# 3 Vectors Game and Balance Multicast Architecture Algorithms for Sensor Grid

Qingfeng Fan * ||, Qiongli Wu†, Naixue Xiong ‡, Athanasios Vasilakos § and Yanxiang He ¶
* University of Versailles-Saint-Quentin Laboratory PRISM,78035 Versailles Cedex France,qingfeng.fan@prism.uvsq.fr
† Ecole Centrale de Paris and INRIA Saclay Île-de-France,France,qiongli.wu@ecp.fr
‡ Department of Computer Science, Georgia State University,Atlanta,GA 30302-3994,USA,nxiong@cs.gsu.edu
§ Department of Computer and Telecommunication Engineering,University of Western Macedonia,Greece,vasilako@ath.forthnet.gr
¶ School of computer science,Wuhan University,430079,Wuhan,China,yxhe@whu.edu.cn
* qingfeng.fan@prism.uvsq.fr

***Abstract*—We propose a scheme to reach shorter multicast delay, better energy utilizing efficiency and higher efficiency of data transferring for Sensor Grid. Our scheme calculates the space, energy and data weight vectors in one cluster. Then it searches a new vector composed by the linear combination of the three individual ones. We build game balance equation, use the equal correlation coefficient between the new and old vectors to find the point of game balance, seek linear parameters, and generate a least weight path tree. Extended simulation results indicate that our scheme attains less average multicast delay, the number of links used and better system robusticity.**

## I. Introduction

A Sensor Grid integrates wireless sensor networks with grid infrastructures to enable the collection of real-time sensor data. It also enables the sharing of computational and storage resources for sensor data processing management [1].

Many well-known multicast schemes have been presented in reference listed: Double-Channel XY Multicast Wormhole Routing (DCXY) uses an extension of the XY routing algorithm to set up the routing scheme. Dual-Path Multicast Routing (DPM) [2] is developed for the $2-D$ mesh. CAN-based multicast is developed for the multicast applications that use the CAN (Content-Addressable Network) configuration.

In the previous work of multicast for network communication, the system only considered one space factor, but the energy factor and the data factor. In fact, the energy factor can maximize the sensor grid life [3], [4], and the data factor can improve the efficiency of the data transmission [5]. As a result, we must synthetically consider the space, energy and data factors, while constructing the multicast tree, by using the concept that the three factors game balance with each other. So that we can design a multicast scheme in $m-D$ Sensor Grid that can achieve not only shorter multicast delay and less resource consumption, but also the better energy efficency and system robusticity. A set of novel algorithms are presented:
1) Cluster formation algorithm that divides the group members into different clusters in terms of static delay distance;
2) Relative weight vectors generation algorithm that seeks the spatial central node in every cluster, calculates the space weight of every node, searches the weight of energy and data quantity of every node;
3) The least weighted path tree algorithm that, after obtaining the space, energy and data weight vector, builds game balance equations, seeks game balance point, resolves linear parameters, and makes out new weight vector according to the algebra sum of the three known vectors, at last generates the least weighted path tree;
4) Multicast routing algorithm that efficiently dispatches the multicast packets in the group on the basis of the architecture constructed by the above three algorithms.

## II. The Mathematics Model to Describe System

### A. The Mathematics Model of System

The multicast group with $l$ members of the system is denoted as: $G=\{U_0,\ldots,U_i,\ldots,U_{l-1}\}$, where $i\in[0,l-1]$. Each member can be identified by $m$ coordinates: $U_i=(u_{i,0},\ldots,u_{i,j},\ldots,u_{i,m-1})$, when $0\le j\le m-1$. For example, member $U_0$: 2 dimension coordinates $(u_{0,0},u_{0,1})$ as $(0,0)$ and member $U_1$: 2 dimension coordinates $(u_{1,0},u_{1,1})$ as $(0,1)$, etc.

As illustrated in Fig. 2, there are two nodes $U_i=(u_{i,0},\ldots,u_{i,j},\ldots,u_{i,m-1})$, where $i\in[0,l-1]$ and $U_{i'}=(u_{i',0},\ldots,u_{i',j},\ldots,u_{i',m-1})$, where $i'\in[0,l-1]$ and $i'\ne i$.

We defined $U_i$ and $U_{i'}$ are neighbors, if and only if $u_{i,j}=u_{i',j}$ for all $j$, except $u_{i,j'}=u_{i',j'}\pm 1$ along only one dimension $j'$. Thus, in the $m-D$ Sensor Grid, a node may have $m$ to $2m$ neighbors.

