Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 429719, 8 pages
doi:10.1155/2012/429719

Research Article

An Energy-Efficient and Fault-Tolerant Convergecast Protocol in Wireless Sensor Networks

Ting Yang,[1] ChunJian Kang,[1] and Guofang Nan[2]

[1] *School of Electrical Engineering and Automation, Tianjin University, Tianjin 300072, China*
[2] *Institute of Systems Engineering, Tianjin University, Tianjin 300072, China*

Correspondence should be addressed to Ting Yang, yangting@tju.edu.cn and Guofang Nan, gfnan@tju.edu.cn

Received 19 April 2012; Revised 9 July 2012; Accepted 9 July 2012

Academic Editor: Jianhua He

The simple graph theory is commonly employed in wireless sensor networks topology control. An inherent problem of small-granularity algorithms is the high computing complexity and large solution space when managing large-scale WSNs. Computed transmission paths are of low fault tolerance because of unattended sensor nodes and frail wireless transmitting channels. This paper uses hyper-graph theory to solve these practical problems and proposes a spanning hyper-tree algorithm (SHTa) to compute the minimum transmitting power delivery paths set for WSNs convergecast. There are three main contributions of this paper: (1) we present a novel hyper-graph model to abstract large-scale and high connectivity WSNs into a robust hyper-tree infrastructure; (2) we present a precise mathematical derivation that solves the "hyper-tree existence" problem; (3) SHTa is proposed to compute the delivery paths set, which is the minimum power transmitting convergecast hyper-tree. Variable scale hyper-edges represented as computing units limit solution space and reduce computing complexity. Mutual backup delivery paths in one hyper-edge improve the capability of fault tolerance. With experiment results, SHTa computes short latency paths with low energy consumption, compared with previous algorithms. Furthermore, in dynamic experiments scenes, SHTa retains its robust transmitting quality and presents high fault tolerance.

1. Introduction

Self-organized wireless sensor networks can be used to cooperatively detect and perceive real objects. Sensors can communicate and exchange information among themselves without human intervention. This is achieved by integrating technologies, including sensors, embedded calculations, distributed information processing, and wireless communication. Wireless sensor networks have huge potential in civil and military applications, such as smart grid, smart home, healthcare monitoring, and intelligent transport.

Self-organized wireless sensor networks are made up of highly distributed systems of small-size, wireless unattended sensors. Each sensor is capable of detecting devices' current operating conditions, such as temperature, noise, vibration, or output signals. This data is preprocessed, transmitted, and exchanged in a machine-to-machine (M2M) network [1, 2]. There is a need for reliable, scalable, and smart protocols and algorithms for self-organized M2M networks or sensor networks.

In traditional communication networks, simple graph theory is always used [3, 4]. But a large-scale self-organized wireless sensor network consists of hundreds or thousands of nodes with a complex topology. Hence, a large number of the control massages are required to establish transmission paths. On the other hand, because of the low reliability of the sensor nodes and wireless communication links, many real-time control messages have to be used to maintain an established path. These tasks use significant amount of bandwidth and consume the extra energy.

To solve this problem, in this paper, we used the hyper-graph theory and proposed Spanning Hyper-Tree algorithm (SHTa) to create a concise and robust hyper-graph infrastructure for large-scale and high connectivity self-organized wireless networks.. Based on the best of our knowledge, it is the first hyper-graph model for self-organized wireless

Table 1: Total sum cost with two transmitting methods.

	Total weight of SPT = 3.82			Total weight of MST = 3	
	Routing based on SPT			Routing based on MST	
Seq and time	Routing path	Consumption	Seq and time	Routing path	Consumption
1	*a–S*	1.41	1	*a-b*	1
2	*b–S*	1.41	1	*c-b*	1
3	*c–S*	1	2	*b–S*	1

networks architecture. Because a dynamic hyperedge is the minimum computing unit during routing in this type of hyper-graph architecture, fewer packets are used, which saves energy and prolongs the network's lifetime. More than one connected pairs in a hyperedge provides high bandwidth and low loss rate during transmission. This effectively improves network fault tolerance. Moreover, SHTa solves the "hyper-tree existence uncertainty problem," which is a new problem that differs from the simple graph model. An axiom "any graph has its spanning tree" is invalidated in a hyper-graph, that is, not each hyper-graph exists spanning hyper-tree with loop-free. SHTa presents an effective spanning hyper-tree method and we proposed the strict mathematical proof to prove the certainty theorem.

The remainder of the paper is organized as follows. Section 2 introduces some background material on wireless communication network architecture and optimal routing problems. Wireless self-organized sensor networks' hyper-graph model is presented in Section 3. Section 4 describes the SHTa in detail, followed by validity proof. Section 5 proposes the computer simulation and evaluation; finally, Section 6 is the conclusion and outline of future research.

