

Research on Query Gain Routing Algorithms and Load Balancing Mechanism in Wireless Sensor Networks

Jiang Hui-Lin, Qiao Li and Fu Yi-De

*School of Computer and Information Technology, Shangqiu Normal College,
Shangqiu, China*

Fu yi-de

*School of Computer Science and Engineering, Nanjiang University of Science and
Technology, Nanjiang, China*

E-mail: hljiangsl@yeah.net

Abstract

It is necessary to reduce the consumption of network node energy and the network load for solving the query design for routing algorithm in wireless sensor network, thus this article presents query gain routing algorithms and load balancing mechanism. This algorithm selects routing node by querying the routing information of gain matrix according to the successful record of historical query. The load balancing mechanism queries the energy information, which records node in the routing process and transfers load to balance the energy consumption of each node in the query path. This algorithm is able to effectively aware routing load to reduce the network energy consumption and avoid the data collision. The experimental result shows that compared with other query routing strategy, the query gain routing can increase the success rate of query under the condition of reducing the node energy consumption, but the load balancing mechanism can reduce the energy consumption of query gain routing further and prolong the network life cycle more effectively.

Keywords: *Routing Algorithms, Load, Gain, Energy*

1. Introduction

Wireless sensor network is a kind of wireless communication network which has been constructed by a large of arranged sensor nodes with data-aware, data storage and communication capabilities over a surveillance area in Ad hoc mode, then monitor the object of the physical world up close in real-time by various kinds of sensor. As the perception nerve endings of the internet of things, wireless sensor network is applied widely in the many fields: national defense military, environmental monitoring, agricultural production, health care, anti-terrorism and disaster relief etc [1-4]. Since sensor produce a variety of observational data continuously, wireless sensor networks can be regarded as a special database, user can issue a query to obtain the required information through the wireless sensor network. Although the application scenarios of wireless sensor network and hardware deployment are not the same, their ultimate goal is for obtaining, processing and transmitting the perceptive data. Therefore, the wireless sensor network is a data-centric network which is also has the characteristics like numerous nodes, short communication distance, limited energy *etc.*, [5-7]. For the query routing algorithms of wireless sensor network design, it must to be tried to reduce the energy consumption of network node. Additionally, since some of the key nodes may use up the energy prematurely due to the heavy load, it will cause the network to be divided into several isolated regions. Therefore, it is necessary to combine the energy information

from nodes during the process of the query routing, so that it can balance the energy consumption, thereby extending the network life cycle [8].

In the Rumor routing algorithm, which BRAGINSKY D presents, source node transmits the data randomly in the network, then keeps a copy in the intermediate node which the data pass by and user query also transmits randomly in the network. Once the transmissions path of both intersects, the user query will get the result. The simplest strategy to implement Rumor routing algorithm is: the source node transmits data along the horizontal direction and keeps a copy in the passing path, user query transmits along the vertical direction, and thus it ensures the path of both can intersect. Rumor routing algorithm is better than flooding algorithm in query efficiency and data transmission rate indicators [9]. Liu Xin proposed a kind of data storage and query routing algorithms that based on comb - pin model [10]. The algorithm is an expansion of Rumor routing algorithm; this idea is similar to using a comb to find the needle in the haystack [11]. Data (like needle) transmits a certain distance along the vertical direction and keeps a copy in the passing path; user query (like comb) transmits along the vertical direction; then transmits along the horizontal direction with multiple paths at regular distance in the same time, this distance can ensure that the transmission path of query requirement intersects data storage path. This process can also be reversed, that is the query requirement transmits a certain distance in a vertical direction, user query transmits along the horizontal direction with multiple paths, it depends on the relation between the rate of data generation and query occurrence [13-15]. Huhua proposed a kind of routing algorithm based on tree clustering, through clustering sensor nodes of the sensor network to form a super node, reducing access to the sensor nodes and improve query efficiency [16]. XIN Li proposed a kind of distributed indexing algorithm to handle the multi-dimension range query in the sensor networks [17-19].

2. Query Routing Algorithm

A. Network Model

Assume all the nodes in wireless sensor networks randomly distributed in a two-dimensional plane, and each node has the same transmission radius, you can use the unit disk graph (UDG) to model network topology, and all connections are bidirectional.