We also defined the Manhattan Distance of two nodes [6]. In a $2-D$ Sensor Grid, the static delay distance of two nodes $(X_0,Y_0)$ and $(X_1,Y_1)$ is $|X_1-X_0|+|Y_1-Y_0|$. The sum of static delay distances from all the other nodes $(X_i,Y_i)$ to $(X_0,Y_0)$ $(i\in[1,n-1])$ is: $f(X_0,Y_0)=\sum_{i=1}^{n-1}(|X_i-X_0|+|Y_i-Y_0|)$.

Then we configure the space, energy and data factors. We established three weight vectors to describe them in each cluster, and the value of every item means the relative weight of every node. For example, the space weight vector of the $j-th$ cluster is $W_j^{'}=(w_{j,0}^{'},\ldots,w_{j,i}^{'},\ldots,w_{j,n-1}^{'})$, $i\in[0,n-1]$, $n$ means that there are $n$ nodes in the cluster,

$w'_{j,i}$ means the space weight of the node $i$ within the $j-th$ cluster; In the same way, the energy weight and data vector $W''_j$, $W'''_j$; and the general weight $W_j$. After that, we will discuss how to get the value of the three weight vectors, and how to combine the three vectors to a general one.

### *B. The Space, Energy and Data Weight Vector*

The system has to study special algorithms for computer to understand the space weight. Firstly, the system should find the central node of the cluster, then to figure out the space weight of each node to the central node according to the shortest path principle. Generally speaking, the greater the space weight, the nearer the node to the cluster core, and vice versa. The node with maximum weight is the central node of the cluster namely the space cluster core. For example, the space weight vector of one cluster is shown in Table I. The weights marked * belong to the cluster member. The node with maximum weight is $(2,2)$, for which $W'_{(2,2)} = 10$ and so the node is the cluster space core.

If we establish the multicast tree for one cluster, only considering the space weight, the tree should be as shown in Fig. 1. And each factor would maximize its own interests.

TABLE I
THE SPACE WEIGHT VECTOR $W'$ IN ONE CLUSTER, THE WEIGHTS MARKED * BELONG TO THE CLUSTER MEMBER.

| | | | | | |
|---|---|---|---|---|---|
| Y=6 | 0 | 1* | 0 | 0 | 0 |
| Y=5 | 0 | 3 | 2* | 1 | 1* |
| Y=4 | 0 | 4* | 2 | 1 | 1 |
| Y=3 | 1* | 5 | 2 | 1 | 1 |
| Y=2 | 2 | 10* | 4 | 2 | 2* |
| Y=1 | 1* | 3* | 1* | 0 | 0 |
| | X=1 | X=2 | X=3 | X=4 | X=5 |

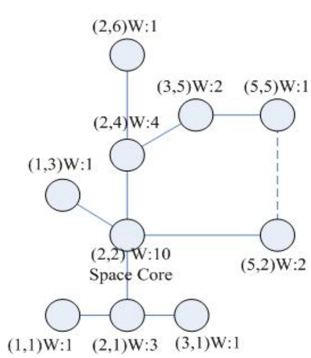



Fig. 1. The multicast tree according to the space weight.

Compared with the space weight vector, the energy weight vector is easier to be obtained, as shown in Table II.

We also can just define the data weight in the same way, as shown in Table III.

## III. THE RELATIONSHIP OF THREE FACTORS

The relationship of the space, energy and data weight vector $W'$, $W''$ and $W'''$ are game balance. The space, energy and data factors are three factors independent with each other,

TABLE II
THE ENERGY WEIGHT VECTOR $W''$, IN THE CLUSTER, THE WEIGHTS MARKED * BELONG TO THE CLUSTER MEMBER.

| | | | | | |
|---|---|---|---|---|---|
| Y=6 | 0 | 4* | 0 | 0 | 0 |
| Y=5 | 0 | 3 | 2* | 1 | 1* |
| Y=4 | 0 | 10* | 2 | 2 | 1 |
| Y=3 | 7* | 4 | 3 | 2 | 1 |
| Y=2 | 2 | 1* | 4 | 3 | 8* |
| Y=1 | 3* | 5* | 6* | 0 | 0 |
| | X=1 | X=2 | X=3 | X=4 | X=5 |

TABLE III
THE DATA WEIGHT VECTOR $W'''$ IN ONE CLUSTER, THE WEIGHTS MARKED * BELONG TO THE CLUSTER MEMBER.

| | | | | | |
|---|---|---|---|---|---|
| Y=6 | 0 | 1* | 0 | 0 | 0 |
| Y=5 | 0 | 3 | 2* | 1 | 0* |
| Y=4 | 0 | 5* | 2 | 1 | 1 |
| Y=3 | 2* | 5 | 2 | 1 | 1 |
| Y=2 | 2 | 1* | 4 | 2 | 2* |
| Y=1 | 3* | 10* | 3* | 0 | 0 |
| | X=1 | X=2 | X=3 | X=4 | X=5 |

which have meaning and formation respectively; any of them tends to maximize their result. Namely the three factors game with each other. On the other hand, the three factors also coexist in a system, common working, mutual interaction and constraint. Namely they balance with each other. We must synthetically consider the space, energy and data factors while constructing the multicast tree. The basic idea goes through the whole process of constructing the hierarchical multicast tree.