2. Convergecast with Data Aggregation in Wireless Sensor Networks

For peer-to-peer (P2P) communication model, Dijkstra and Bellman-Ford algorithms are often employed to build a shortest path tree (SPT), such as OSPF used in IP backbone networks. Each router with OSPF stores an SPT in which the root is itself. Packets are transmitted following SPT's branches to arrive the minimum cost.

Different from P2P, self-organized wireless sensor network collects data from each sensor nodes to "Sink," called convergecast. During transmission, data aggregation is used to eliminate the redundancy in collection data. Many algorithms are presented to establish data aggregation tree, such as EADAT [5], E-Span [6], and HEED [7]. These algorithms set transmitting energy consumption as link weight and build SPT as an aggregation tree. Reference [8] presented the DCTC algorithm to detect and track a mobile target. DCTC used Dijkstra to establish collection tree, which is also an SPT.

Not every data packet will be transmitted from source to destination, due to data aggregation, in the intermediate nodes. A wireless sensor network is generalized as data center, and the optimum number of transmissions required per datum in the DC (Data Centre) is equal to the number of edges in the Minimum Steiner Tree (MST). Therefore, MST, not SPT, is the truly minimized sum cost tree in convergecast protocols with data aggregation.

Figure 1 shows an example to explain this optimization problem. Three nodes transmit information to Sink. SPT and MST are shown, respectively, in Figures 1(a) and 1(b). Table 1 presents the two routing configurations: the total cost of SPT method is 3.82, larger than MST. Therefore, MST is better than SPT. If the weight of edges is defined as energy consumption, MST is just the optimal energy consumption tree in the wireless sensor network.

3. Hypergraph Model for Wireless Self-Organized Sensor Networks

Whether for IP backbone network, cellar mobile network or Ad-hoc network, the simple graph theory is the main tool for research on architecture control [8–11] and counting routing protocols [3, 4, 12–14]. But in large-scale wireless self-organized sensor networks, the number of sensor nodes can be hundreds or thousands of times of that of backbone network or mobile network. Each node can connect with any one neighbour by omnidirectional antenna, which creates high node connectivity and complication in topology controlling. A simple graph algorithm with tiny granularity often has high computing complexity and uses a large amount of memory. On the other hand, in wireless sensor networks, a single transmitting path has a low fault tolerance level because of the low reliability of sensor nodes and wireless links. During data transmission, lots of control messages need to be transmitted frequently to maintain the connectivity of a delivery path, which may use lots of links' bandwidth and consume significant amount of energy.

To solve this problem, Hyper-graph theory is used as a novel mathematical tool to generalize high connectivity wireless self-organized networks into concise and robust hyper-graph infrastructure. As far as we know, it is the first hyper-graph architecture model in wireless sensor network. In the model, special nodes and connected edges among them are generalized as hyper-edges. With the growth of hyper-edges, as the minimum computing unit, fewer extra packets are used and the energy consumption is effectively reduced.

Proposition 1. *Let* $X = \{x_1, x_2, \ldots, x_n\}$ *be a finite set, and let* $\varepsilon = \{E_i \mid i \in I\}$ *be a family of subsets of* X. *If the following two conditions are satisfied: (1)* $E_i \neq \varnothing$ $(i \in I)$; *(2)* $\bigcup_{i \in I} E_i = X$, *the couple* $H = (X, \varepsilon)$ *is called a hyper-graph.* $|X| = n$ *is called the*

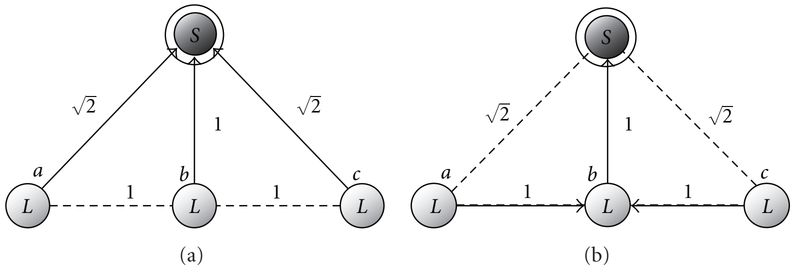
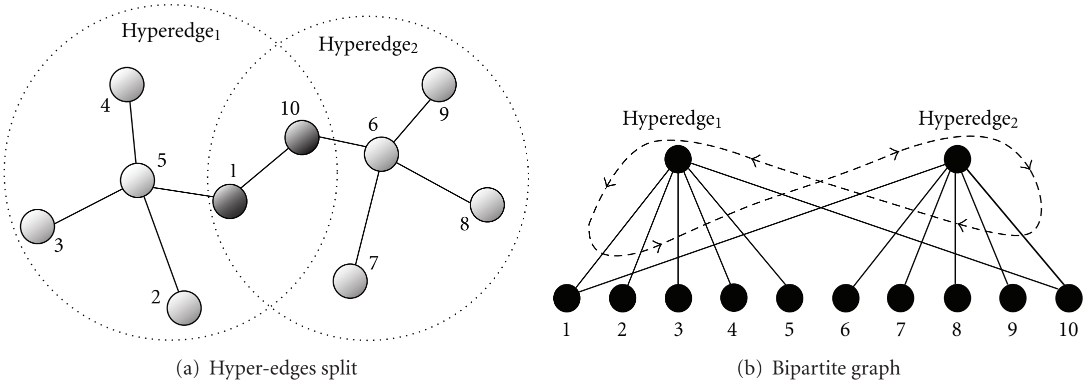