Define use an undirected graph with $G(V, E)$ to represent a wireless sensor network, $V \in V \times V$ represents the sensor node set of Euclidean plane; $E \in N \times N$ represents the set of corresponding nodes edges. Assume that all nodes have the same communication range r , the distance between any two nodes of s_u and s_v is given from the function of (u, v) :

$$\delta(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2} \quad (1)$$

If and only if the distance $\delta(u, v)$ between any two nodes of s_u and s_v is less than r , s_u and s_v become neighboring nodes, and s_u and s_v exists a link between each other.

The energy consumption of wireless sensor networks use the formula LEACH model gave, it consist of packet to send, receive, and event perception these three parts, and make use the formula LEACH model gave can calculate the residual energy of nodes.

Consider s_u and s_v on the two minimum hop distance h , there are thresholds of $u(h)$ and $v(h)$, to meet $u(h) \leq \delta(u, v) \leq v(h)$ $u(h)$. Among them, the threshold size depends on node density p of wireless sensor network, and for any $h > 0$ exists:

$$\lim_{p \rightarrow \infty} (v(h) - u(h)) = r \quad (2)$$

Definition 2 given source node s_v and the destination node s_u , the sensor model is defined as

$$r(u, v) = \frac{\lambda}{\delta(u, v)^i + \varepsilon} \quad (3)$$

In the above formula, λ represents the wireless signal amplitude $\lambda > 0$; k is the sensor parameters; ε is a predefined parameters used to process the same situation for the source node and the destination node position, $\varepsilon > 0$.

Sensor node sends a data packet use the following *six-tuple* to represent: (SenderID, SeqNo, state, hops, DestID, data). Sender ID is a identifier to create the sensor nodes of data package; SeqNo represent the serial number of the data package, it was assigned by the sensor node the packet created; state presents the current state of the data packet, including the initial state, the forwarding state and end state; hops used to define the data packet's life cycle, when the message is passed from one node to another node, the amount of hops plus one, in order to control the times of packets transmitted, and avoid too much bandwidth consumption, it need to be set an upper limit value of the data package life cycle; DestID represents identifier of destination node; data field contains the data information that packet stored.

B. Gain Query Routing Algorithm

Query gain matrix (query gain matrix, QGM) is the key components to maintain routing information in the query routing algorithm. Before constructing QGM, define two auxiliary data structures Query Table and Sensor Table. Query Table of one node contains a set sends or forwards from this node and eventually query the hit query vector set. While another node Sensor Table records the node forward one query to its neighbors and finally query neighbor set of the hits. Therefore, the Sensor Table of some node is subset of all neighboring node collection. When the query is hit, all nodes (not including the destination node) from the source node to the destination node will record the query in the Query Table of this node, and recorded the neighboring nodes of this query hit path in the node Sensor Table.

Give an object set O containing k elements, a query vector set Q containing n elements and a node set S containing m element, for any one object $o_a \in O (1 \leq a \leq k)$, any one query $q_j \in Q (1 \leq j \leq n)$ and any node $s_i \in S (1 \leq i \leq m)$, then the query result object collection is $o_{qj} \in O (1 \leq j \leq n)$. QGM is a matrix $(m \times n)$ shown as below:

$$\left\{ \begin{array}{c} s_1 \\ \vdots \\ s_i \\ \vdots \end{array} \right\} \left\{ \begin{array}{c|ccc|c} gain_{10} & gain_{11} & \cdots & gain_{1j} & gain_{1n} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline gain_{i1} & gain_{i2} & \cdots & gain_{ij} & gain_{in} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline gain_{n1} & gain_{n2} & \cdots & gain_{nj} & gain_{nk} \end{array} \right\} \quad (4)$$

Each of matrixes is composed of a complex data structure. The number of rows for QGM on a node should be equal to the length of the corresponding Sensor Table; the number of columns is equal to the length of the corresponding Query Table.

Each node maintains a neighboring node list and a query gain matrix. The neighboring node list can update and maintain by the methods of the broadcast and monitoring query request. Any node can create a query and then query messages through intermediate nodes routing is forwarded to the destination node.