After generating the space, energy and data weight vectors $W'$, $W''$, $W'''$, we combine the three old ones to a new general weight vector $W$. Now the system just knows $W = f(w', w'', w''')$, but it does not know the expression of the $f()$. There are a lot of formats of f() can be used, but for one simple and effective trial, we just used the linear form: $W = \alpha W' + \beta W'' + \gamma W'''$. After that we build game balance equations, seek game balance point, resolve linear parameters $\alpha$, $\beta$, $\gamma$ generate new weight vector $W$. At last, we generate the least weighted path tree as hierarchical multicast tree in one cluster.

**(1). To define the weights of the nodes:**

$$W_{i,j} = \alpha_i W'_{i,j} + \beta_i W''_{i,j} + \gamma_i W'''_{i,j} \quad (1)$$

$W_{i,j}$: The weights of the nodes;
$\alpha_i$, $\beta_i$, $\gamma_i$: Linear relation modulus, $\alpha_i, \beta_i, \gamma_i \in r$, $\alpha_i, \beta_i, \gamma_i \geq 0$,as $\alpha_i, \beta_i, \gamma_i < 0$ nonsense;
$W'_i$: The space weight vector;
$W''_i$: The energy weight vector;
$W'''_i$: The data weight vector.

**(2). The linear relation modulus of the weights of the nodes satisfied:**

$$\alpha_i + \beta_i + \gamma_i = 1; 0 < \alpha_i, \beta_i, \gamma_i < 1; \alpha_i, \beta_i, \gamma_i \in r. \quad (2)$$

**Theorem 1.** *If, three linear no-relationship vectors $W'_i$, $W''_i$, $W'''_i$ their linear combination $W_{i,j} = \alpha_i W'_{i,j} + \beta_i W''_{i,j} +$*

$\gamma_i W_{i,j}^{'''}$, $\alpha_i, \beta_i, \gamma_i$ *are Linear relation modulus,* $\alpha_i, \beta_i, \gamma_i \in r$, $\alpha_i, \beta_i, \gamma_i \geq 0$, *then following express is satisfied:*

$$\alpha_i + \beta_i + \gamma_i = 1,\ 0 < \alpha_i, \beta_i, \gamma_i < 1,\ \alpha_i, \beta_i, \gamma_i \in r$$

**(3). The space, energy and data factors are game balance with each other, the game balance point is:**

$$\frac{W_i \cdot W_i'}{\|W_i'\|} = \frac{W_i \cdot W_i''}{\|W_i''\|} = \frac{W_i \cdot W_i'''}{\|W_i'''\|} \tag{3}$$

**Theorem 2.** *If three linear no-relationship vectors,* $W'_i = (w'_{i,0}, \ldots, w'_{i,j}, \ldots, w'_{i,m-1})$, $W''_i = (w''_{i,0}, \ldots, w''_{i,j}, \ldots, w''_{i,m-1})$, $W'''_i = (w'''_{i,0}, \ldots, w'''_{i,j}, \ldots, w'''_{i,m-1})$ *their linear combination* $W_i = (w_{i,0}, \ldots, w_{i,j}, \ldots, w_{i,m-1})$, *and* $W_i = \alpha_i W_i^{'} + \beta_i W_i^{''} + \gamma_i W_i^{'''}$, $\alpha_i, \beta_i, \gamma_i$ *are Linear relation modulus,* $\alpha_i, \beta_i, \gamma_i \in r$, $\alpha_i, \beta_i, \gamma_i \geq 0$. *The game balance point of* $W'_i$, $W''_i$*and* $W'''_i$ *is* $\frac{W_i \cdot W_i'}{\|W_i'\|} = \frac{W_i \cdot W_i''}{\|W_i''\|} = \frac{W_i \cdot W_i'''}{\|W_i'''\|}$.