Figure 1: (a) Routing based on SPT. (b) Routing based on MST.



(a) Hyper-edges split

(b) Bipartite graph

Figure 2: Hyper-graph model.

order of this hyper-graph. The elements $x_1, x_2, \ldots, x_n$ are called vertices and the $E_1, E_2, \ldots, E_m$ are called hyper-edges [15].

We describe a wireless self-organized sensor network as a hyper-graph $H = (X, \varepsilon)$, in which $X = \{x_1, x_2, \ldots, x_n\}$ is the sensor nodes set. Special characteristics of nodes are represented as a hyper-edge, that is, $E_i = \{N_1, N_2, \ldots, N_j, e_1, e_2, \ldots, e_k\}$, and hyper-edge set is $\varepsilon = \{E_1, E_2, \ldots, E_m\}$. It is obvious that cluster in simple graph is a special type of hyper-edge, and hyper-graph is the extended cluster.

In the hyper-graph model, we should also establish MST for optimal convegercast. But the binary relation of hyper-edge and vertices in hyper-graph is not the one-to-one mapping relation of vertices and edges as it is in a simple graph, which is more complex. Therefore, the axiom "any graph has its spanning tree" is invalid in hyper-graph, that is, not each hyper-graph exists spanning hyper-tree with loop-free. A hyper-graph example with no hyper-tree is shown in Figure 2. Two hyper-edges are split and Theorem 2 proposed hyper-tree does not exist, because of existing the loop (1-Hyperedge$_1$-10-Hyperedge$_2$-1) in the bipartite graph $G\langle H\rangle$, shown in Figure 2(b).

Theorem 2. *Hyper-graph H is a hyper-tree, if and only if the bipartite graph $G\langle H\rangle$ is a tree.*

To ensure one hyper-graph certainly has hyper-tree, we proved two conditions must be satisfied:

(i) *if $E_i \cap E_j \neq \emptyset$, called relative hyper-edges, then $|E_i \cap E_j| = 1$;*

(ii) *if condition one is satisfied and if $|E_i| \neq 2$, it must have $|E_j| = 2$.*

The precise mathematical proof is shown here. Firstly, if there is no hyper-cycle in $H = (X, \varepsilon)$, the proposition is true. Otherwise, if there are hyper-cycles, a break-cycle method is used. In a hyper-clcye C, three connecting hyper-edgs E_i, E_j, E_k always can be found easily, then there must be a 2-degree chained hyper-edge among them, assuming $|E_i| = 2, x_1, x_2 \in E_i, E_i \cap E_j = \{x_1\}, E_i \cap E_k = \{x_2\}$. When E_i is broken, C is not a closed cycle. But x_1 and x_2 still belong to the E_j, E_k, which is $x_1, x_2 \in \varepsilon', \varepsilon' = \varepsilon - E_i$. Then a new hyper-graph $H' = (X', \varepsilon')$ is spanned, where $X' = X$ and H' is also connective. Repeating this process till there is no hyper-cycle, the result is a hyper-graph $T = (X', \varepsilon'), X' = X$ and $\varepsilon' \subseteq \varepsilon$, which is the final spanning hyper-tree.

In Section 4, we presented a novel topology controlling algorithm to split hyper-edge, establish hyper-graph with satisfying the above two conditions, and span the minimum hyper-tree for minimum energy consumption convergecast.

4. Minimum Spanning Hyper-Tree Algorithm

In the implement of SHTa, a type of generalized synchronization mechanism with "synchronous round" was used. Firstly, we describe this synchronization mechanism.

Time synchronization is an important feature of distributed systems including wired and wireless communication systems. Many time synchronization schemes were designed including GPS [16] and Network Time Protocol

(NTP) [17] used in IP networks applications. In M2M and sensor networks, time synchronization is also used frequently for various purposes including sensor data fusion, coordinated actuation, and power-efficient duty cycling: for example integrating a time series of proximity detections into a velocity estimate; measuring the time of flight of sound for localizing its source; distributing a beam forming array; suppressing redundant messages by recognizing that they describe duplicate detections of the same event by different sensors; or supporting energy efficient scheduling and power management. Now, many good time synchronization algorithms, such as Reference Broadcast (RBS [18]), TINY/MINI-SYNC [19], and Level Synchronization [20], are presented to provide time accuracy in wireless self-organized sensor networks.