In the initial stages of query processing, the Query Table, Sensor Table on each node and the corresponding QGM are empty.

Figure 1 depicts exponential decaying transmission of the gain value in the query gain routing algorithm.

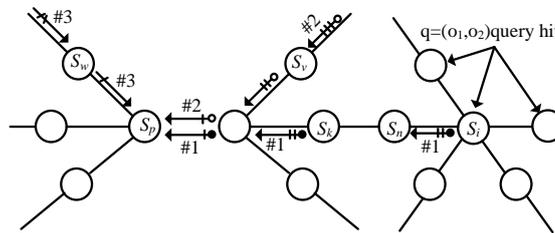


Figure 1. The Exponential Decaying Transmission of Gain Value

Each circle in Figure 1 represents a sensor node, circles filled with gray represent query hit nodes, and each arrow represents some query hits path that gain value transmits. In Figure 1, there are altogether three hits query paths are labeled #1, #2 and #3; assuming a node s_p issues query q_j contains an object, and transmits the query to the neighboring node s_i . Finally, query is hit on the target node s_t . If the present hops value of query message is less than TTL, then it will forwards query continually to the neighbor nodes of s_i , for searching more nodes which satisfy the query. Assuming q_j is the j -th element in the Query Table of s_p , s_i is the i -th element in the Sensor Table of s_p then $add \prec \{s_i, s_k, \dots, s_u, s_t\}, value \succ$ to $gain_{ij}$, the computing method of $gain_{ij}$ is:

$$|gain_{ij}| = \sum_{s_i \in D} \frac{(1-\alpha)\beta_{s_t} + n_a}{dist(s_p, s_i)} \quad (5)$$

Where: D is a collection of all nodes which passed s_i and the query hit; β_{s_t} is the total amount of all neighboring nodes hit queried on the target node; α is a weight parameters to adjust the query between target node and the neighboring node hits of target node, $0 < \alpha < 1$; C is a constant, to present weight of query in the target node hits; $dist(s_p, s_i)$ represents the path length from current node s_p to the target node s_i ; exponential decay parameter λ presents the influence of $dist(s_p, s_i)$ on gain value, if $\lambda \rightarrow \infty$, the gain value only can be transmitted from the target node to a node along the query path, if $\lambda \rightarrow 0$, then the gain value can be transmitted from the target node to every node along the query path without any difference, $0 < \lambda < \infty$. Assuming there is r rows on the QGM of s_p , then the probability that s_p passed the query q_j to s_i is:

$$q(s_i, q_j) = \frac{|gain_{ij}|}{\sum_{j=1}^i |gain_{ij}|} \quad (6)$$

Where: $|gain_{ij}|$ represents the total gain value from s_i passed to s_p . The specific procedure of $gain$ query routing was shown as algorithm 1.

Algorithm 1 Query gain routing algorithm

Input: wireless sensor network size, TTL, query the number of nodes.

Output: the message id, message hops.

- 1) Taking the query message $m_{es} \langle id, s_p, state, hops, q_j, path \rangle$ from node s_i
- 2) If q_j hit then on the node s_i

- 3) Forward m_{es} to neighboring node of s_i to calculate the total number of hits;
- 4) For all sensors in $mes.path$ do
- 5) If $s_p.QueryTable \neq AND q_j \in s_p.QueryTable$ then
- 6) return q_j to position j ;
- 7) Else $s_p.QueryTable.add(q_j)$;
- 8) Sensor $s_i \leftarrow mes.path.next()$
- 9) If $s_i \in s_p.SensorTable$ then
- 10) return s_i to position i ;
- 11) Else $s_p.SensorTable.add(s_i)$;
- 12) According to formula (5), calculating the gain value v_k from s_i passed to s_p
- 13) $Gain_{ij}.add(m_{es}.path, v_k)$;
- 14) $s_p \leftarrow s_i$;
- 15) End if
- 16) End if
- 17) End for
- 18) Return $mes.id, mes.hops$;
- 19) End if

In algorithm 1, the space complexity of each node is $O(m \times n)$, when QGM is full on some node, then the latest least used storage item will be replaced. Because Query Table and Sensor Table on each node are linear length, so their storage cost is negligible compared with QGM. Since each node in the query routing algorithm gain does not need the global information of any network, it only need the neighboring node's related information of this node, the successful queries target records of target node only pass along the query path between adjacent nodes, so query cost is low. Record node and transfer the load energy information during the query routing process, to balance the energy consumption of each node in query path.