Combining (4), (5), the paper builds the liner binary simple equations:

$$\begin{cases} \alpha_i + \beta_i + \gamma_i = 1 \\ \frac{W_i \cdot W_i'}{\|W_i'\|} = \frac{W_i \cdot W_i''}{\|W_i''\|} = \frac{W_i \cdot W_i'''}{\|W_i'''\|} \end{cases}$$

For $W_i = \alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i'''$ $\frac{(\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i'}{\|W_i'\|} = \frac{(\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i''}{\|W_i''\|}$ $= \frac{(\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i'''}{\|W_i'''\|}$ then

$$(\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i' \cdot \|W_i''\| \cdot \|W_i'''\|$$
$$= (\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i'' \cdot \|W_i'\| \cdot \|W_i'''\|$$
$$= (\alpha_i W_i' + \beta_i W_i'' + \gamma_i W_i''') \cdot W_i''' \cdot \|W_i'\| \cdot \|W_i''\|$$

For $\alpha_i + \beta_i + \gamma_i = 1$,then

$$\alpha_i = \frac{\|W_i''\| + \|W_i'''\|}{2(\|W_i'\| + \|W_i''\| + \|W_i'''\|)} = \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{''\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'''\,2}}}{2\left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{''\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'''\,2}}\right)}$$

$$\beta_i = \frac{\|W_i'\| + \|W_i'''\|}{2(\|W_i'\| + \|W_i''\| + \|W_i'''\|)} = \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'''\,2}}}{2\left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{''\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'''\,2}}\right)}$$

$$\gamma_i = \frac{\|W_i'\| + \|W_i''\|}{2(\|W_i'\| + \|W_i''\| + \|W_i'''\|)} = \frac{\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{''\,2}}}{2\left(\sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{''\,2}} + \sqrt{\sum_{j=0}^{m-1} w_{i,j}^{'''\,2}}\right)}$$

So: $0 < \alpha_i, \beta_i, \gamma_i < 1, \alpha_i, \beta_i, \gamma_i \in r$

According to the above data table, the algorithm figures out $\alpha_i = 0.385$, $\beta_i = 0.249$, $\gamma_i = 0.369$.

**(4). To get the weight vector and choose the maximum value node as the cluster core**

According to the above all, the algorithm gets the weight vector (as Table IV) and chooses the maximum value node as the cluster core $C^* = (c_0^*, \ldots, c_i^*, \ldots, c_{m-1}^*)$, in this cluster it chooses $(2, 1)$ as cluster core $c_i$.

TABLE IV
THE WEIGHT VECTOR $W$ , IN THE CLUSTER, THE WEIGHTS MARKED * BELONG TO THE CLUSTER MEMBER

| | | | | | |
|---|---|---|---|---|---|
| Y=6 | 0 | 1.74* | 0 | 0 | 0 |
| Y=5 | 0 | 3 | 2.00* | 1 | 0.63* |
| Y=4 | 0 | 5.84* | 2 | 2 | 1 |
| Y=3 | 2.84* | 4 | 3 | 2 | 1 |
| Y=2 | 2 | 4.47* | 4 | 3 | 3.48* |
| Y=1<br>hline | 2.23*<br>X=1 | 6.08*<br>X=2 | 2.97*<br>X=3 | 0<br>X=4 | 0<br>X=5 |

## IV. ALGORITHMS FOR GAME BALANCE MULTICAST ARCHITECTURE

### A. Cluster Formation Algorithm

In our algorithms, the group members are initially split into several clusters by some management nodes (called as Rendezvous Points - RP). The cluster size is normally set as:

$$S = (k, 3k - 1) \tag{4}$$

The expression $(k, 3k-1)$ represents a random constant between $k$ and $3k-1$. Like NICE, uses a fixed value k. k is a constant using $k = 3$ [**?**]. The definition of cluster size is to avoid the frequent cluster splitting and merging [7]. For the cluster formation algorithm, the RP initially selects the left lowest end host (say U) among all unassigned members,as shown in Fig. 2.

### B. Relative Weight Vectors Generation Algorithm

This sub-algorithm generates the space, energy and data weight vectors $W^{'}$, $W^{''}$ and $W^{'''}$; in addition, the space, energy and data cores $c_{i,a}$, $c_{i,b}$ and $c_{i,c}$. Hence it can be divided into six steps:

**(1). To find the space center nodes as the space core** $C_{i,a}$ The following theorem provides the sufficient and necessary conditions to select a spatial core in each cluster.