Compared with accurate time slots synchronization, generalized synchronization mechanisms with "synchronous round" can save a large number of timescale check packets, which ensures the accurate time synchronisation, and reduce the complexity of designing communication protocols, therefore reducing the transmission energy consumption. In a generalized synchronous mechanism, each processor unit should complete two steps during one synchronous round: in the first step, the processor transmits event driven messages to its neighbor; in the second step, processor switches its current state with a state transition function, once it has received any valid messages.

When synchronous network is a deterministic system, a state transition function with the same valid input must achieve the same output in each time. Mapping the two steps onto a sensor node processor, the following two operations would be implemented.

(i) Each main hyper-edge ε_m^k exchanges information with its neighbours and builds a subminimal power chained hyper-edge set $\{\varepsilon_{j1}^k, \varepsilon_{j2}^k, \ldots\}$.

(ii) A minimal power chained hyper-edge $\varepsilon_{m,n}^k$ is elected in set $\{\varepsilon_{j1}^k, \varepsilon_{j2}^k, \ldots\}$. Then ε_m^k, ε_n^k, and $\varepsilon_{m,n}^k$ merge into new main hyper-edge $\varepsilon_{m,n}^{k+1}$.

Two types of messages are employed in the above operations: (i) *Op_HC*: excite *HC* operation packet; (ii) *R_mPCHe*: request minimal power chain hyper-edge packet. The structures of the two types of packets are shown in Algorithm 1.

Without loss of generality, we suppose SHTa implements at k_{th} "synchronous round." Each main hyper-tree ε_m^k initiate to search for a minimal power chain hyper-edge connected with its neighbours. Nodes v_i, at the edge of ε_m^k, broadcasts *R_mPCHe* message. Any node receiving this message will implement SHTa as in Algorithm 2.

As soon as two main hyper-edges ε_m^k, ε_n^k confirm their conjunct minimal power chain hyper-edge, *Op_HC* messages are broadcasted in these three units. Any one node received this message will implement the operations described in Algorithm 3.

Whenever SHTa cannot find any new chained hyper-edge ε_j^k or implement consolidation operation, the algorithm stops and the spanning forest gathers into a hyper-tree. In the following section, we prove that this hyper-tree is just a minimum spanning hyper-tree.

We rewrote the conclusion of the spanning hyper-tree from SHTa algorithm: In hyper-graph $H = (X, \varepsilon), \cup_j E_j = \{(X, \varepsilon) : 1 \leq j \leq k\}$ is one of the hyper-graph's spanning forests. If e is the minimum weight chain hyper-edge in the set $\cup_j E_j$, there must be a hyper-tree, which is made up of $\cup_j E_j$ and e. Moreover, this hyper-tree is the minimum hyper-tree in all of the spanning hyper-trees which include $\cup_j E_j$.

Proof by Contradiction. Suppose that the conclusion is erroneous, that is, there is a hyper-tree T, which includes $\cup_j E_j$, but does not include the e. And T is strictly less than any other hyper-tree, which includes the $\cup_j E_j$ and the e. Now put e into T, and then obtain the graph T'. Obviously, there is a cycle in T', which includes another chain hyper-edge e', $e' \neq e$, and $e' \in \cup_j E_j$.

Based on the definition, weight$(e') \geq$ weight(e) is obtained. e' can be safely deleted from T'. And another hyper-tree T'' is made, including $\cup_j E_j$ and e. the power of T'' is not larger than the power of T. There is a contradiction for T. The supposition is in error, and the original proposition is true.

5. Computer Simulation

This section evaluates the performance of the novel algorithm using simulation. Firstly, seven different sensor scenes are studied, in a $200 \times 200\,\text{m}^2$ square area, and a number of sensor nodes are uniformly dispersed, ranging from 50 to 350 nodes with increment step of 50 nodes. Each node has a radio range of 40 m. We used this environment to simulate how different network density affects the energy consumption during the processing of spanning tree or hyper-tree. Then, transmission performance metrics, average latency and loss packets ratio, are evaluated when data packets are delivered following SHTa, compared with Directed Diffusion (DD) [21] and its improved algorithm EADD [22]. We use the same parameter as [21]: (1) using the 802.11 MAC protocol to ensure the data link connected; (2) setting the idle time power dissipation about 35 mW, receiving data power dissipation 395 mW, and transmiting data power dissipation 660 mW; (3) setting events modelled as 64 bytes and information control packet 36 bytes. Finally, in the simulations, we use a fixed events generated model—after every ten-second interval, ten nodes were randomly selected as sensor sources and generated constant bit rate (CBR) data streams with packet intervals of 0.1 seconds. The duration of each data streams is 5 seconds.

