end{aligned}\tag{9}$$

327

Proof. From Eq. (7) in Theorem 2, we have $\frac{(\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W'_i}{\|W'_i\|} = \frac{(\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W''_i}{\|W''_i\|} = \frac{(\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W'''_i}{\|W'''_i\|}$. This gives

$$\begin{aligned}(\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W'_i \cdot \|W''_i\| \cdot \|W'''_i\| \\ = (\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W''_i \cdot \|W'_i\| \cdot \|W'''_i\| \\ = (\alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i) \cdot W'''_i \cdot \|W'_i\| \cdot \|W''_i\|.\end{aligned}$$

328 It follows that $\alpha_i = \frac{\|W'_i\| + \|W'''_i\|}{\|W'_i\| + \|W''_i\| + \|W'''_i\|}$, $\beta_i = \frac{\|W'_i\| + \|W''_i\|}{\|W'_i\| + \|W''_i\| + \|W'''_i\|}$, and $\gamma_i =$
 329 $\frac{\|W''_i\| + \|W'''_i\|}{\|W'_i\| + \|W''_i\| + \|W'''_i\|}$. From these relationships, we obtain Eq. (9). \square

330 *4.5. An Example of Using the Three Theorems*

331 Consider the example shown in Table 2. The linear combination given in
 332 Theorem 1 is employed to derive the general weight vector from the space,
 333 energy and data weight vectors. Using the results in Theorem 3, we achieve
 334 Nash equilibrium when setting $\alpha_i = 0.385$, $\beta_i = 0.249$, and $\gamma_i = 0.369$. From
 335 these settings, we further obtained general weights in Table 3, where the boxed
 336 cells with asterisk (*) belong to the same cluster.

337 From Table 3, the node at (2,1) has the maximum general weight 5.89 and
 338 thus is chosen as the cluster core. Then, a multicast tree with a balanced
 339 consideration of all the three factors can be built for data aggregation and
 340 traffic routing in the sensor grid network.

Table 3: The general weight vector W derived from the three weight vectors at Nash equilibrium. The boxed cells with asterisk (*) belong to the same cluster.

	X=0	X=1	X=2	X=3	X=4	X=5	X=6
Y=6	0	0	2.27*	1.71*	3	0	0
Y=5	0	1	3	4	1	0.39*	0
Y=4	0	0	6.29*	6	5	1	0
Y=3	2.31*	1	3	4	3	3	0
Y=2	2	3.238*	5.10*	3	5	7	1
Y=1	1	3	5.89*	3	0	0	3.64*
Y=0	0	1	4	3.11*	0	0	0

341 4.6. Extension for Arbitrary Linear Combination Coefficients

So far, the multi-factor multicast problem has been investigated through a standardized system model given in Eq. (4) in Theorem 1. For a specific application, one may expect to weigh a factor more heavily. Additional weights α_i^+ , β_i^+ and γ_i^+ can be introduced to the space, energy and data factors, where $\alpha_i^+, \gamma_i^+, \gamma_i^+ \geq 0, \alpha_i^+ + \gamma_i^+ + \gamma_i^+ \neq 0$. Thus, we have an extended model as

$$\begin{aligned}
 W_{i,j}^+ &= \alpha_i^* \cdot \alpha_i^+ \cdot W'_{i,j} + \beta_i^* \cdot \beta_i^+ \cdot W''_{i,j} + \gamma_i^* \cdot \gamma_i^+ \cdot W'''_{i,j}, \\
 \alpha_i^*, \beta_i^*, \gamma_i^* &\in [0, 1], \quad \alpha_i^* + \beta_i^* + \gamma_i^* = 1, \\
 \alpha_i^+, \gamma_i^+, \gamma_i^+ &\geq 0, \quad \alpha_i^+ + \gamma_i^+ + \gamma_i^+ \neq 0.
 \end{aligned} \tag{10}$$

Denote

$$\begin{aligned}
 W_{i,j} &= \frac{W_{i,j}^+}{\alpha_i^* \cdot \alpha_i^+ + \beta_i^* \cdot \beta_i^+ + \gamma_i^* \cdot \gamma_i^+}, \quad \alpha_i = \frac{\alpha_i^* \cdot \alpha_i^+}{\alpha_i^* \cdot \alpha_i^+ + \beta_i^* \cdot \beta_i^+ + \gamma_i^* \cdot \gamma_i^+}, \\
 \beta_i &= \frac{\beta_i^* \cdot \beta_i^+}{\alpha_i^* \cdot \alpha_i^+ + \beta_i^* \cdot \beta_i^+ + \gamma_i^* \cdot \gamma_i^+}, \quad \gamma_i = \frac{\gamma_i^* \cdot \gamma_i^+}{\alpha_i^* \cdot \alpha_i^+ + \beta_i^* \cdot \beta_i^+ + \gamma_i^* \cdot \gamma_i^+}.
 \end{aligned} \tag{11}$$

342 Then, the extended system model in Eq. (10) is reduced to the standard model
 343 given in Eq. (4) in Theorem 1. Therefore, the theoretical results derived in
 344 Theorems 1 to 3 can be applied to the extended system scenario.

345 4.7. Extension to More Than Three Dimensions

Foe M -dimensional factors, where $M > 3$, consider M weight vectors $W_{i,j}^1, \dots, W_{i,j}^m$. An extended linear combination of these M weight vectors is defined as

$$\begin{aligned}
 W_{i,j}^* &= \alpha_i^{(1*)} W_{i,j}^{(1)} + \dots + \alpha_i^{(l*)} W_{i,j}^{(l)} + \dots + \alpha_i^{(m*)} W_{i,j}^{(m)}, \\
 \alpha_i^{(1*)}, \dots, \alpha_i^{(l*)}, \dots, \alpha_i^{(m*)} &\geq 0, \quad \sum_{k=1}^m \alpha_i^{k*} \neq 0,
 \end{aligned} \tag{12}$$

346 where $\alpha_i^{(1*)}, \dots, \alpha_i^{(l*)}, \dots, \alpha_i^{(m*)}$ are coefficients.

Similar to the results in Theorem 1, the following standardized system model can be derived for the extended M -dimensional model in Eq. (12):

$$\begin{aligned} W_{i,j} &= \alpha_i^{(1)} W_{i,j}^{(1)} + \dots + \alpha_i^{(l)} W_{i,j}^{(l)} + \dots + \alpha_i^{(m)} W_{i,j}^{(m)}, \\ \alpha_i^{(1)}, \dots, \alpha_i^{(l)}, \dots, \alpha_i^{(m)} &\in [0, 1], \quad \sum_{k=1}^m \alpha_i^k = 1, \end{aligned} \quad (13)$$

where $\alpha_i^{(1)}, \dots, \alpha_i^{(m)}$ are standardized coefficients. Its Nash equilibrium is

$$\frac{W_i \cdot W_i^{(1)}}{\|W_i^{(1)}\|} \dots = \frac{W_i \cdot W_i^{(l)}}{\|W_i^{(l)}\|} \dots = \frac{W_i \cdot W_i^{(m)}}{\|W_i^{(m)}\|}. \quad (14)$$

The corresponding weigh vector coefficients at Nash equilibrium are given by

$$\left\{ \begin{aligned} \alpha_i^{(1)} &= \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(2)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m)2}}}{2 \left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(1)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(l)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m)2}} \right)}, \\ \dots & \\ \alpha_i^{(l)} &= \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(1)2}} \dots \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(l-1)2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(l+1)2}} \dots \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m)2}}}{2 \left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(1)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(l)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m)2}} \right)}, \\ \dots & \\ \alpha_i^{(m)} &= \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(1)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m-1)2}}}{2 \left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(1)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(l)2}} + \dots + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{(m)2}} \right)}. \end{aligned} \right. \quad (15)$$

347 5. Algorithm Design

348 To implement the theoretical results derived above, five algorithms are de-
349 signed in this section: 1) cluster formation algorithm, 2) related weight vectors
350 generation algorithm, 3) least weighted path tree algorithm, 4) multicast routing
351 algorithm, and 5) general finite M -dimensional vectors algorithm.

352 5.1. Cluster Formation Algorithm

353 The cluster formation algorithm clusters the sensor nodes in terms of static
354 delay distance. It is shown in Algorithm 1.

355 Initially, the sensor nodes are split into several clusters through management
356 nodes (MN). The cluster size is usually configured as an integer $S = (k, 3k - 1)$.
357 The expression $(k, 3k - 1)$ represents a random constant between k and $3k - 1$,
358 and a typical value of k is $k = 3$ as in the NICE protocol [14]. After this
359 initial clustering process, the nodes that have not been assigned are marked as
360 unassigned. All unassigned nodes form the node set G , which is the input of
361 the Cluster Formation Algorithm.