3. Load Balancing Mechanism

Sensor networks become an unstable network due to the performance limitation of sensor node, it showed mainly in the node failure rate is too high. Generally, there are two ways to define life cycle of wireless sensor networks: a) All nodes in the entire network lose efficiency; b) a certain proportion of node lose efficiency, resulting in network partitioning. No matter which definition method, all of them is expected that the time of all nodes for working in the same time can be as long as possible, that is able to meet the load balance. Existing research in the designing the query routing protocols, it is considered an optimal theoretical results, while ignoring the feasibility in reality and the node load balance.

The inconsistency between the distribution of hot object and node processing capacity is the one of the reasons of leading the sensor node to load imbalance. A good query routing strategy should be able to balance the load distribution of all sensor nodes. This can avoid some of the node to become a bottleneck and thus run out of energy prematurely, and it is also important to prolong the life cycle of the entire network. Let the size of load item k on node s_i equal to $s_{i,k}$, the accessing times of load item k during some time on node s_i equals to $v_{i,k}$, the load for load item k on node s_i equals to i,k , then $L_{i,k} = s_{i,k} \cdot v_{i,k}$. The actual load L_i of node s_i is the total load of all load items on this node, that is $L_i = \sum_{k=1}^m L_{i,k}$, where m is the load item numbers of node s_i . Let the processing capacity of the node s_i be c_i , load threshold of node s_i be p_i , the load

rate for s_i be μ_i , then $\mu_i = L_i / c_i$. When a node's load rate μ_i is not over its load threshold p_i , the node will be called under loaded node; On the contrary, it will be called overloaded node. Load balancing algorithm's goal is to achieve the global load balancing under the make the load transfer times as less as possible. The measure standard for load balancing can be defined by the variance of all nodes load:

$$g(A, G) = \frac{\sum_{i=1}^m (L_i - \bar{L})^2}{m}$$

Where: A represents query routing algorithm; G stands for Wireless Sensor Networks; L_i is a sum of all load items load of node s_i ; \bar{L} -the average value of wireless sensor network load.

In order to achieve load balancing mechanism based on routing, the query message path variable stores all node identifier and load rate list of binary vectors which process this query, the specific steps were shown in Algorithm 2.

Algorithm 2 the load balancing algorithm based on routing

- 1) Take query message $m_{es} \langle id, s_p, s_{rate}, hops, q_j, path \rangle$ on node s_i
- 2) If q_j hit then on node s_i
- 3) Search path all binary vectors and identify overloaded nodes;
- 4) Transfer the redundant load from over loaded node to under loaded neighboring.
- 5) Else if $m_{es}.hops = TTL$ then
- 6) Update routing table;
- 7) End if
- 8) Forward the query message m_{es} to the neighboring node of s_i
- 9) While s_i receive ERROR message do
- 10) Update routing table ;
- 11) Forward the query message m_{es} to next neighboring node of p_i
- 12) End while

Seeing from algorithm 2, based on the route load balancing mechanism can record the energy information of node during query routing process, then balance the energy consumption between each node during query path. Furthermore, since the load transmission process only processes between nodes, so the distribution and scalability of the algorithm is good.

4. Experimental Simulation and Analysis

This simulation environment for this article use MICA2 node Crossbow produced as standard, the node's communication distance is 50 m. As the cost of wireless sensor network data transmission between nodes is far exceed the costs of data arithmetic processing, computing processing cost is ignored in simulation experiment, and only use the transferring cost to measure the total energy consumption.

Typically, it cost $\sigma_{tr} + b \delta_{tr}$ that the node sends a message of length b Byte. Which $\sigma_{tr} = 0.635mj, \delta_{tr} = 0.0145mj / Byte$. accordingly, it cost $\sigma_{tr} + b \delta_{re}$ that the node receive a message of the length b Byte, which $\sigma_{tr} = 0.389mJ, \delta_{tr} = 0.00867mJ / Byte$.