We first compute the maximum and average nodes' degree and the standard deviation of nodes' degree in seven different scenes to analyze the network density, shown in Table 2.

Figure 3 shows the average dissipated energy per packet as a function of networks size. DD and EADD have almost the same energy consumption and a half less than flooding. SHTa consumes less energy than DD and EADD. With the increase of the network size, SHTa can save 23.7% energy

```
Packet Op_HC struct:

typedef struct Op_HC_st          // the struct of Op_HC packet
{
    ULONG seq;                   // the serial number of the Op_HC packet
    UNIT new_mHe_ID;             // new consolidation main hyper-edge ID
    UINT Og_mHe_ID_1;            // original main hyper-edge ID
    UINT Og_mHe_ID_2;            // original main hyper-edge ID
    UINT Og_cHe_ID;              // chain hyper-edge ID between the above two main hyper-edges
    ULONG ttl;                   // the packet's lifetime
} Op_HC;

Packet R_mPCHe struct:

typedef struct R_mPCHe_st        // the struct of R_mPCHe packet
{
    ULONG seq;                   // the serial number of the R_mPCHe packet
    ULONG Source_E_ID;           // the R_mPCHe packet's source main hyper-edge ID
    UINT Source_ID;              // the R_mPCHe packet's source node ID
    UINT Dest_ID;                // the R_mPCHe packet's destination node ID
    ULONG Tran_Pw                // transmission power in the chain Hyper-edge
    ULONG ttl;                   // the packet's lifetime
} R_mPCHe;
```

Algorithm 1: Structures of the *Op_HC* and *R_mPCHe* packets.

```
$v_j \in V$ receive R_mPCHe message and implement the following operation in SHTa

if ($v_j \leftarrow$ R_mPCHe)
{    if (R_mPCHe.ttl <= 0)
         { ignore this R_mPCHe packet;
           return UNSUCCESS;                          // can not recover the new delivery path
         }
      if (MPCHe.Source_E_ID == $v_j$.He_ID)           // $v_j$ and R_mPCHe belong to the same main hyper-edge
         { ignore this R_mPCHe;
           R_mPCHe.ttl= R_mPCHe.ttl-1;
         }
      else                                            // $v_j$ receive this R_mPCHe at first time
         if ($v_j \in \varepsilon_n^k$)
            R_mPCHe.Tran_Pw = Pw ($v_i, v_j$);        // sub-minimal power is transmission power in $\varepsilon_j^k = \{v_i, v_j\}$
     if (Min_Pw(*$\varepsilon_{j,i}^k$) == Pw ($v_i, v_j$))   // $\varepsilon_{m,n}^k$ is the minimal power in set $\{\varepsilon_{j1}^k, \varepsilon_{j2}^k, \ldots\}$.
        $\varepsilon_m^k$ Broadcast Op_HC;            // initiate Hyper-edges' Consolidation operation
 }
```

Algorithm 2: Implement flow while node receiving *R_mPCHe* message.

```
$v_j \in V$ receive Op_HC message and implement the following operation in SHTa

if ( $v_j \leftarrow$ Op_HC)                          // implement Hyper-edges' Consolidation operation
   if ($v_i \in \varepsilon_m^k || v_i \in \varepsilon_n^k$)   // only the nodes in $\varepsilon_m^k$ and $\varepsilon_n^k$ implement the HC operation
      {  $v_i$. E_ID = Op_HC.new_mHe_ID          // new consolidation main hyper-edge ID: $\varepsilon_m^{k+1}$;
         $v_i$. syn_round = $v_i$. syn_round+1;    // put into the next synchronous round
         return SUCCESS;                         // HC operation success and generate new main hyper-edge
      }
```

Algorithm 3: Implement flow while node receiving *Op_HC* message.

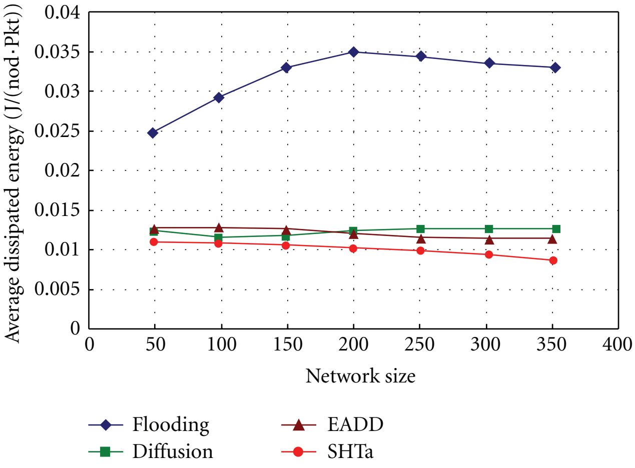


Figure 3: Average dissipated energy in different network densities.