362 In the cluster formation algorithm, as long as there are unassigned nodes
363 (Line 1), the algorithm keeps executing the cluster formation process. Each

Algorithm 1: Cluster formation

Input: The RP and unassigned group member set: $G = \{U_0, \dots, U_{l-1}\}$
Output: Cluster set: C ;

```
1 while  $G$  is not empty do
2   while  $ClusterSize \leq 3k - 1$  do
3     // circle for cluster to generate CS-Cluster-No:
4      $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,-1})$ .
5     RP selects the left lowest end host  $U$  in  $G$ ;
6     Add this host  $U$  to CS-Cluster-No and remove it from  $G$ ;
7     for  $j = 0$  to  $m - 1$  do
8       //  $m$  is the # of dimensions
9       RP selects the unassigned closest member in the  $j$ -th
10      dimension;
11      Add this member to CS-Cluster-No and remove it from  $G$ ;
12      for  $i = 0$  to  $j - 1$  do
13        MN selects the closest unassigned member in sub-grid
14         $k_i \times k_j$ ;
15        Adds this member to CS-Cluster-No and remove it from  $G$ ;
16      MN selects the closest unassigned member in grid  $k_i \times k_j$ ;
17      Adds this member to CS-Cluster-No and remove it from  $G$ ;
18    Increment CS-Cluster-No by 1;
```

364 cluster is filled with $3k - 1$ sensor nodes (Line 2). In the clustering, the MN
365 initially selects the left lowest end node (say U) among all unassigned nodes
366 (Line 3). The left lowest node U is the node with the minimum coordinates
367 along m dimensions among all nodes in the unassigned node set G . It is added
368 to the cluster, and removed from the unassigned node set G (Line 4). Then,
369 the same operation is conducted in m dimensions (Lines 5-7). For each of the
370 dimension, also search a sub-grid, add the node to the cluster and marked it as
371 signed (Lines 9 and 10). After that, check the sub-grid including the boundary,
372 add the node to the cluster and remove it from set G (Lines 11 and 12)

373 After the sensor nodes are scattered into different clusters, a tree can be
374 built to connect cluster members with one another. Then, other factors are also
375 considered for optimization of the cluster tree. Different clusters are connected
376 by hooking the tree roots. For example, in Energy factor algorithms LEACH and
377 TEEN, each node needs to rotate the cluster core based on its energy situation.
378 This achieves a balance energy consumption for all nodes in the cluster. LEACH
379 and TEEN deploys a randomized rotation of cluster heads to evenly distribute
380 the energy load among all sensor nodes.

381 5.2. Related Weight Vectors Generation Algorithm

382 This algorithm generates the space, energy and data weight vectors W' , W''
383 and W''' , respectively. It also generates the space, energy and data cores $c_{i,a}$,

384 $c_{i,b}$ and $c_{i,c}$, respectively. The algorithm is illustrated in Algorithms 2 and 3. It
 385 consists of six main steps, as described below in detail.

386 **Step 1). Find the spatial center core $C_{i,a}$ in every cluster C'_i .** This
 387 step is shown in Algorithm 2. Each cluster will have a spatial center node as
 388 the space core. If only the space factor is considered, the space core can be the
 389 root of the multicast tree in the cluster. Consider selecting a spatial core in
 390 each cluster such that the sum of the static delay distances to all other cluster
 391 members is minimized. The following theorem gives a sufficient and necessary
 392 condition for an optimal selection.

Algorithm 2: Relative Weighted Vectors Generation - Step 1)

Input: Cluster Member C ;
Output: The space, energy & data weight vectors W', W'', W''' ;
 The space, energy & data cores: C_a^*, C_b^*, C_c^* ;

1 *Step 1). Find the spatial center node (core) $c_{i,a}$ in every cluster C_i*
 2 **begin**
 3 Initialize $\{a_{\{c_j\}min}, \dots, a_{\{c_j\}t}, \dots, a_{\{c_j\}max}\} = \{0, \dots, 0, \dots, 0\}$;
 // $a_{\{c_j\}t}$ records the # of cluster members whose j -th coordinate
 equals to $(C_j)_t$, $(C_j)_{min} \leq (C_j)_t \leq (C_j)_{max}$ and $0 \leq j \leq m-1$.
 4 **for** $k = 0$ to $n' - 1$ **do**
 5 **if** the j -th coordinate of C_k is equal to $(C_j)_{td}$ **then**
 6 $a_{(c_j)t} \leftarrow a_{(c_j)t} + 1$;
 7 **for** $i = 0$ to $n' - 1$ **do**
 8 **for** $j = 0$ to $m - 1$ **do**
 9 **if** $\left(\left| \sum_{l=(C_i)_{min}}^{C_{i,j}} a_t - \sum_{i=C_{i,j}}^{(C_j)_{max}} a_t \right| \leq a_{(C_{i,j})} \right)$ **then**
 10 $C_j^* \leftarrow C_{i,j}^*$; $j \leftarrow j + 1$;
 11 **else**
 12 $j \leftarrow m - 1$; $i \leftarrow i + 1$;
 13 $C_a^* = \{c_{0,a}^*, \dots, c_{m-1,a}^*\}$;

Theorem 4. Let U be the cluster member that occupies the node $(u_0, \dots, u_j, \dots, u_{m-1})$ in an m -D grid. Also let $n_{>j}$, $n_{<j}$ and $n_{=j}$ be the numbers of cluster members with the J -th coordinates greater than (right nodes of J -th row), less than (left nodes of J -th row), and equal to u_j (nodes just on the J -th row), respectively. Then, U is the spatial center node if and only if the following inequalities hold simultaneously:

$$|n_{<j} - n_{>j}| \leq n_{=j}, \quad j = 0, 1, \dots, m-1. \quad (16)$$

393

394 *Proof.* Assume $U = (u_0, \dots, u_j, \dots, u_{m-1})$ is a spatial center node. Then, for
 395 any member U' in the sensor grid network, there exists static delay distance

396 $f(U) \leq f(U')$. To achieve inequalities in (16), we firstly consider a node $U' =$
397 $(u_0, \dots, u_{j+1}, \dots, u_{m-1})$ and its multicast static delay distance $f(U')$. Given
398 any member $U_i = (u_{i,0}, \dots, u_{i,j}, \dots, u_{i,m-1})$ and $u_j \leq u_{i,j}$, the distance from
399 U_i to the end host U is one unit longer than that from U_i to the node U' .

400 Similarly, it is seen that to any member $U_i = (u_{i,0}, \dots, u_{i,j}, \dots, u_{i,m-1})$
401 and $u_{i,j} \leq u_j$, the distance from U_i to the end host U is one unit shorter than
402 the distance from U_i to U' . There exist $(n_{>u_j} + n_{=u_j})$ members whose J -th
403 coordinates are larger than or equal to u_j , and $n_{<u_j}$ cluster members whose
404 J -th coordinates are less than u_j .

405 Then, it is concluded that $0 \leq f(u') - f(u) = \sum_{j=0}^{n'} (d(u', u_j) - d(u, u_j))$
406 $= n_{>u_j} + n_{=u_j} - n_{<u_j}$. This gives $n_{<u_j} - n_{>u_j} \leq n_{=u_j}$. By comparing
407 $f(u_0, \dots, u_{j-1}, \dots, u_{m-1})$ with $f(U)$ in the same way as above, the inequalities
408 in (16) can be achieved. The condition in Theorem 4 is a sufficient condition.

409 It is easy to demonstrate that if inequalities in (16) are violated, then U
410 cannot be the spatial center nodes. Assume $n_{>u_j} - n_{<u_j} > n_{=u_j}$. Then, $n_{>u_j} >$
411 $n_{<u_j} + n_{=u_j}$. Let us firstly consider node $U' = (u_0, \dots, u_{j+1}, \dots, u_{m-1})$ and
412 its multicast static delay distance $f(U')$. Given any member $U_i = (u_{i,0}, \dots,$
413 $u_{i,j}, \dots, u_{i,m-1})$ and $u_j \leq u_{i,j}$, the distance from U_i to the end host U is one
414 unit longer than the distance from U_i to U' . Similarly, it is also seen that to
415 any member $U_i = (u_{i,0}, \dots, u_{i,j}, \dots, u_{i,m-1})$ and $u_{i,j} \leq u_j$, the distance from
416 U_i to the node U is one unit shorter than the distance from U_i to U' . Thus,
417 $f(u) - f(u') = \sum_{i=0}^{n'} (d(u, u_i) - d(u', u_i)) = n_{<u_j} + n_{=u_j} - n_{>u_j} > 0$. This
418 means $f(u) > f(u')$. Therefore, the distance from U to those end nodes is
419 longer than to some other end nodes. However, this is a desired contradiction.
420 The condition in Theorem 4 is a necessary condition. \square

421 The physical meaning of the theorem is shown in Fig. 3. For the sensor grid
422 in the figure, we process the nodes on X axis first. For example, $N_{=2} = 4$,
423 indicating there are 4 nodes on the second column: $(2, 6), (2, 4), (2, 2), (2, 1)$.
424 $N_{<2} = 2$, implying that there are 2 nodes to the left of the second row: $(0, 3),$
425 $(1, 2)$. $N_{>2} = 4$, meaning that there are 4 nodes to the right of second row
426 $(3, 6), (3, 0), (5, 5), (6, 1)$. It follows that $|n_{<2} - n_{>2}| \leq n_{=2}$. The condition in
427 Theorem 4 is satisfied. Thus, $N_{=2}$ is an optimal choice for the core on the X
428 axis. Similarly, the same process is applied to Y axis. Considering the results
429 from both the X and Y axes, we find that node $(2, 2)$ is the best central core.

430 It is worth mentioning that in Step 1) of Algorithm 2, Lines 4-6 can be
431 executed in time complexity $O(n)$. Lines 7-12 can be improved by using a
432 binary search algorithm that yields $O(\ln(n))$ complexity. In our example, linear
433 search has been demonstrated for simplicity.

434 **Step 2).** Calculate the space weight vector $W'_{i,j}$ for every node
435 (Lines 1-10 in Algorithm 3). Initially, when considering the space factor only,
436 the system establishes a multicast tree to transmit data packets. The tree is
437 established according to the space weights of the sensor nodes. The root of the
438 tree is the space root. The tree should maximize the sharing of the utilization

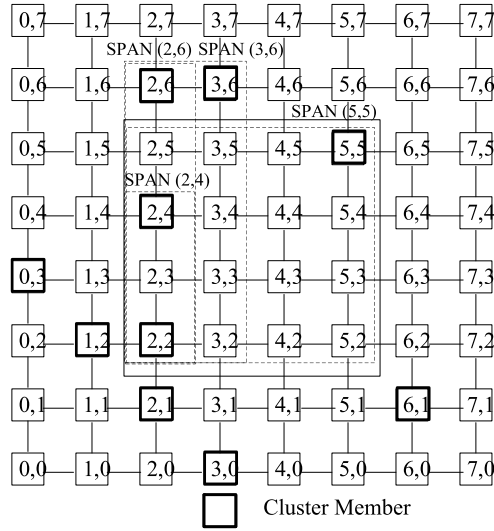


Figure 3: Shortest Path Area Nodes (SPAN) in a 2-D sensor grid networks. For example: the node (2; 4) is 4 node's SPAN: (2; 6), (3; 6), (5; 5), (2; 4). The rectangle means shortest path area, and the space core is (2; 2), so that the node (2; 4) is in the rectangles of 4 nodes of the space core.

439 of the link within the clusters. Our algorithm uses the following concepts:

- 440 1) *Shortest Path Area Nodes (SPANs)*: For any two nodes (x_0, y_0) and (x_1, y_1) ,
441 let $X_{min} = \min\{x_0, x_1\}$, $X_{max} = \max\{x_0, x_1\}$, $Y_{min} = \min\{y_0, y_1\}$ and
442 $Y_{max} = \max\{y_0, y_1\}$. They uniquely define a rectangle area $[x_0, y_0] \times$
443 $[x_1, y_1]$. Each node (x, y) is in $[x_0, y_0] \times [x_1, y_1]$. If the node is on one of
444 the shortest paths between (x_0, y_0) and (x_1, y_1) , it is referred to as the
445 shortest path area node (SPAN) between (x_0, y_0) and (x_1, y_1) .
- 446 2) *SPANs of a cluster*: When a tree is built in the cluster of size n , all nodes
447 $c_i(x_i, y_i)$, $(i \in [0, n-1])$ in the SPAN area $[x_0, y_0] \times [x_i, y_i]$ covering the core
448 (i.e., the root of the tree) $c^*(x^*, y^*)$ are SPAN nodes of c_i . For example, in
449 Fig. 3, if the core is the node (2, 2), all nodes in the area $[2, 2] \times [5, 5]$ are
450 the SPAN nodes of this cluster member (core). A node may be a SPAN
451 node of multiple cluster members.
- 452 3) *The space weight of a node*: If a node is a SPAN node of k cluster members,
453 it is assigned a weight k . The upper part of Table 3 lists the space weights
454 of all nodes in Fig. 2(a). As an example, in Fig. 3, the node (2, 4) is in the
455 SPAN area of four nodes (2, 6), (3, 6), (5, 5), (2, 4). Thus, it is a SPAN
456 of those four nodes. The weight of this node (2, 4) is 4, indicating that 4
457 cluster members may pass through node (2, 4) to the cluster core (2, 2) by
458 the shortest paths. For node (2, 2), its weight is 10.

459 In general, the greater the space weight is, the nearer the node is to the
460 core. If the space weight of a node is k , then there are k nodes that must pass

Algorithm 3: Relative Weighted Vectors Generation - Steps 2) to 6)

1 *Step 2). Calculate the space weight vector W'_j for every node*

2 **begin**

3 Initialize $T \leftarrow \{\}$; For any node $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,m-1})$ with
 $(C_j)_{min} \leq (C_j)_t \leq (C_j)_{max}$, initialize its weight $W'_{c,j} \leftarrow 0$;

4 **for** $j = 0$ to $n' - 1$ **do**

5 **for** $i = 0$ to $n' - 1$ **do**

6 **if** C_i is a SPAN node of $C_j = (C_{j,0}, C_{j,1}, \dots, C_{j,m-1})$ **then**

7 $W'_{c,j} \leftarrow W'_{c,j} + 1$;

8 $W' = \{W'_0 = (w'_{0,0}, w'_{0,1}, \dots, w'_{0,m-1}), \dots,$

9 $W'_i = (w'_{i,0}, w'_{i,1}, \dots, w'_{i,m-1}), \dots,$

10 $W'_{n'-1} = (w'_{n'-1,0}, w'_{n'-1,1}, \dots, w'_{n'-1,m-1})\}$, $i = 0, \dots, n' - 1$;

11 *Step 3). Find the energy weight $W''_{i,j}$ for every node in cluster c_j*

12 **begin**

13 In all members $C = \{C_0 = (C_{0,0}, C_{0,1}, \dots, C_{0,m-1}), \dots,$
 $C_i = (C_{i,0}, C_{i,1}, \dots, C_{i,m-1}), \dots, C_{n'-1} = (C_{n'-1,0}, C_{n'-1,1},$
 $\dots, C_{n'-1,m-1})\}$, $i = 0, \dots, n' - 1$, search the energy weight vector
 W'_j for every node;

14 $W'' = \{W''_0 = (w''_{0,0}, w''_{0,1}, \dots, w''_{0,m-1}), \dots,$
 $W''_i = (w''_{i,0}, w''_{i,1}, \dots, w''_{i,m-1}), \dots,$
 $W''_{n'-1} = (w''_{n'-1,0}, w''_{n'-1,1}, \dots, w''_{n'-1,m-1})\}$, $i = 0, \dots, n' - 1$;

15 *Step 4). Find the max energy node in $W''_{i,j}$ in each core $C_{i,b}$ of every cluster*

16 Find the maximal energy node in $W''_{i,j}$, $C_b^* = (c_{0,b}^*, \dots, c_{m-1,b}^*)$;

17 *Step 5). Find the data weight $W'''_{i,j}$ for every node in cluster C_j*

18 **begin**

19 In all members $C = \{C_0 = (C_{0,0}, C_{0,1}, \dots, C_{0,m-1}), \dots,$
 $C_i = (C_{i,0}, C_{i,1}, \dots, C_{i,m-1}), \dots,$
 $C_{n'-1} = (C_{n'-1,0}, C_{n'-1,1}, \dots, C_{n'-1,m-1})\}$, $i = 0, \dots, n' - 1$,
 search the data weight vector of every node W'_j ;

20 $W''' = \{W'''_0 = (w'''_{0,0}, w'''_{0,1}, \dots, w'''_{0,m-1}), \dots,$
 $W'''_i = (w'''_{i,0}, w'''_{i,1}, \dots, w'''_{i,m-1}), \dots,$
 $W'''_{n'-1} = (w'''_{n'-1,0}, w'''_{n'-1,1}, \dots, w'''_{n'-1,m-1})\}$, $i =$

21 $0, \dots, n' - 1$;

22 *Step 6). Find the node with the maximal data weight $W'''_{i,j}$ as the data core $C_{i,c}$ in every cluster*

23 Find the maximal data node in $W'''_{i,j}$, $C_c^* = (c_{0,c}^*, \dots, c_{m-1,c}^*)$;

461 through this node to the space core for data transmissions. Thus, the weight of
462 the node represents the degree of the closeness of the node to the core.

463 **Step 3). Find the energy weight W'' for every node in cluster C_i**
464 (Lines 11-14 in Algorithm 3). After the space weight vector $W'_{i,j}$ is determined,
465 the system periodically checks the energy status of every node. Then, it gener-
466 ates the energy weight accordingly. An example of energy weights is shown in
467 the middle part of Table 3.

468 **Step 4). Find the node with the maximal energy weight in $W''_{i,j}$**
469 **as the energy core $c_{i,b}$ in every cluster** (Lines 15 and 16 in Algorithm 3).
470 From the energy weights in the middle part of Table 3, node $B(2,4)$ has the
471 maximal energy weight 10. It is chosen as the energy core.

472 **Step 5). Find the data weight W''' for every node in cluster C_i**
473 (Lines 17 to 21 in Algorithm 3). The system generates data weights in a similar
474 way to that for energy weight generation in Step 3). The resulting data weights
475 are shown in the lower part of Table 3.

476 **Step 6). Find the node with the maximal data weight in $W'''_{i,j}$ as the**
477 **data core $c_{i,c}$ in every cluster** (Lines 22 and 23 in Algorithm 3). According
478 to the data weights in the lower part of Table 3, node $B(2,1)$ has the maximal
479 value data weight 10. Therefore, it is selected as the data core.

480 5.3. Least weighted path tree generation algorithm

481 From Algorithms 2 and 3, the related weighted vectors W' , W'' and W'''
482 have been generated. The space, energy, and data cores $c_{i,a}$, $c_{i,b}$ and $c_{i,c}$ have
483 also been found. Then, the least weighted path tree generation algorithm inte-
484 grates W' , W'' and W''' to form a single weight vector $W = f(W', W'', W''')$.
485 As discussed in Section 4, a linear combination of the three weight vectors is
486 used in our work, i.e., $W = \alpha W' + \beta W'' + \gamma W'''$, as described in the standardized
487 system model in Eq. (4) in Theorem 1. Three linear combination coefficients
488 α , β , and γ are determined at Nash equilibrium according to Theorems 2 and
489 3. Finally, the algorithm presents the least weighted path tree as a hierarchical
490 multicast tree. The whole process of the least weighted path tree generation
491 algorithm is described in Algorithm 4, which is self-explained.

492 5.4. Multicast Routing algorithm

493 Following the determination of the general weight vector and its coefficients
494 in Algorithm 4, the multicast routing algorithm is activated. Shown in Algo-
495 rithm 5, the multicast routing algorithm builds a multicast tree for each cluster
496 to connect the cluster members. After that, the algorithm connects different
497 clusters by hooking the roots of the trees.

498 5.5. General Finite M -dimensional Vector Algorithm

499 Extending the three-factor scenario to finite M -dimensional factors, the the-
500 oretical results developed in Section 4.7 are also implemented. The implemen-
501 tation is described in a General Finite M -dimensional Vector Algorithm, as
502 shown in Algorithm 6, which is self-explained. For every cluster, the algorithm

Algorithm 4: Least Weighted Path Tree Generation

Input: The space, energy and data weight vectors W', W'', W''' ;

The space, energy and data cores: C_a^*, C_b^*, C_c^* ;

Output: Tree $T\{\}$

```
1 begin
2   Tree  $T \leftarrow \{\}$ , node weight  $W \leftarrow 0$ ;           /* Initialization */
3   for  $i = 0$  to  $n - 1$  do
4     Calculate  $\alpha_i, \beta_i$  and  $\gamma_i$  from Eq. (9);
5     Calculate  $W_i = \alpha_i W'_i + \beta_i W''_i + \gamma_i W'''_i$ ;
6     Choose the node with the maximal weight in  $W_i$  in every cluster;
7     Set this node as cluster core  $C^* = (c_0^*, \dots, c_{m-1}^*)$ ;
8     for  $i = 0$  to  $n - 1$  do
9       Select the shortest path  $P = \langle (C_0^*, \dots, C_{(m-1)}^*), \dots, (C_{i,0}^*, \dots,$ 
           $C_{i,(m-1)}^*) \rangle$  with the maximum weight; add this path to tree  $T$ ;
```

Algorithm 5: Multicast Routing

- 1 Source node s sends its multicast messages to its cluster core c_0 ;
 - 2 The cluster core c_0 sends the messages to all other cores c_i ;
 - 3 c_0 routes the messages to its own cluster members along the cluster tree;
 - 4 Upon receiving the multicast messages, all cluster cores c_i transmit them along the cluster trees to all cluster members m_i within the clusters.
-

Algorithm 6: General Finite M-Dimensional Vectors

- 1 Divide the group members into clusters in terms of static delay distance;
 - 2 Calculate the finite M-dimensional weight vector $W_{i,j}^{(1)}, \dots, W_{i,j}^{(m)}$;
 - 3 Build a game balance relationship equation, resolve linear parameters, make out a new weight vector according to the algebraic sum of the M-dimensional known vectors, and generate the least weighted path tree: $\alpha_i^{(1)}, \dots, \alpha_i^{(m)}$;
 - 4 Dispatch the multicast packets in the group on the basis of the tree.
-

503 calculates the finite M-dimensional weight vectors. Then, for each cluster, it
504 generates the least weighted path tree. The trees for all clusters are connected
505 to form a complete tree for dispatching multicast packets.

506 6. Performance Evaluation

507 This section evaluates the performance of our game balance based approach.
508 It starts with descriptions of evaluation criteria and benchmark methods [33].

509 Then, experimental configurations, settings and scenarios are discussed. After
510 that, experimental results are presented under various scenarios.

511 *6.1. Evaluation Criteria and Benchmark Methods*

512 Three performance metrics are used to quantify the performance of data
513 aggregation and multicast routing. They are average multicast delay, average
514 number of used links, and average packet arrival rate. The delay performance
515 characterizes the timeliness of the communications. The number of used links
516 for message multicast to all group members marks the amount of used resources.
517 The packet arrival rate implies the reliability of data transmissions.

518 Our game balance based approach will be compared with some benchmark
519 methods. The popularly used DCXYP, which considers the space factor only,
520 is chosen as the SPACE-based benchmark method. The well adopted LEACH
521 and TEEN methods, which consider the energy factor only, are selected as
522 ENERGY-based benchmark methods. The DATA-based benchmark method is
523 designed to consider the data factor only with involvement of data quantity,
524 query hot degree and semantic cache value.

525 *6.2. Experimental Configurations, Setting and Scenarios*

526 Our experiments consider a 2D sensor grid network in an office building
527 environment. There are 360 mobile phones in the network. Each cluster includes
528 eight mobile phones. The network system will build a multicast tree to link all
529 these mobile phones for query message transmission.

530 Other network configurations and settings are given below. The system runs
531 on a group of 40 IBM four-core PCs. The dataset is transferred to a 2D grid
532 configuration of 1 km by 1 km. The number of mobile devices varies from 0 to
533 200, and the locations of those mobile devices are randomly generated. The data
534 (weight) for each of the nodes is also randomly generated in the range from 1 to
535 10 units by following the Poisson distribution model. Similarly, energy weight
536 for each node is randomly generated as well, and it decays over time based
537 on the operation of the node. The underlying routing protocol is AODV. The
538 bandwidth of the wireless communications is 10 Mbps for each link. During the
539 simulation, 100 and 10,000 multicast packets are randomly generated for light
540 and heavy traffic load, respectively. The average size of the packets is 2,400
541 bytes. From the packet generation, the average delay for transmitting a packet
542 on a defined link is about 1 sec. As the system is highly dynamic, a periodic
543 calculation of the multicast tree is necessary with the period being defined as 60
544 sec. Furthermore, the *Time_energy_unite*, which means the operating duration
545 rate that an energy unit can support, is set to be 10.

546 To investigate the energy factor, we use a variable, *Energy_degree*, to quan-
547 tify the level of energy in each node. In the experiments, *Energy_degree* is de-
548 signed to have 10 levels from 1 to 10. The system sets a model *Time_Energy_Threshold*
549 $TET = Energy_degree^{Weight_energy} * Time_energy_unite$, where *Weight_energy*
550 is the value of energy the node attaining. Once $t > TET$ is reached, the node
551 is withdrawn from the system.

552 Another variable, `Proportion_degree`, is also introduced to represent the de-
553 gree of difference of the proportions of multiple factors. It is characterized by 10
554 levels from 1 to 10. In this paper, only the following three levels are presented:
555 1) `Proportion_level 1`: 45% of space, 10% of energy, 45% of data;
556 2) `Proportion_level 3`: 33% of space, 33% of energy, 33% of data; and
557 3) `Proportion_level 5`: 25% of space, 50% of energy, 25% of data.
558 `Proportion_level 3` gives an equal weight for all three factors.

559 *6.3. Experimental Results under Different Proportion_degree Values*

560 The first set of experiments is carried out under a fixed `Energy_degree` of
561 3. The `Proportion_degree` takes the values of 1, 3 and 5, as specified above.
562 Light and heavy traffic scenarios are considered in evaluating the five types of
563 methods: *SPACE*, *ENERGY*, *DATA*, our Previous work [9], and Our work
564 in this paper. The results are shown in Figs. 4 through to 9.

565 Figs. 4 and 5 show Average Multicast Delay for the five types of methods.
566 It is seen from these figures that our approach in this paper displays mostly
567 smaller multicast delays on average in both light and heavy traffic scenarios.

568 The results of Average Number of Used Links and Average Packet Arrival
569 Rate are depicted in Figs. 6 and 7, and Figs. 8 and 9, respectively, in light and
570 heavy traffic conditions. These figures show that our approach in this paper
571 clearly outperforms the others types of methods in the sense that it uses fewer
572 links and also achieves high packet arrival rate performance.

573 *6.4. Experimental Results under Different Energy_degree Levels*

574 The second set of experiments is conducted under a fixed `Proportion_degree`=
575 3 (an equal weight for all three factors). The `Energy_degree` takes values of 3,
576 5 and 7. Five different types of methods are compared: *SPACE*, *ENERGY*,
577 *DATA*, Previous work [9] and Our work in this paper. The results are shown in
578 Figs. 10 through to 12 for heavy traffic conditions. Simulation results for light
579 traffic conditions are omitted here because the the conclusions drawn from the
580 results are mostly similar to those drawn from heavy traffic scenarios.

581 Among these five methods, the *DATA* approach is much poorer than the
582 other four methods. *DATA* usually suits the scenarios in which data quantity
583 is much more concentrated in a few important nodes. For better observation of
584 the experiment results, we have omitted *DATA* in the figures.

585 Fig. 10 presenta the results of average multicast delay under light and heavy
586 traffic load scenarios, respectively. It is seen from these two figures that our
587 approach mostly outperforms the other types of methods in terms of the Average
588 Multicast Delay performance.

589 The results of Average Number of Used Links are depicted in Fig. 11. They
590 shows that in both light and heavy traffic load conditions, our approach mostly
591 uses fewer links on average than the other methods do.

592 Fig. 12 demonstrate the results of Average Packet Arrival Rate. It is ob-
593 served from the figures that in both light and heavy traffic load environments,
594 our approach is mostly superior to the other methods in the sense that it achieves
595 higher average packet arrival rates over the time.

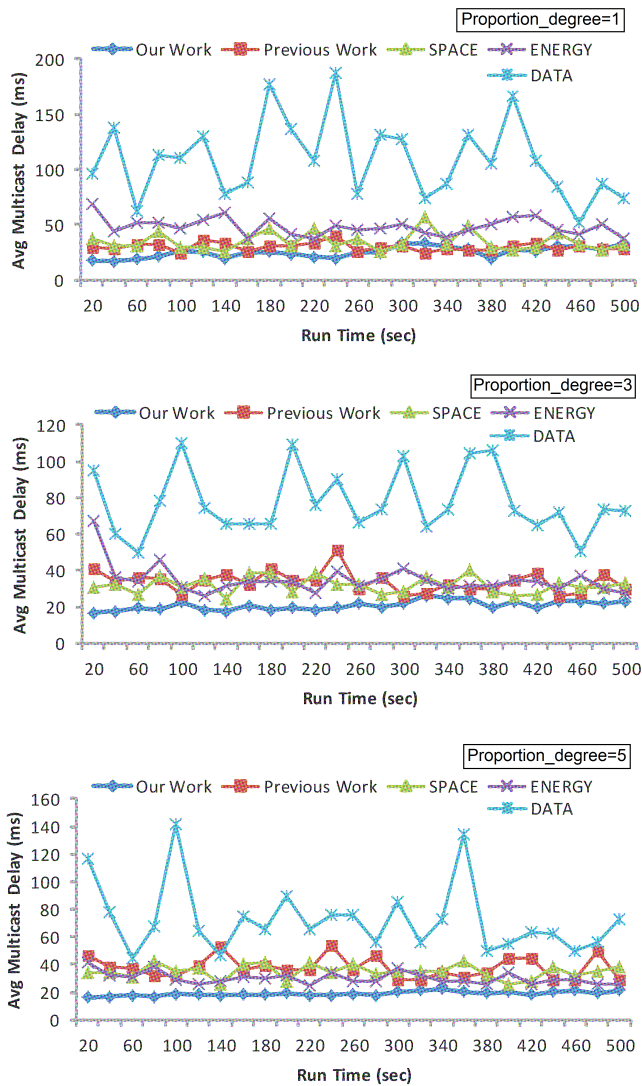


Figure 4: Average Multicast Delay performance in *light traffic condition* for five types of methods: *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and our work in this paper (from top to bottom: Proportion_degree=1, 3 and 5, respectively).

596 **7. Conclusion**

597 An effective and efficient multicast routing approach has been presented in
 598 this paper. A significant feature of the approach is the simultaneous consid-
 599 eration of three factors of space, energy and data. This makes the approach
 600 multi-factor aware. Integrating all the three factors into a unified model, the
 601 approach has also derived a theoretical solution to the multicast problem from

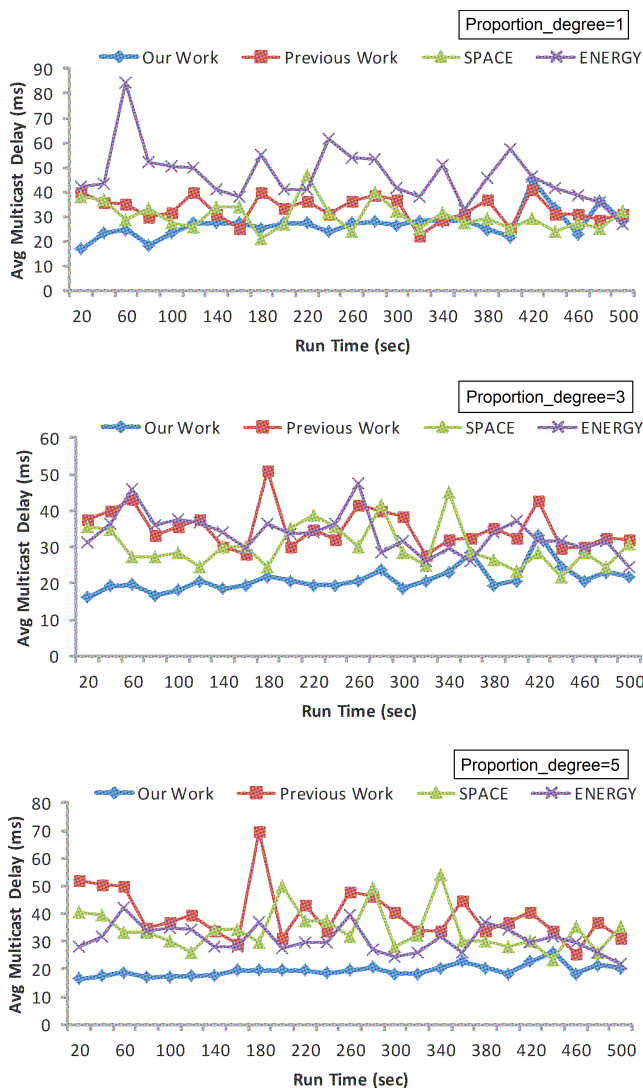


Figure 5: Average Multicast Delay performance in *heavy traffic condition* for four types of methods: *SPACE*, *ENERGY*, our Previous work [9] and our work in this paper (from top to bottom: Proportion_degree=1, 3 and 5, respectively).

602 the game balance perspective, particularly at Nash Equilibrium. This ensures
 603 a fair treatment of all the three factors without a bias to a specific factor.
 604 Moreover, the approach can be easily extended to simpler or more complicated
 605 scenarios, such as two factors, more than three factors, and a heavier weight to
 606 a specific factor. All these features make the presented approach distinct from
 607 existing methods.

608 The theoretic results and algorithms of the resented approach have been

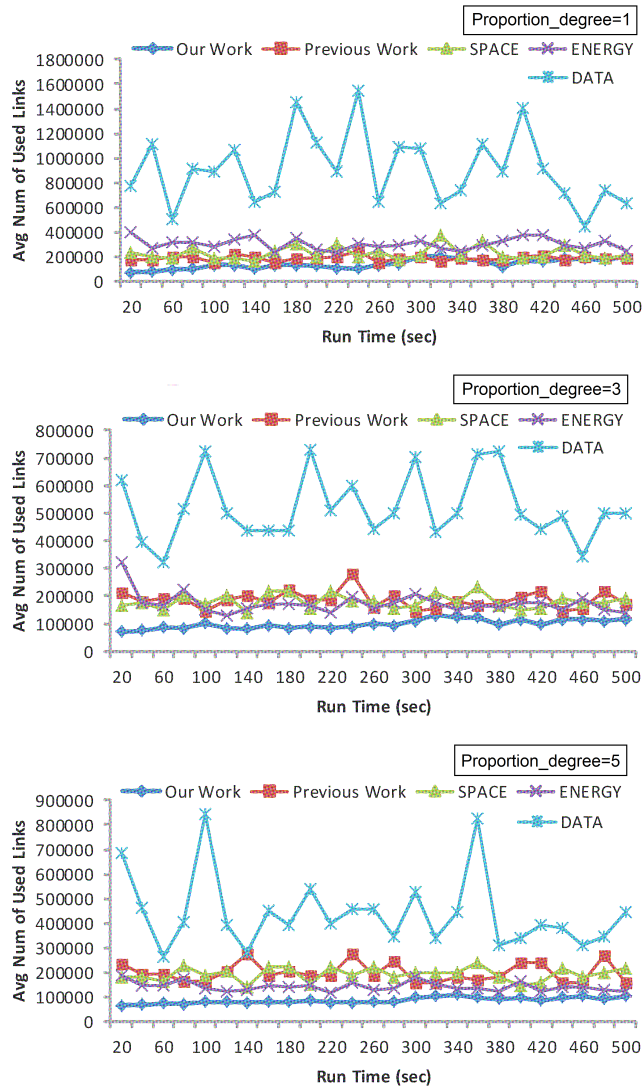


Figure 6: Average Number of Used Links in *light traffic load* for five types of methods: *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and our work in this paper (from top to bottom: Proportion_degree=1, 3 and 5, respectively).

609 evaluated comprehensively. Three criteria have been used to characterize the
610 performance of our approach and benchmark methods. In some extreme con-
611 ditions, existing methods designed exclusively for such conditions show advan-
612 tages. However, in practical applications, network environments, energy level
613 and traffic load change over time. Considering all these dynamic changes, the
614 approach presented in this paper has demonstrated mostly better performance

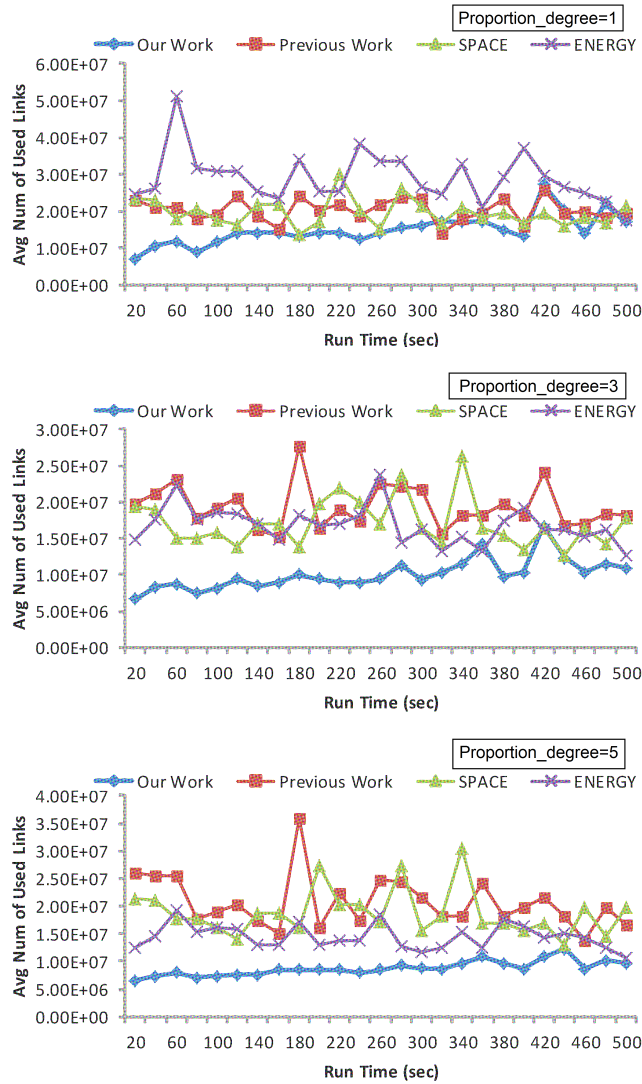


Figure 7: Average Number of Used Links in *heavy traffic load* for four types of methods: *SPACE*, *ENERGY*, our Previous work [9] and our work in this paper (from top to bottom: *Proportion_degree*=1, 3 and 5, respectively).

615 than existing methods under various conditions. It is a promising tool for data
 616 communications in wireless and mobile sensor grid networks.

617 Acknowledgements

618 Authors QF and QW would like to acknowledge Prof. Weijia Jia from City
 619 University of Hong Kong, and Prof. Patrick Valduriez from the Universite de

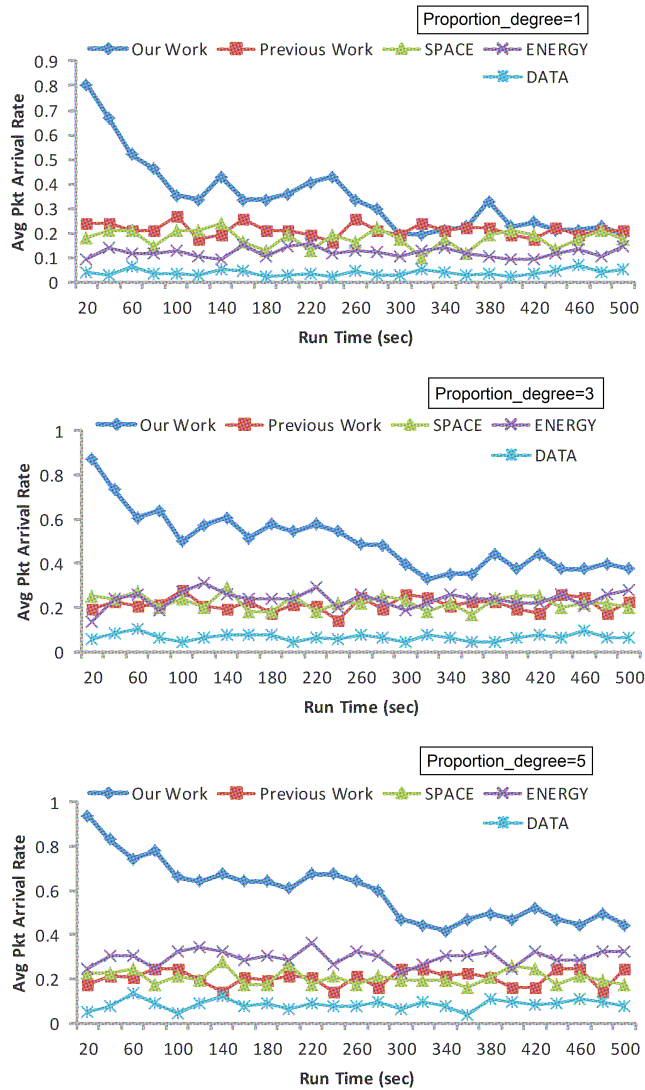


Figure 8: Average Packet Arrival Rate in *light* traffic load for five types of methods: *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and our work in this paper (from top to bottom: Proportion_degree=1, 3 and 5, respectively).

620 Nantes, for their useful discussions. Authors QF, KZ and QU are also grateful
 621 to SAP Business Objects in France for its partial financial support. Author Y.-
 622 C. Tian thanks the Australian Research Council (ARC) for its support under
 623 the Discovery Projects Scheme (Grant ID: DP1601025871).

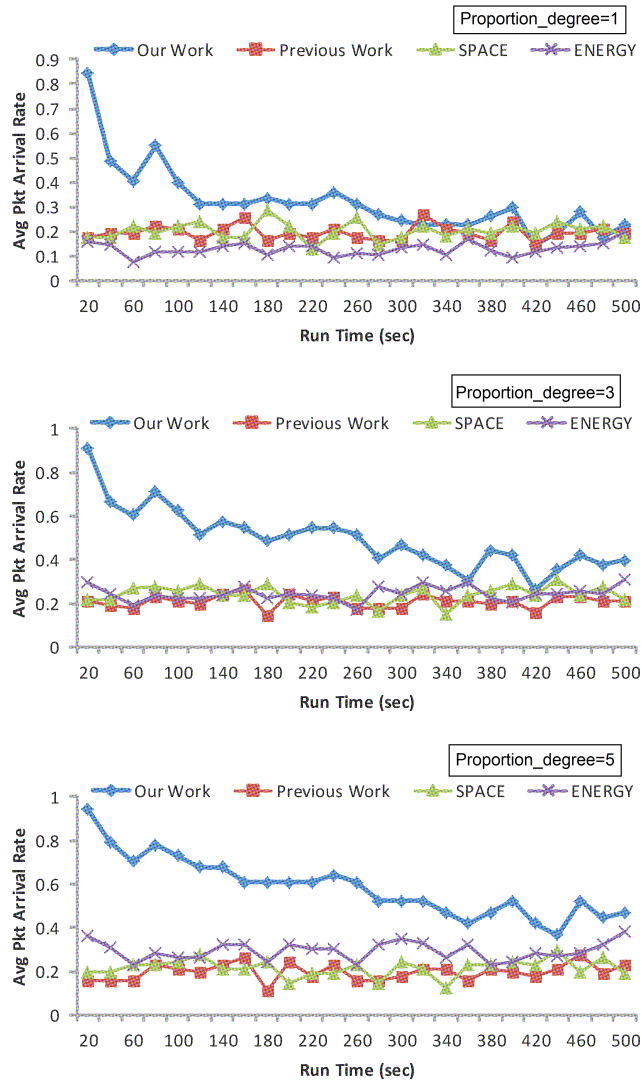


Figure 9: Average Packet Arrival Rate in *heavy* traffic load for four types of methods: *SPACE*, *ENERGY*, our Previous work [9] and our work in this paper (from top to bottom: Proportion_degree=1, 3 and 5, respectively).

624 References

- 625 [1] H. Zhuge. A scalable p2p platform for the knowledge grid. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1721–1736, 2005.
- 626
- 627 [2] I. J. Perez, F. J. Cabrerizo, and E. Herrera-Viedma. A mobile decision support system for dynamic group decision-making problems. *IEEE Trans-*
- 628

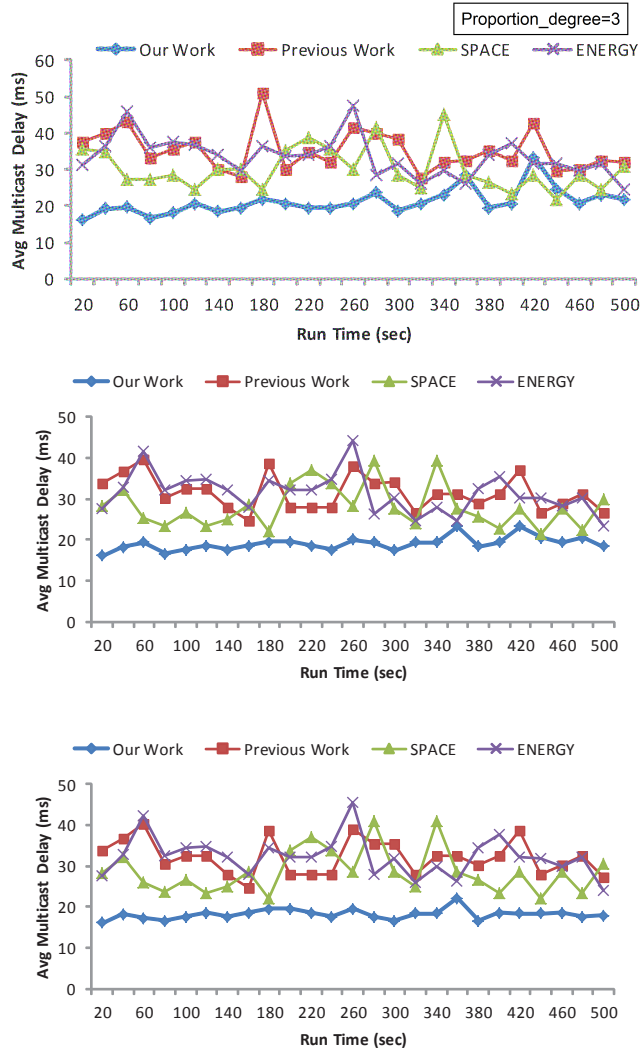


Figure 10: Average Multicast Delay under *heavy* traffic load and Proportion_degree = 3 for different methods *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and Our work in this paper (from top to bottom: Energy_degree=3, 5 and 7, respectively).

629 *actions on Systems Man and Cybernetics, Part A: Systems and Humans*,
 630 pages 1244 – 1256, 2010.

631 [3] S. Ghosh, S. Rao, and B. Venkiteswaran. Sensor network design for smart
 632 highways. *IEEE Transactions on Systems, Man and Cybernetics, Part A:
 633 Systems and Humans*, 24:1291 – 1300, 2012.

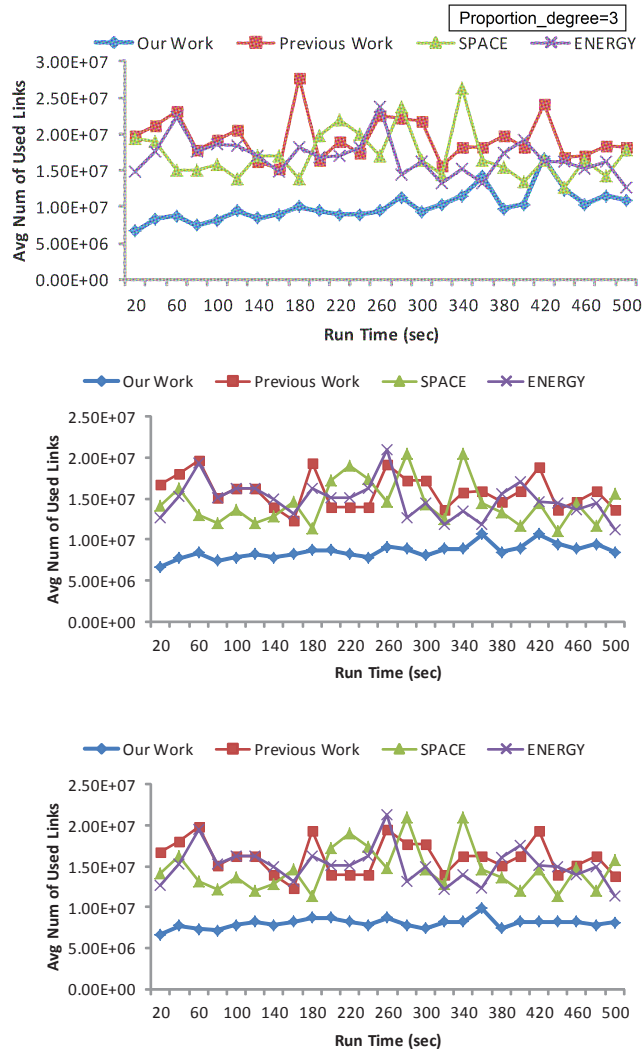


Figure 11: Average Number of Used Links under *heavy* traffic load and $\text{Proportion_degree} = 3$ for different methods *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and Our work in this paper (from top to bottom: $\text{Energy_degree}=3, 5$ and 7 , respectively).

- 634 [4] J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee. A novel routing proto-
635 col providing good transmission reliability in underwater sensor networks.
636 *Journal of Internet Technology*, 16:171–178, 2015.
- 637 [5] C. Chow, H. Leong, and A. Chan. Grococa: group-based peer-to-peer co-
638 operative caching in mobile environment. *IEEE Journal on Selected Areas*
639 *in Communications*, 25:179 – 191, 2007.

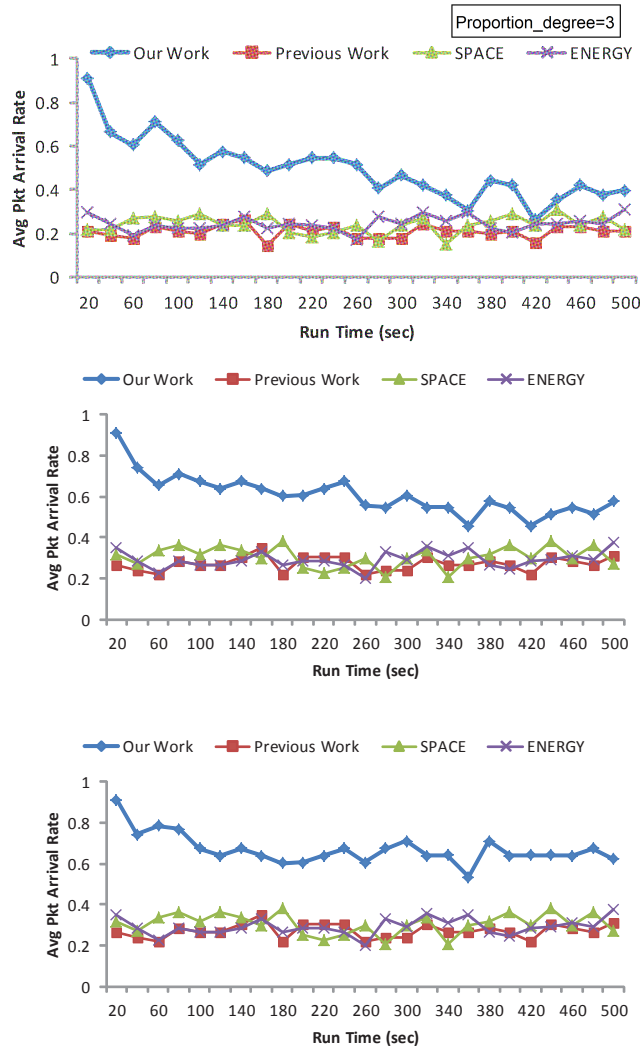


Figure 12: Average Packet Arrival Rate under *heavy* traffic load and Proportion_degree = 3 for different methods *SPACE*, *ENERGY*, *DATA*, our Previous work [9] and Our work in this paper (from top to bottom: Energy_degree=3, 5 and 7, respectively).

- 640 [6] X. Liu, A. Ghorpade, Y.L. Tu, and W.J. Zhang. A novel approach to
641 probability distribution aggregation. *Information Sciences*, 200:269–275,
642 2012.
- 643 [7] A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay
644 networks. In *Proc. of ACM SIGCOMM*, pages 11–18, 2003.
- 645 [8] X. Wen, L. Shao, Y. Xue, and W. Fang. A rapid learning algorithm for

- 646 vehicle classification. *Information Sciences*, 295:395–406, 2015.
- 647 [9] Q. Fan, Q. Wu, F. Magoules, N.Xiong, A.V. Vasilakos, and Y. He. Game
648 and balance multicast architecture algorithms for sensor grid. *Sensors*,
649 9:7177–7202, 2009.
- 650 [10] Q. Fan, Q. Wu, N.Xiong, A.V. Vasilakos, and Y. He. 3 vectors game
651 and balance multicast architecture algorithms for sensor grid. In *Proc. of*
652 *INFOCOM Workshop on Wireless Sensor, Actuator and Robot Networks*,
653 pages 566–571, 2011.
- 654 [11] D. Agrawal. TEEN: a routing protocol for enhanced efficiency in wire-
655 less sensor networks. In *Proc. 15th Int. Parallel & Distributed Processing*
656 *Symposium*, pages 2009–2015, Apr 2001.
- 657 [12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An applica-
658 tion specific protocol architecture for wireless microsensor networks. *IEEE*
659 *Transaction on Wireless Networking*, 1(4):660–670, 2002.
- 660 [13] B. Xu, F. Vafaei, and O. Wolfson. In-network query processing in mobile
661 p2p databases. In *Proc. 17th ACM SIGSPATIAL Int. Conf. on Advances*
662 *in Geographic Information Systems*, pages 207–216, 2009.
- 663 [14] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application
664 layer multicast. In *Proc. of ACM SIGCOMM*, pages 205–217, 2002.
- 665 [15] Y. Shavitt and T. Tankel. Big-bang simulation for embedding network
666 distances in euclidean space. In *Proc. of IEEE INFOCOM*, pages 993–
667 1006, 2004.
- 668 [16] S. Xie and Y. Wang. Construction of tree network with limited delivery
669 latency in homogeneous wireless sensor networks. *Wireless Personal Com-*
670 *munications*, 78:231–246, 2014.
- 671 [17] X. Ruan, S. Yin, A. Manzanares, M. Alghamdi, and X. Qin. A message-
672 scheduling scheme for energy conservation in multimedia wireless systems.
673 *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE*
674 *Transactions on*, 41:272 – 283, 2011.
- 675 [18] M. Saleem, I. Ullah, and M. Farooq. Beesensor: An energy-efficient and
676 scalable routing protocol for wireless sensor networks. *Information Sci-*
677 *ences*, 200:38–56, 2012.
- 678 [19] H. Zhang, J. Kurose, and D. Towsley. Can an overlay compensate for a
679 careless underlay? In *Proc. of IEEE INFOCOM*, pages 1–12, 2005.
- 680 [20] P. A. Bernstein, A. Fekete, H. Guo, R. Ramakrishnan, and P. Tamma.
681 Relaxed-currency serialize ability for middle-tier caching and replication.
682 In *Proc. of ACM SIGMOD*, pages 599–610, 2006.

- 683 [21] P. Guo, J. Wang, B. Li, and S. Lee. A variable threshold-value authenti-
684 cation architecture for wireless mesh networks. *Journal of Internet Tech-*
685 *nology*, 15:929–936, 2014.
- 686 [22] L. Zhou, H. Chao, and A.V. Vasilakos. Joint forensics-scheduling strategy
687 for delay-sensitive multimedia applications over heterogeneous networks.
688 *IEEE Journal on Selected Areas in Communications*, 29(7):1358–1367, Aug
689 2011.
- 690 [23] R. Akbarinia, E. Pacitti, and P. Valduriez. Data currency in replicated
691 dhds. In *Proc.s of SIGMOD*, pages 211–222, 2007.
- 692 [24] L. Savary, G. Gardarin, and K. Zeitouni. Geocach a cache for gml geo-
693 graphical data. *Int. Journal of Data Warehousing & Mining*, 3:66 – 87,
694 2007.
- 695 [25] W. Fang, X. Yin, Y. An, N. Xiong, Q. Guo, and J. Li. Optimal schedul-
696 ing for data transmission between mobile devices and cloud. *Information*
697 *Sciences*, 301:1–344, 2015.
- 698 [26] M. Kaya and R. Alhajj. Development of multidimensional academic infor-
699 mation networks with a novel data cube based modeling method. *Informa-*
700 *tion Sciences*, 265:211–224, 2014.
- 701 [27] G. Bell, J. Gray, and A. Szalay. Petascale computational systems. In *Proc.*
702 *of IEEE INFOCOM*, pages 211–222, 2006.
- 703 [28] Y. Park, D. Seo, J. Yun, C. T. Ryu, J. Kim, and J. Yoo. An efficient
704 data-centric storage method using time parameter for sensor networks. *In-*
705 *formation Sciences*, 180:4806–4817, 2010.
- 706 [29] N. Xiong, X. Jia, L. T. Yang, A.V. Vasilakos, Y. Li, and Y. Pan. A
707 distributed efficient flow control scheme for multirate multicast networks.
708 *IEEE Journal on Selected Areas in Communications*, 27:1254 – 1266, 2010.
- 709 [30] Qingfeng Fang, K. Zeitouni, N. Xiong, Qiongli Wu, Seyit Camtepe, and Yu-
710 Chu Tian. Nash equilibrium based semantic cache in mobile sensor grid
711 database systems. *IEEE Transactions on Systems, Man and Cybernetics:*
712 *Systems*, April 2016.
- 713 [31] N. See-Kee and W. Seah. Game-theoretic model for collaborative protocols
714 in selfish, tariff-free, multihop wireless networks. In *Proc. of INFOCOM*,
715 pages 169–180, 2008.
- 716 [32] D. Niyato and E. Hossain. A game theoretic analysis of service competition
717 and pricing in heterogeneous wireless access networks. *IEEE Transactions*
718 *on Wireless Communications*, 7:150–155, 2008.
- 719 [33] M. N. Omidvar, X. Li, and K. Tang. Designing benchmark problems for
720 large-scale continuous optimization. *Information Sciences*, 316:419–436,
721 2015.