In this paper, use random data to simulate sensing data of wireless sensor network nodes in the simulation process. Specific methods are as following: First, initialize randomly the value v that each node on each dimension to be rand (0, 1). In each subsequent every experiment, the max change range of value v is $f(0 < f < 1)$, compared to the last round of experiment. Assuming the value of the i_{-th} dimension is

$v(r)[i]$ in round r , the value change is $v(r)[i] + rand(-f, f)$ in round $r + 1$, and it is always ensured to be into $[0, 1]$.

Additionally, let the query gain matrix size in the query gain routing algorithm be 20×50 , $\delta = \alpha = 0.6$, $\lambda = 2$, $c = 10$, the other experimental parameters and the default values are shown in Table 1.

Table 1. Experimental Parameters and Default Values

Experimental parameters	Default
Network coverage area/ m^2	500×500
Number of sensor nodes	500
Aggregation nodes	8
Average Metrics Network Node	13
Replica distribution	<i>Zipf</i> ($\delta = 0.83$)
Query distribution	<i>Zipf</i> ($\delta = 0.92$)
Size of sensing data/Byte	42
Size of query data/Byte	30
Average query length	3.147

In this paper, several common query routing algorithm performance were compared and analyzed, including the Rumor routing algorithm (RR), query routing tree (QRT) algorithm and query gain routing (QGR) algorithms. Which, QGR algorithm is divided into basic QGR algorithms and QGR algorithm (QGR-LB) with load balancing mechanism.

First set of experiments mainly study in the case of the different queries in different query hops distance (hops), the query success rate of using a variety of query routing strategy.

Query hops distance was mainly used in reflecting the distant degree relative data sources node to the sink node. In this set of experiments, the total number of sensor nodes using the default value 500, the simulation results shown in Table 2. In the same circumstances, the query success rate of QGR and QGR-LB is higher than the QRT and RR, when the average jump distance is equal to 7, QGR and the query success rate of QGR-LB and QRT is 15.1% higher than QRT, 46.3% higher than RR. If QGR is the same as the successful query rate of QGR-LB, then it indicates the load balancing mechanism based on route has no effect on improving the query success rate indicator.

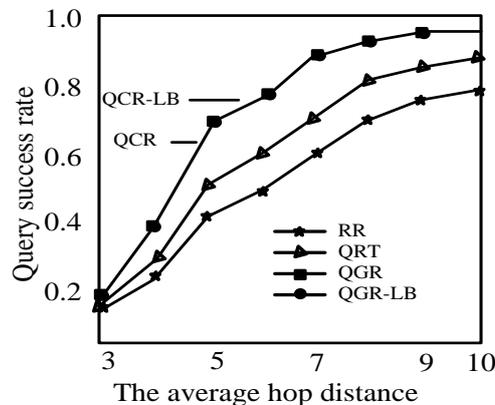


Figure 2. Effect of Average Hops on the Query Success Rate

Second set of experiments mainly compares in the cases of different node density, the query success rate for different query routing strategies. In this set of experiments, the number of sensor nodes change from 200 to 800, the simulation results shown in Figure 3. In the same circumstances, QGR and QGR-LB query success rate is higher than the QRT and RR, and with the number of nodes increasing, the range of QGR and QGR-LB query success increasing rate is significantly higher than QRT and RR, it indicates when the node number increase, the record of query routing historical information accuracy in the query gain matrix is improved accordingly.

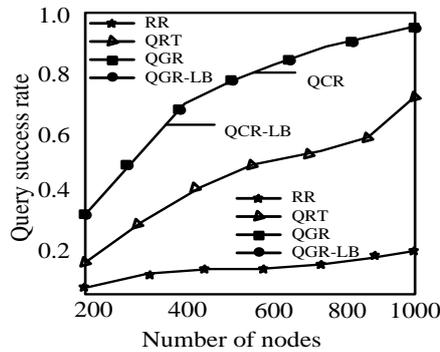


Figure 2. Effect of Node Density on the Query Success Rate

Third set of experiments mainly compares in the case of different query hops, the necessary energy consumption for using a variety query routing strategies. In this set of experiments, the total number of sensor nodes using the default value of 500, the simulation results shown in Figure 4. With the average jump distance increasing, the necessary energy consumption for a variety of query routing strategies increases accordingly. However, in the same circumstances, QGR energy consumption is lower than the QRT and RR, when the average jump distance is equal to 7; QGR energy consumption is 28.3% lower than the QRT, 38.9% lower than the RR. In addition, QGR-LB energy consumption is 18.3% lower than the QGR, it indicates the load balancing mechanism based on route can reduce energy consumption.