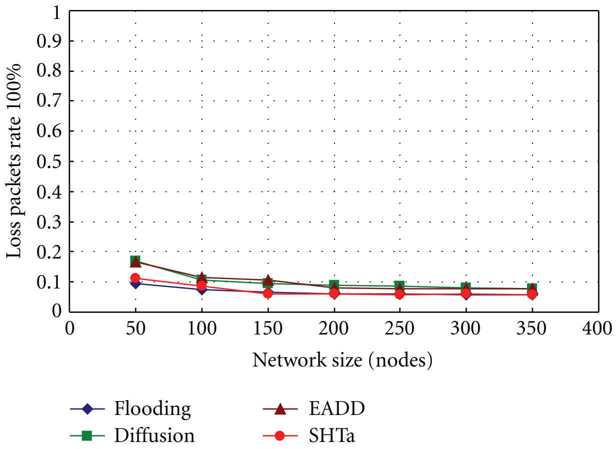


Figure 5: Packets loss rates in different densities of networks.

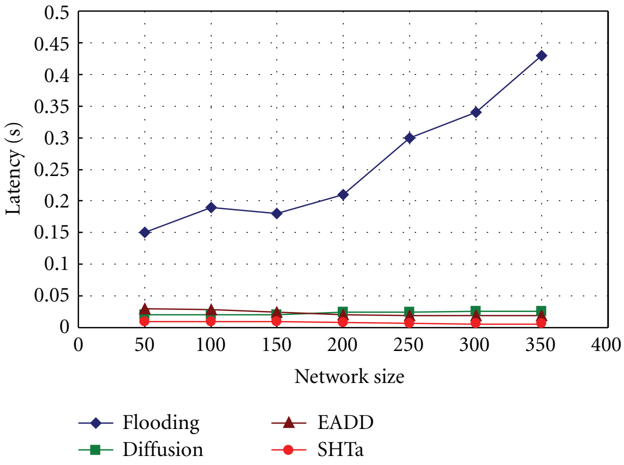


Figure 4: Average latency in different network densities.

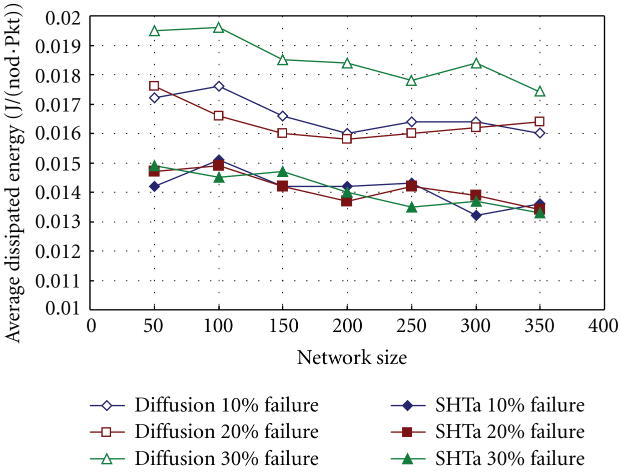


Figure 6: Average dissipated energy in dynamic experiments.

Table 2: Maximum node degree, average nodes' degree, and the standard deviation of nodes' degree in different network size.

N	50	100	150	200	250	300	350
$\max\{d_i\}$	12	21	33	39	50	56	61
$\text{avg}\{d_i\}$	5.23	10.50	15.84	20.89	26.25	31.37	36.85
$\sigma(d)$	2.18	3.90	5.20	6.06	7.60	8.66	9.45

that of directed diffusion. It is because SHTa used hyper-edge as computing granularity, with consolidation operation, that hyper-edges become larger and less in the networks during the SHTa processing, which is completely different from the trivial nodes operation. Therefore it effectively reduces the number of overhead packets and reduces the size of the solution space, which results in the reduction of energy consumption and a prolonged network's lifetime.

Figure 4 plots the average latency observed as a function of network size. Using the shortest path, DD and EADD algorithms have lower delays than flooding algorithm. Because more available energy can give nodes a faster response time, EADD just selects these vigorous nodes as relay stations and achieves lower delay than DD. Differing from the shortest edge path algorithm, SHTa uses hyper-edge, in which a set of identified nodes and edges composing multiple paths transmit information at the same time; therefore the lowest delays can be reached in four algorithms.

Figure 5 presents another performance metric-loss packets rate. SHTa and flooding have lower value than DD and EADD. Compared with signal path in DD or EADD, multiple delivery paths in one hyper-edge in SHTa or duplication flooding in the Flooding algorithm improves the transmission reliability.