**Theorem 3.** *Let* $U$ *be the cluster member that occupies the node* $(u_0, \ldots, u_j, \ldots, u_{m-1})$ *in a* $m-D$ *Sensor Grid and* $n>j$, $n<j$ *and* $n=j$ *be the number of cluster members with the* $j-th$ *coordinates larger than (right nodes of* $j-th$ *row ), less than (left nodes of* $j-th$ *row), and equal to* $u_j$ *(the nodes just on* $j-th$ *row) respectively. Then* $U$ *is the spatial center node if and only if the following inequalities hold simultaneously:*

$$|n_{<j} - n_{>j}| \leq n_{=j}, j = 0, 1, \ldots, m-1 \tag{5}$$

**Proof:** We have proposed the proof of Theorem 3 in [**?**].

The physical meaning of the theory is obvious. Firstly, we process on $X$ axis. For example $N_{=2} = 4$, namely there are 4 nodes just on of second row:$(2,6)$, $(2,4)$, $(2,2)$, $(2,1)$; $N_{<2} = 2$, namely there are 2 nodes in the left of second row:$(1,3)$, $(1,1)$; $N_{>2} = 4$, namely there are 4 nodes in the right of second row$(3,5)$, $(3,1)$, $(5,5)$, $(5,2)$, so $|n_{<2} - n_{>2}| \leq n_{=2}$. So that $N_{=2}$ is satisfied coordinates on $X$ axis. On other hand, $N_{=3} = 2$, including$(3,5)$, $(3,1)$; $N_{<3} = 6$, including $(2,6)$,

$(2,4)$, $(2,2)$, $(2,2)$, $(1,3)$, $(1,1)$; $N_{>3} = 2$, including $(5,5)$, $(5,1)$, so $|n_{<3} - n_{>3}| \geq n_{=3}$. So that $N = 3$ is not satisfied coordinates.

In the same way, we can do it again on $Y$ axis. Then we can find the $(2,2)$ is the space central node, namely the space core of the cluster.

**(2). To calculate the value of the space weight vector $W^{'}_{i,j}$**
For only considering the space factor, the system establishes a multicast tree to transfer data packets, which choose the space core as the root and organize the architecture according to the space weight vector. The tree should maximize the sharing of links utilization within the clusters, so that the rest of the links may be used for other traffic. Our approach is to connect all the members, according to (1) the branch on the tree between two adjacent members is the shortest path in the cluster, (2) the total number of links on the tree should also be minimized. Before discussing the algorithm, it is necessary to define the following terminologies (using a $2-D$ cluster as the model):

**(I) Shortest path area nodes (SPAN):** For any two nodes $(x_0, y_0)$ and $(x_1, y_1)$, let $X_{min} = min\{x_0, x_1\}$, $X_{max} = max\{x_0, x_1\}$, $Y_{min} = min\{y_0, y_1\}$ and $Y_{max} = max\{y_0, y_1\}$. They uniquely define a rectangle area $[x_0, y_0] \times [x_1, y_1]$. Each node $(x, y)$ in $[x_0, y_0] \times [x_1, y_1]$, which is on one of the shortest paths between $(x_0, y_0)$ and $(x_1, y_1)$, so it is called the shortest path area nodes (SPAN) between $(x_0, y_0)$ and $(x_1, y_1)$.

**(II) SPAN nodes of a cluster member:** When the tree is built in the cluster with the size of $n$, all nodes $C_j(x_j, y_j)$ in the SPAN area $[x_0, y_0] \times [x_i, y_i]$ from the core (i.e. the root of the tree)$c^*(x^*, y^*)$ to a cluster member $c_i(x_i, y_i)(i \in [0, n-1])$can be regarded as the SPAN nodes of $c_i$. Take Fig. 2 as an example. Assume that the core is in the node $(2,2)$. All nodes in $[2,2] \times [5,5]$ are the SPAN nodes of this cluster member.

**(III) The space weight of the node:** A node may be the SPAN node of several $k$ cluster members. If a node is the SPAN node of $k$ cluster members, this node is assigned the weight of $k$. Table I gives the space weights of all nodes, and takes the node $(2,4)$ as an example, as shown in Fig. 2. The node $(2,4)$ is $4$ node's Shortest Path Area Nodes (SPAN):$(2,6)$, $(3,5)$, $(5,5)$, $(2,4)$, because it is in the Shortest Path Area of these nodes. Therefore its weight $4$ means that $4$ cluster members may pass through node $(2,4)$ to the cluster core $(2,2)$ by the shortest paths. Apparently, the weight of $(2,2)$ is $10$.

In general, if the space weight of the node is $k$, it means that there are $k$ nodes which must pass this node to the space core to send packets, which represent the degree near the center. The greater the space weight is, the nearer the node to the cluster core is, and vice versa.