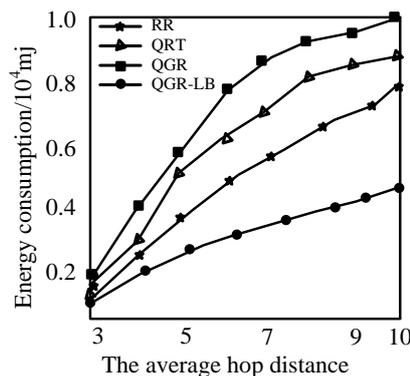


Figure 3. Average Jump Distance of the Energy Consumption

Fourth set of experiments compares in the case of the different nodes densities, the energy consumption of different query routing strategies. In this set of experiments, the number of sensor nodes change from 200 to 800, the simulation results shown in Figure 5. In the same circumstances, QGR-LB energy consumption is lowest, and with the number of nodes increasing, the energy consumption decreased slightly; the energy consumption of QGR was significantly lower than RR and QRT in the same circumstances and with the

number of nodes increasing, QGR energy consumption increasing range is lower than QRT and RR. Experiment shows that: with number of nodes increasing, the efficiency of load balancing mechanism based on the route is better.

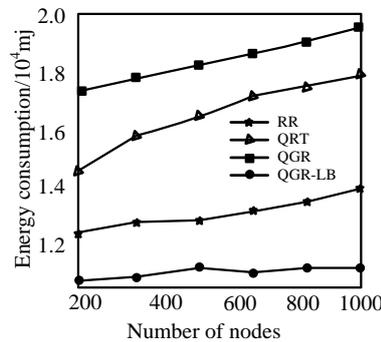


Figure 4. Impact of Node Density on Energy consumption

5. Conclusion

Query success rate and energy consumption is the key indicator of impact performance of wireless sensor network query routing. The query success rate of existing wireless sensor network query routing algorithm is not high and the energy consumption is large, and there are some limitations. This paper presents a query gain routing algorithm, in routing node selection process, without relying on global topology information of network, which has good flexibility and adaptability. In addition, the route-based load balancing mechanism can be used in the process of query routing process.

Acknowledgment

This research work is supported by the Science and Technology Department of Henan province Fund (14230410186).

References

- [1] J. M. Muhammad and N. Anjum, "Association of Moving Objects Across Visual Sensor Networks", *Journal of Multimedia*, vol. 7, no. 1, (2012), pp. 2-8.
- [2] B. Li, P. Wang and Y. Zhang, "Classification-based Multi-client Video Transmission over Heterogeneous Networks", *Journal of Multimedia*, vol. 8, no. 4, (2013).
- [3] S. Azad and M. Murshed, "An efficient transmission scheme for minimizing user waiting time in video-on-demand systems," *IEEE Communications Letters*, vol. 11, no. 3, (2007) March, pp. 285-287, <http://dx.doi.org/10.1109/LCOMM.2007.061469>.
- [4] P. Chou and Z. Miao, "Rate-distortion optimized sender-driven streaming over best-effort networks," *2001 IEEE Fourth Workshop on Multimedia Signal Processing*, (2001), pp. 587-592, <http://dx.doi.org/10.1109/MMSP.2001.962796>.
- [5] D. L. Pintado, "Diffusion in Complex Social Networks", *Games and Economics Behavior*, vol. 62, no. 2, (2008), pp. 573-590, <http://dx.doi.org/10.1016/j.geb.2007.08.001>.
- [6] A. Borodin, Y. Filmus and J. Oren, "Threshold Models for Competitive Influence in Social Networks", *Internet and Network Economics*, vol. 6484, (2010), pp. 539-550, 2010.
- [7] C. D. Vleeschouwer, J. F. Delaigle, B. Macq, "Invisibility and application functionalities in perceptual watermarking: An overview", *Proceedings of the IEEE*, vol. 90, no. 1, (2002), pp. 64-77.
- [8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient routing protocols for wireless microsensor networks", *Proceeding the 33rd Hawaii International Conference on System Sciences*, pp.8020-8030, 2000. <http://dx.doi.org/10.1109/HICSS.2000.926982>.
- [9] P. Dubey, R. Garg, B. D. Meyer, "Competing for customers in a social network: The quasi-linear case", *Internet and Network Economics*, vol. 4286, (2006), pp. 162-173.
- [10] S. Suthaharan, "Fragile Image Watermarking Using a Gradient Image for Improved Localization and Security", *Pattern Recognition Letters*, vol. 25, no. 16, (2004), pp.1893-1903.
- [11] B. Shen, S. Zhang, and Y. Zhong, "Cluster-based routing protocols for wireless sensor networks", *Journal of Software*, vol. 17, (2006), pp. 1588-1600.