We also study the impact of dynamics in wireless sensor networks with 10%, 20%, and 30% random failure nodes. Figure 6 presented the average dissipated energy per packet as a function of network size. By increasing the failure percentage from 10% to 30%, both of DD and SHTa algorithms significantly consume more energy in transmitting per packet, the increase rate of DD is 33.40%, more than 28.48% of SHTa.

Figure 7 presents the average latency measurement. The results show that SHTa also provides the lower average

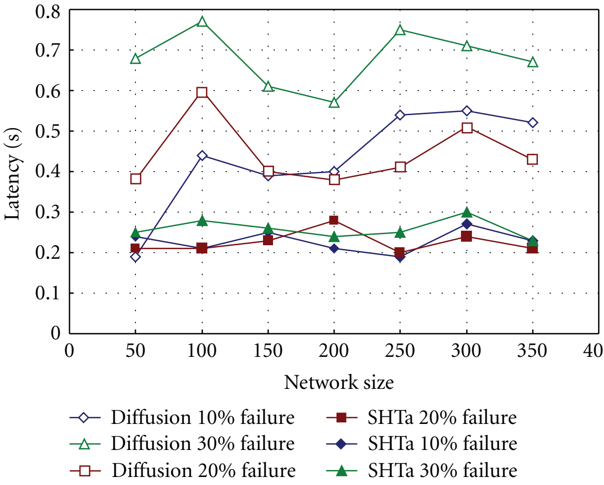


Figure 7: Average delay in dynamic experiments.

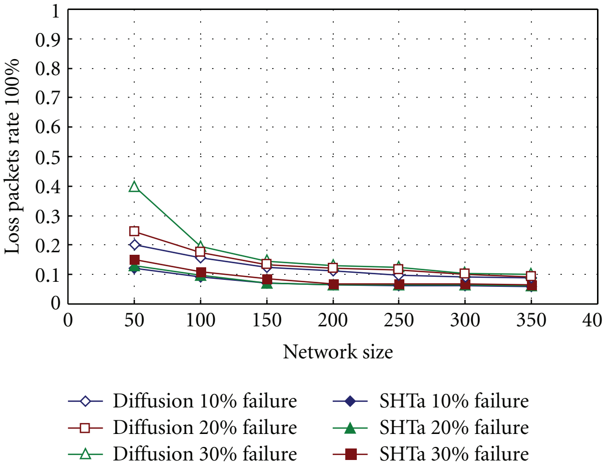


Figure 8: Packets loss rate in dynamic experiments.

delay for various fault percentages, that is, average 0.259 s when 30% of nodes fail. This is mainly because SHTa presents mutual backup delivery paths in one hyper-edge, which improves the capability of fault tolerance. In the final experiment, we evaluated the loss packets rate when the fault percentage of faulty nodes is increased. Figure 8 clearly shows that SHTa drops lower number of data packets compared with DD protocols, that is, 8.82% when 30% of nodes fail, and DD performs slightly worse, 17.12% for the same situation. All of results fairly present that SHTa is great robustness and can offers significant performance gain in networks with high fault percentage.

6. Conclusion

To consistently provide reliable communication services for machine to machine applications, scalable and smart network architecture control algorithms are needed for wireless self-organized communication networks. This paper generalizes large-scale wireless self-organized sensor networks into concise and robust hyper-graph infrastructure and proposes an algorithm called SHTa to achieve minimum spanning hyper-tree. We proved algorithm's validity with mathematical deduction and computer simulation. Based on experimental results, the SHTa algorithm can save more energy and have lower latency and packets loss rates than previous algorithms, and the algorithm is more robust in the dynamic experiments. All of these results show that SHTa is an effective technique for wireless sensor networks and M2M applications.

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China No.61172014 and no. 60702037; Natural Science Foundation of Tianjin no. 12JCZDJC21300 and 09JCYBJC00800; National Basic Research Program of China (973 Program) no. 2009CB219700. The authors thank their colleagues in the laboratory.

References

[1] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.

[2] B. S. Choi and J. J. Lee, "Sensor network based localization algorithm using fusion sensor-agent for indoor service robot," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1457–1465, 2010.

[3] I. C. Paschalidis and B. Li, "Energy optimized topologies for distributed averaging in wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2290–2304, 2011.

[4] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM'00)*, pp. 243–254, August 2000.

[5] M. Ding, X. Cheng, and G. Xue, "Aggregation tree construction in sensor networks," in *Proceedings of the 58th IEEE Vehicular Technology Society (VTC'03)*, vol. 4, Orlando, Fla, USA, October 2003.

[6] M. Lee and V. W. S. Wong, "An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks," in *Proceedings of the IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing (PACRIM'05)*, pp. 300–303, August 2005.