**(3). To find the value of energy weight $W^{''}$ and the energy core $c_{i,b}$**

**(4). To find the value of the data weight $W^{'''}$ and the data core $c_{i,c}$**

---

**Algorithm 1:** Relative Weighted Vectors Generation

---

**Input**: Cluster Member:
$C = \{C_0 = (C_{0,0}, C_{0,1}, \ldots, C_{0,m-1}), \ldots,$
$C_i = (C_{i,0}, C_{i,1}, \ldots, C_{i,m-1}), \ldots,$
$C_{n'-1} = (C_{n'-1,0}, C_{n'-1,1}, \ldots, C_{n'-1,m-1})\}$,
where $i \in [0, n^{'} - 1]$;
**Output**: The space weight vector:
$W' = \{W'_0 = (w'_{0,0}, w'_{0,1}, \ldots, w'_{0,m-1}), \ldots,$
$W'_i = (w'_{i,0}, w'_{i,1}, \ldots, w'_{i,m-1}), \ldots,$
$W'_{n'-1} = (w_{n'-1,0}, w'_{n'-1,1}, \ldots, w'_{n'-1,m-1})\}$,
where $i \in [0, n^{'} - 1]$;
The energy weight vector:
$W'' = \{W''_0, \ldots, W''_i, \ldots, W''_{n'-1}\}$,
where $i \in [0, n^{'} - 1]$;
The data weight vector:
$W''' = \{W'''_0, \ldots, W'''_i, \ldots, W'''_{n'-1}\}$,
where $i \in [0, n^{'} - 1]$;
and the space core $C^*_a = \{c^*_{0,a}, \ldots, c^*_{m-1,a}\}$,
the energy core $C^*_b = \{c^*_{0,b}, \ldots, c^*_{m-1,b}\}$,
the data core $C^*_c = \{c^*_{0,c}, \ldots, c^*_{m-1,c}\}$.

*(1).To find the spatial center nodes as the spatial core $c_{i,a}$ in every cluster $c_i$;*
**begin**
  Initiate $\{a_{\{c_j\}min}, \ldots, a_{\{c_j\}t}, \ldots, a_{\{c_j\}max}\}$ =$\{0, \ldots, 0, \ldots, 0\}$;
  *// $a_{\{c_j\}^t}$ records the number of cluster members whose j-th coordinates equal to $(C_j)_t$, where whose j-th coordinates equal to $(C_j)_t$, where $(C_j)_{min} \leq (C_j)_t \leq (C_j)_{max}$ and $0 \leq j \leq m-1$.*
  **for** $k = 0$ *to* $n^{'} - 1$ **do**
    **if** *the j-th coordinate of* $C_k == (C_j)_{td}$ **then**
      $a_{(c_j)t} = a_{(c_j)t} + 1$;
    **end**
  **end**
  **for** $i = 0$ *to* $n^{'} - 1$ **do**
    **for** $j = 0$ *to* $m - 1$ **do**
      **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**
        $C^*_j = C^*_{i,j}$;
        $j = j + 1$;
      **else**
        $j = m - 1; i = i + 1$;
      **end**
    **end**
  **end**
  $C^*_a = \{c^*_{0,a}, \ldots, c^*_{m-1,a}\}$;
**end**

---

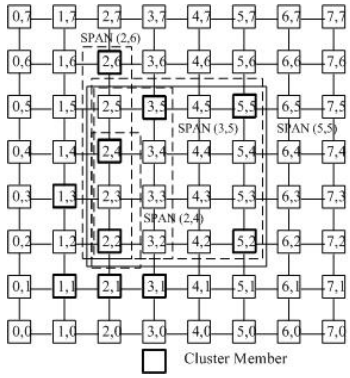



Fig. 2. Selecting the spatial center nodes in the members of one cluster of a $2-D$ Sensor Grid. And the shortest path area nodes (SPAN) in a $2-D$ Sensor Grid, for example: The node (2; 4) is 4 node's Shortest Path Area Nodes(SPAN):(2; 6),(3; 5),(5; 5),(2; 4).

---

*(2).To calculate the space weight vector $W_j^{''}$;*
**begin**
  T={};
  For any node $C_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,m-1})$ with $(C_j)_{min} \leq (C_j)_t \leq (C_j)_{max}$, initialize its weight $W''_{c,j} = 0$;
  **for** $j = 0$ *to* $n^{'} - 1$ **do**
    **for** $i = 0$ *to* $n^{'} - 1$ **do**
      **if** $C_i$ *is a SPAN node of* $C_j = (C_{j,0}, C_{j,1}, \ldots, C_{j,m-1})$ **then**
        $W^{'}_{c,j} = W^{'}_{c,j} + 1$;
      **end**
    **end**
  **end**
  $W' = \{W'_0 = (w'_{0,0}, w'_{0,1}, \ldots, w'_{0,m-1}), \ldots, W'_i = (w'_{i,0}, w'_{i,1}, \ldots, w'_{i,m-1}), \ldots, W'_{n'-1} = (w_{n'-1,0}, w'_{n'-1,1}, \ldots, w'_{n'-1,m-1})\}$, where $i \in [0, n^{'} - 1]$;
**end**
*(3).To find the energy weight $W^{'}_{i,j}$ and the energy core* $c_{i,b}$in the cluster $C_j$;
(4).To find the data quantity weight $W^{'''}_{i,j}$ and the data core $c_{i,c}$ in the cluster $C_j$;

---

### *C. Least Weighted Path Tree Generation Algorithm*

After the Relative Weighted Vectors Generation Algorithm generates the space, energy and data weight vectors $W^{'}$, $W^{''}$, $W^{'''}$, and the space, energy, data cores $c_{i,a}$, $c_{i,b}$, $c_{i,c}$, the Least Weighted Path Tree Generation Algorithm wants to combine the three old weight vectors $W'$, $W''$ and $W'''$ to a new weight vector $W = f(W^{'}, W^{''}, W^{'''})$. As we mentioned in section III, we used the linear form: $W = \alpha W^{'} + \beta W^{''} + \gamma W^{'''}$. After that the sub-algorithm builds binary simple equations, resolves linear parameters, $\alpha$, $\beta$, $\gamma$ generates new weight vector $W$. At last generates the least weighted path tree as hierarchical multicast tree.

### *D. Multicast Routing Algorithm*

Multicast Routing Algorithm efficiently dispatches the multicast packets in the group on the basis of the architecture constructed by the above three algorithms.

## V. Performance Evaluation

### *A. The Model of Simulation*

We evaluated 3 Vectors Game Balance Multicast Algorithms with the simulation developed by C++ [8] and run by a group of 40 IBM double cores PCs. We chose four multicast routing approaches for $2-D$ Sensor Grid used for the performance testing and comparison: SPACE, ENERGY, DATA and GBMASG which synthetically considers space, energy and data factors. Moreover we chose DCXYP as our SPACE approach, which is the most popular multicast technology, among exist approaches. Here we use ENERGY and DATA approach according to energy and data weight vector to generate least weight path tree.

In the simulation environment, the network topology used in the simulation is a $2-D$ Sensor Grid. The bandwidth of each link is 10Mbps. During the simulation, 1000 and 1000,000 multicast packets are randomly generated as time seed and the average size of the packets is 2400 bytes so that the average time to transmit a packet on the defined link is about 1ms. The following three metrics are employed to evaluate these multicast schemes:

**Average multicast delay:** is computed by

$$AD = (\sum_{i=0}^{n-1} d(s, u_i))/n \tag{6}$$

where $d(s, u_i)$ is the packet delay from the source $s$ to the member $u_i$ and $n$ is the group size.

**Number of links used:** The total number of links used.

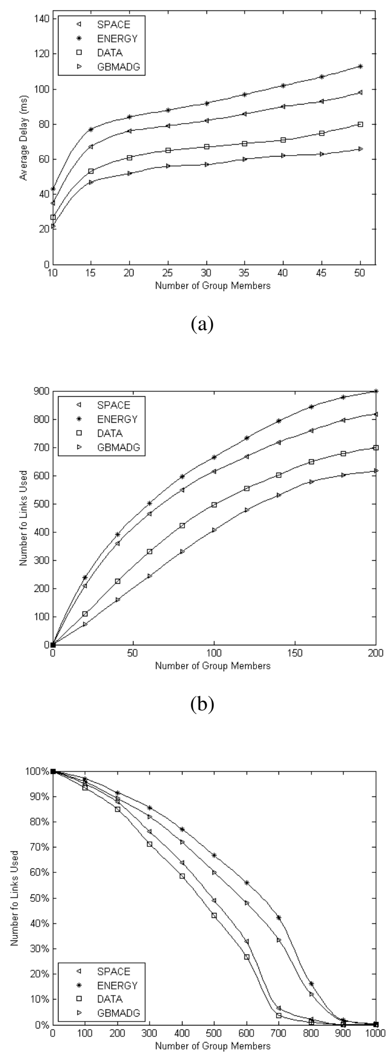
**Packets Arrival Rate:** The rate of arrival data packets.

### *B. The Result of Simulation*

The average delay metric under the heavy load of network is shown in Fig. 3. The link usage for different algorithms under the light and heavy load of network is shown in Fig. 3. The packets arrival-rate under the heavy load of network is shown in Fig. 3. From these simulation results, it can be obtained the following observations:

1. Under a heavy load circumstance, the delay is mainly decided by the source of the data quantity, and certainly relates to the space and energy of the nodes too(Fig. 3 (a). In the Sensor Grid circumstance, a majority of data quantity will concentrate in minor nodes. Now that SPACE just generate the multicast tree according to space factor, so that the delay increases a little rapidly in the mass data quantity. Obviously energy poorer than SPACE. Our approach GBMASG synthetically considers space, energy and data factors, so that it gets the best result. DATA achieves the quite well delay performance almost as GBMASG here.

2. Under a heavy load circumstance, the number of links is mainly decided by both the source of the data quantity and the




(a)

(b)

(c)

Fig. 3. Simulation results for SPACE, ENERGY, DATA and our GBMASG: (a) The Delay under the Different No. of the Group Members; (b) Links Used under the Different No. of Group Members; (c) Packets Arrival Rate under the Different No. of Group Members.

space of the node (Fig. 3 (b). So that for SPACE the number of links increases rapidly in the mass data quantity, and the DATA is much better, at last our approach GBMASG get the least the number of links. ENERGY is not very good also.

3. Fig. 3 (c) show the packets arrival-rate used by these approaches. In general, the packet arrival-rate will be decreased with the time and is mainly decided by ENERGY. In heavy load circumstance decreased much more quickly. ENERGY better than SPACE and DATA, at last our approach GBMASG get the best the packets arrival-rate, as good as ENERGY.

It reveals that under the same condition, GBMASG obtains the best balance over the performance parameters i.e., the less resource a system consumes, the higher the throughput and the shorter the delay under the weight traffic load and the higher system robusticity. The GBMASG is especially suitable for a great deal data quantity and long duration of Sensor Grid.

## VI. Conclusions and Future Work

### A. Conclusions

In the Sensor Grid, when the system constructs the hierarchical tree, it should synthetically consider the factors of the space, energy and data, whose relationship is game balance. We tried to draw an elaborate balance between them, and uses the basic idea to construct the hierarchical multicast tree in this paper.

### B. Future Work

**(1). To extend to N-vectors correlation.**

After discussing two and three vectors correlation, the algorithm can be extended to N-vectors correlation.

The relationship of the weight can be defined as $W_{i,j} = \alpha_i^{(1)} W_{i,j}^{(1)} + \ldots + \alpha_i^{(k)} W_{i,j}^{(k)} + \ldots + \alpha_i^{(n)} W_{i,j}^{(n)}$.

And the equation can be extended to

$$\begin{cases} \alpha_i^{(1)} + \ldots + \alpha_i^{(k)} + \ldots + \alpha_i^{(n)} = 1 \\ \frac{W_i \cdot W_i^{(1)}}{\left\| W_i^{(1)} \right\|} \ldots = \frac{W_i \cdot W_i^{(k)}}{\left\| W_i^{(k)} \right\|} \ldots = \frac{W_i \cdot W_i^{(n)}}{\left\| W_i^{(n)} \right\|} \end{cases}$$

It can be solved by mathematical induction.

**(2). To extend from linear non-relationship condition to linear relationship condition**

Moreover, as the real world meaning of different factors are independent, these factors are linear non-related, so their cardinal number is accountable infinite. If some factors are linear related, we can use GramCSchmidt process Orthonormalization and fussy logic to turn these to be linear non-related. Then their cardinal number is unaccountable infinite.

## Acknowledgements

The authors acknowledge Professor Weijia Jia City University of Hong Kong and Professor Patrick Valduriez Universit de Nantes en France. The authors acknowledge partial financial support form the SAP Business Objects company.

## References

[1] R. Duff and Y. E. Khamra, “A sensor and computation grid enabled engineering model for drilling vibration research,” *MG ’08: Proceedings of the 15th ACM Mardi Gras conference*, 2008.

[2] S. Chen and B. Shi, “Acom: Any-source capacity-constrained overlay multicast in non-dht p2p networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 9, pp. 1188–1201, 2007.

[3] U. O. Candogan, I. Menache, A. Ozdaglar, and P. A. Parrilo, “Near-optimal power control in wireless networks: A potential game approach,” *in Proceedings of IEEE INFOCOM*, 2010.

[4] H. Inaltekin and S. B. Wicker, “The analysis of nash equilibria of the one-shot random-access game for wireless networks and the behavior of selfish nodes,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1094–1107, 2008.

[5] M. Iqbal and H. B. Lim, “A sensor grid infrastructure for large-scale ambient intelligence,” *in Proceedings of the 2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2008.

[6] R. Akbarinia, E. Pacitti, and P. Valduriez, “Data currency in replicated dhts,” *in Proceedings of SIGMOD*, 2007.

[7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast,” *in Proceedings of ACM SIGCOMM*, 2002.

[8] W. Tu and W. Jia, “A scalable and efficient end host multicast for peer-to-peer systems,” *in Proceedings of IEEE Globecom*, 2004.