- [12] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion", *Inter. Journal of CV*, vol. 25, no.1, (1997), pp. 23-48.
- [13] X. Han and Li Niu, "Word of Mouth Propagation in Online Social Networks, *Journal of Networks*", vol. 7, no. 10, (2012).
- [14] L. Yan, Y. Pan, and J. Zhang, "Trust cluster head election algorithm based on ant colony systems", *Proceedings of the 3rd International joint Conference on Computational Science and Optimization*, (2010), pp. 419-422., <http://dx.doi.org/10.1109/CSO.2010.205>.
- [15] S. Bartlett, G. Littlewort, G. Frank, C. Lainscsek, I. Fasel and J. Movellan, "Fully automatic facial action recognition in spontaneous behavior", In *Proc. Conf. Automatic Face & Gesture Recognition*, (2006), pp. 223-230.
- [16] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: A stable election protocol for clustered heterogeneous wireless sensor networks", *Proceeding of the International Workshop on SANPA*, (2004), pp. 660-670.
- [17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, (2001).
- [18] J. He, Y. Geng and K. Pahlavan, "Modeling Indoor TOA Ranging Error for Body Mounted Sensors", 2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), Sydney, Australia, (2012) September, pp. 682-686.
- [19] Y. Geng, J. Chen and K. Pahlavan, "Motion detection using RF signals for the first responder in emergency operations", A PHASER project, 2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), London, Britain, (2013) September.
- [20] L. Zhihan, L. Feng, H. Li, and S. Feng. "Hand-free motion interaction on Google Glass." In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, (2014), p. 21, ACM.
- [21] Z. Chen, S. M. Arisona, X. Huang, M. Batty and G. Schmitt, "Detecting the dynamics of urban structure through spatial network analysis", *International Journal of Geographical Information Science*, vol. 28, no. 11, (2014), pp. 2178-2199.
- [22] L. Wubin, J. Tordsson and E. Elmroth, "An aspect-oriented approach to consistency-preserving caching and compression of web service response messages", In *Web Services (ICWS)*, 2010 IEEE International Conference on, (2010), pp. 526-533. IEEE.
- [23] L. Zhihan, L. Feng, S.Feng, and H. Li, "Extending Touch-less Interaction on Vision Based Wearable Device", *Virtual Reality (VR)*, (2015), iEEE.
- [24] Z. Mengxin, Z. Lv, X. Zhang, G. Chen, and K. Zhang, "Research and Application of the 3D Virtual Community Based on WEBVR and RIA", *Computer and Information Science*, vol. 2, no. 1, (2009), p. 84.

Author



Jiang Huilin was born in Henan, China, in 1977. He received B.S. and M.S. degrees in computer science and technology from Hernan Normal University and Nanjing University of Science and Technology, in 2003 and 2012, respectively. His research interests direct computer network and embedded system. In 2009 he achieved his intermediate title and now is a lecturer with School of Computer and Information Technology, Shangqiu Normal University, Henan, China.