[7] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[8] C. C. Tseng, H. H. Chen, K. C. Chen, S. C. Lo, and M. Y. Liu, "Quality of service-guaranteed cluster-based multihop wireless ad hoc sensor networks," *IET Communications*, vol. 5, no. 12, pp. 1698–1710, 2011.

[9] J. Baek, S. K. An, and P. Fisher, "Dynamic cluster header selection and conditional re-clustering for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2249–2257, 2010.

[10] M. T. Hajiaghayi, N. Immorlica, and V. S. Mirrokni, "Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1345–1358, 2007.

[11] Y. Liu, Q. Zhang, and L. Ni, "Opportunity-based topology control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 3, pp. 405–416, 2010.

[12] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "A spatiotemporal communication protocol for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 995–1006, 2005.

[13] E. Lee, S. Park, F. Yu, and S. H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1433–1441, 2010.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion:a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM'00)*, pp. 56–67, August 2000.

[15] C. Rerge, *Graph and Hypergraph*, North-Holland, Amsterdam, The Netherlands, 1973.

[16] A. Seuret, F. Michaut, J. P. Richard, and T. Divoux, "Networked control using GPS synchronization," in *Proceedings of the American Control Conference*, pp. 4195–4200, June 2006.

[17] T. Neagoe, V. Cristea, and L. Banica, "NTP versus PTP in computer networks clock synchronization," in *Proceedings of the International Symposium on Industrial Electronics 2006 (ISIE'06)*, pp. 317–362, Montreal, Canada, July 2006.

[18] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th Symposium on Operating systems Design and Implementation*, Boston, Mass, USA, December 2002.

[19] M. L. Sichitiu, C. Veerarittiphan, and Simple, "Accurate Time Synchronization for Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC'03)*, 2003.

[20] S. Ganeriwal, R. Kumar, S. Adlakha, and M. Srivastava, "Network-wide time synchronization in sensor networks," Technical Report UCLA, 2002.

[21] J. Choe and K. Kim, "EADD: energy aware directed diffusion for wireless sensor networks," in *Proceedings of the International Symposium on Parallel and Distributed Processing with Applications (ISPA'08)*, pp. 779–783, Sydney, Australia, December 2008.

[22] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

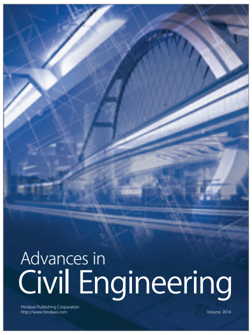
Journal of
Engineering

The Scientific World Journal

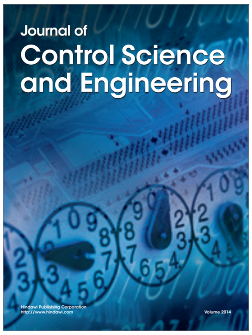
International Journal of
Rotating Machinery

Journal of
Sensors

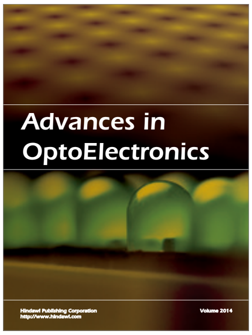
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering

Journal of
Robotics

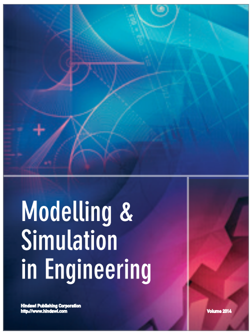

Journal of
Control Science and Engineering

Journal of
Electrical and Computer Engineering

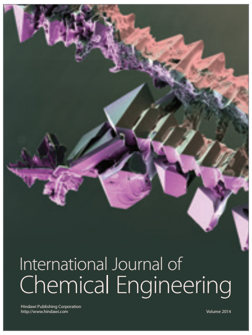

Advances in
OptoElectronics

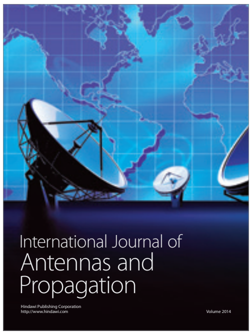
VLSI Design

International Journal of
Navigation and Observation

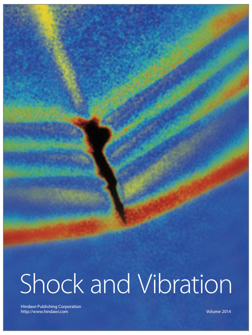

Modelling & Simulation in Engineering

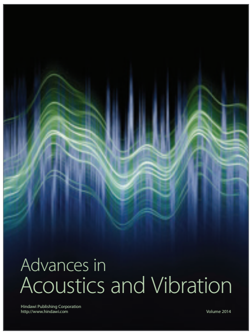
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration