

Synchronous Aggregation Scheduling with Minimal Latency in Multihop SensorNet

^{1,3}Liqun Shan, ^{1,2}Jinkuan Wang, ³Yanchao Zhao and ³Yanchang Liu

¹School of Information Science Engineering, Northeastern University, Shenyang, 110819, China

²Engineering Optimization and Smart Antenna Institute, Northeastern University at Qinhuangdao, Qin huang dao, 066004, China

³Northeast Petroleum University, Daqing 163318, China

Abstract: This study has addressed the problem of data aggregation scheduling problem to minimize the latency in wireless sensor networks (WSNs). An efficient distributed synchronous aggregation scheduling method is proposed to structure a collision-free schedule for data aggregation in WSNs. By using a Connected Dominating Set (CDS) as an aggregation tree we implement the synchronous aggregation scheduling. We prove that the latency of the aggregation schedule generated by our algorithm is at most $4\delta+15R-16$ time-slots where R is the network radius and δ is the maximum node degree in the communication graph of the original network. Analysis and simulation results show the validity and superiority of the algorithm.

Key words: Wireless sensor networks, connected dominating set, aggregation scheduling, latency, collision-free schedule

INTRODUCTION

Data aggregation is a primitive operation in wireless sensor network applications, such as emergency medical situations, environmental monitoring and battlefield surveillance. In most applications, quite often we need to gather data from sensors to a fixed sink node that collects a packet from every other node and every intermediate node combines all received packets with its own packet into a single packet of fixed-size according to some aggregation function such as logical and/or, maximum, or minimum. This type of application is called data aggregation (Krishnamachari *et al.*, 2002). We focus on generic environmental monitoring applications which collect sensing data continuously over a long period of time and aim at providing collision-free and shorter packet delay.

In data aggregation, many issues need to be resolved such as energy conservation, deployment strategies, routing in dynamic environment, latency and so on (Idris *et al.*, 2009; Guo *et al.*, 2010). Previous researches on in-network data aggregation seldom consider the collision problem but leave it to the MAC layer. Solving this problem in MAC layer results in latency too high to be practical. Recently, a few researchers begin to consider the collision problem for reducing data aggregation latency and try to construct a feasible schedule to eliminate collisions during data aggregation (Kesselman and Kowalski, 2005; Liu *et al.*, 2010a, b) (Wei *et al.*, 2010a, b).

The main contributions of this paper are as follows. We propose an efficient algorithm that will construct a data aggregation tree and a synchronous TDMA schedule for all links in the tree such that the latency of aggregating all data to the sink node is approximately minimized. An efficient distributed method is proposed to structure a collision-free schedule for data aggregation in WSNs. We prove that the latency of the aggregation schedule generated by our algorithm is at most $4\delta+15R-16$ time-slots where R is the network radius and Δ is the maximum node degree in the communication graph of the original network.

The data aggregation scheduling problems have been extensively studied in the works of Annamalai *et al.* (2003), Intanagonwiwat *et al.* (2002), Madden *et al.* (2002) and Yu *et al.* (2004) in recent years. The most related ones are as follows. Krishnamachari *et al.* (2002) proposed a randomized and distributed algorithm for aggregation in a n -node sensor network with an expected latency of $O(\log n)$. In their model, there are two assumptions. One is that each sensor node has the capability of detecting whether a collision occurs after transmitting data. Another one is that sensor nodes can adjust their transmission range without any limitation. These assumptions pose some challenging issues for hardware design and the latter assumption is almost impossible when the network scale is very large. Chen *et al.* (2005) proposed an algorithm to generate a collision-free schedule with latency bound of $(\Delta-1)R$. They also proved that the minimum data

aggregation time problem is NP-hard. Huang *et al.* (2007) proposed a scheduling algorithm with the latency bound of $23R+\Delta+18$ time-slots where R is the network radius and Δ is maximum node degree. However the interference model used is a simple primary interference model: no node can send and receive simultaneously. Under Protocol Interference Model, Yu *et al.* (2008) proposed a distributed scheduling algorithm generating collision-free schedules that have a latency bound of $24D+6\Delta+16$ time-slots where D is the network diameter. Xu *et al.* (2009) proposed a scheduling algorithm generating collision-free schedules that has a latency bound of $16R+\Delta-14$ time-slots which choose the topology center of the UDG as the root that is not practical, because the root is common node with limited energy and prone to be failure. Wan *et al.* (2009) proposed a scheduling algorithm generating collision-free schedules that has a latency bound of $15R+\Delta-4$ time-slots.

Among these algorithms mentioned above, some work on centralized aggregation scheduling that are not practical, especially when the network topology often changes in a large sensor network and other work on distributed aggregation scheduling that can need the requirement of the dynamic network properties. But data scheduling of these algorithms uses a bottom-up or up-bottom approach that don't consider nodes in different dominate nodes execute the transmission at the same time and still have high latencies. Moreover, each dominant node costs at most Δ time slots that are not accurate.

PROBLEM DEFINITION

We consider a WSN consisting of n nodes, given a graph $G = (V, E)$ representing the WSN and a base station (sink node) $s \in V$, to find a data aggregation scheduling with minimum latency. Each node is equipped with an RF transceiver that can be used to send or receive data. We consider omni-directional antennae only and a node's transmission or reception range is roughly a disk centered at that node. For simplicity we assume that all nodes other than the sink have the same transmission range (radius) r such that both node and form a communication link whenever their Euclidean distance $\|u-v\| \leq r$. We normalize their transmission radius to one unit and use a unit disk graph (UDG).

Let $X, Y \subseteq V$ and $X \cap Y = \emptyset$. If all the nodes in X transmit data simultaneously in one time-slot and all data are received by some nodes at the same time in Y with collision-free we say data are aggregated from X to Y in one time-slot. A data aggregation scheduling can be thought of as a sequence of senders $\{X_1, X_2, \dots, X_l\}$ (in which $X_i \subseteq V, \forall i$) satisfying the data aggregation property. This sequence represents that all nodes in X_1 transmit in the first time slot, followed by all nodes in X_2 transmitting in the second time slot and so on. All data

will be aggregated to one single node s which is the sink node. Then a data aggregation scheduling with latency l can be a sequence of sender sets X_1, X_2, \dots, X_l satisfying the following conditions:

- $X_i \cap X_j = \emptyset, \forall i \neq j$
- $\bigcup_i X_i = V \setminus \{s\}$
- Data are aggregated from X_i to $v \in \bigcup_{i=1}^k X_i$ at time-slot k , for all $k = 1, 2, \dots, l$ and all the data are aggregated to the sink node s in l time-slots

The number l is called data aggregation latency. This paper focuses on constructing a distributed aggregation scheduling algorithm to find a schedule in a distributed way such that is minimized. This problem is proved to be NP-hard. This paper proposes an approximate distributed algorithm with at most latency $4\delta+15R-16$ where R is the network radius and δ is the maximum node degree.

SYNCHRONOUS AGGREGATION SCHEDULING ALGORITHM

For simplicity we assume that all communications proceed in synchronous time-slots and each node can transmit/receive at most one packet of a fixed size in each time-slot. We assume that the sensors are aware of their locations. The location information can be obtained either through GPS (McNeff, 2002) or various localization techniques (Hu and Evans, 2004; Savvides *et al.*, 2001; He *et al.*, 2003; Goldenberg *et al.*, 2006).

In this study we construct a Connected Dominating Set (CDS) as data aggregation tree. The data aggregation scheduling is executed according to the CDS tree. The details of our algorithm are as follows.

Distributed aggregation tree construction: The CDS is constructed by the distributed approach to generate a spanning data aggregation tree. Assume each node knows own neighbors list that contains ID and Level. The pseudo code of a Maximal Independent Set (MIS) is as follows.

Algorithm 1: Construct a max independent set (MIS)

-
- 1 : Construct a BFS tree rooted at s (the sink node) with the radius R of the original network
 - 2 : Initially, each node colors itself WHITE
 - 3 : for each white node v do
 - 4 : If the distance between v and s is greater than l^*r and less than $(l+1)^*r$ then
 - 5 : v sets itself in the level l
 - 6 : End if
 - 7 : end for
 - 8 : the nodes in the level 0 color themselves GREY
 - 9 : for each white node v do
 - 10 : finds node u among white nodes with the max gree
 - 11 : u colors itself BLACK
 - 12 : u broadcasts a GREY message to all its one-hop neighbors in G
 - 13 : the node receives a GREY message colors itself GREY
 - 14 : end for
-

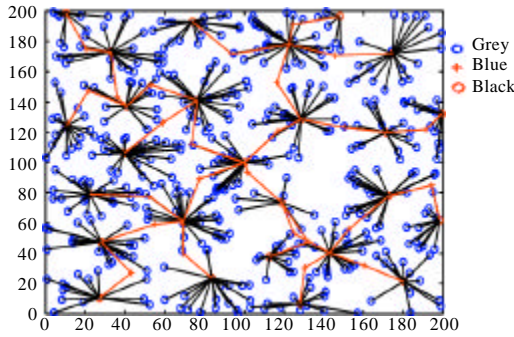


Fig. 1: Data aggregation tree

In the algorithm 1, all black nodes form a MIS set by a distributed approach. Next we will construct a connected dominating set (CDS).

Algorithm 2: Construct a CDS to form an aggregation tree level by level

```

1 : Every grey node selects a nearest black node as its dominator in its
   one-hop neighbors to join
2 : for  $i \leftarrow 1$  to  $l$  do
3 :   for all black nodes  $b$  in Level $_i$  do
4 :     Find its parent  $P_b$  in Level $_{i-2}$  by computing the first-fit path from  $b$  to
        $P_b$ 
5 :     Color the node that connects  $b$  and  $P_b$  BLUE
6 :   end for
7 : end for
    
```

According to the algorithm 1 and 2 we construct the aggregation tree just like Fig. 1. All the black nodes that are dominators form an MIS. Each blue node connects two black nodes and is lower-level black node's parent node. The grey nodes are leaves and dominates. All nodes form a CDS. For aggregation, the CDS is an aggregation tree.

Distributed parallel aggregation scheduling: According to the constructed aggregation tree to perform aggregation scheduling. Let B denote all black nodes set, $b_j \in B$ where $1 \leq j \leq |B|$. Let W denote all grey nodes, $w_i \in W$. Set the coordinate of b_j is (x_{b_j}, y_{b_j}) and the coordinate of w_i is (x_{w_i}, y_{w_i}) , where $1 \leq i \leq |W|$. We construct different virtual disks centered at every black nodes of radius one and divide every disk into four equal portions: $area_{1,1}$, $area_{1,2}$, $area_{1,3}$ and $area_{1,4}$ along the horizontal and vertical directions. The $area_{1,1}$ is formed by the nodes in the range $x_{b_j} \leq x_{w_i}$ and $y_{b_j} > y_{w_i}$ in b_j 's children nodes. The $area_{1,2}$ is formed by the nodes in the range $x_{b_j} > x_{w_i}$ and $y_{b_j} \leq y_{w_i}$ in b_j 's children nodes. The $area_{1,3}$ is formed by the nodes in the range $x_{b_j} \geq x_{w_i}$ and $y_{b_j} > y_{w_i}$ in b_j 's children nodes. The $area_{1,4}$ is formed by the nodes in the range $x_{b_j} < x_{w_i}$ and $y_{b_j} \leq y_{w_i}$ in

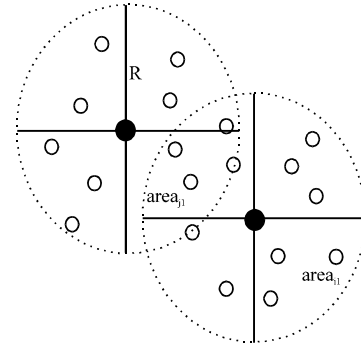


Fig. 2: The different fields constructed by the dominant nodes

b_j 's children nodes. As shown in Fig. 2, the area is how to be structured. The communications process in $area_{1,1}$ are executed at the same time in synchronous time-slots, where $\forall j, j \in |B|$. Nodes of the four fields are scheduled in clockwise order, that is to say, every node in the $area_{1,1}$ finishes transmitting the information, next the b_j receives the information of every node in the $area_{1,2}$ and so on.

We will determine the number of time slot in four equal portions in the following first-fit manner. Assume that D_i and D_j are both neighboring dominant nodes in B . $D_{i,1}$ denotes the set of nodes in the $area_{1,1}$ of D_i . The set of nodes with respect to the $area_{1,1}$ of D_j are denoted by $D_{j,1}$. Consider ID increasing order $\langle v_{1,1}, v_{1,2}, \dots, v_{1,|D_{i,1}|} \rangle$ of $D_{i,1}$ and $\langle v_{j,1}, v_{j,2}, \dots, v_{j,|D_{j,1}|} \rangle$ of $D_{j,1}$. Initially, $U = \{v_{1,1}\}$ and $v = \emptyset$. For $x = 2$ up to $|D_{i,1}|$ for $y = 1$ up to $|D_{j,1}|$, if $v_j \notin V$ and $v_{j,y}$ is first not adjacent to v_{ix} , add $v_{j,y}$ to V . Because the matching nodes in U and V are pairwise conflict-free, $|U|$ is equal to $|V|$ and they use $|U|$ time slot to complete the communication. The nodes in $D_{i,1}/U$ and $D_{j,1}/V$ are conflicting and need separate time slot. According to the algorithm, we can achieve the number of time slot is equal to $\delta_{i,1} = |D_{i,1}/U| + |D_{j,1}/V| + |U|$.

The b_j separately gets the number $\delta_{i,1}, \delta_{i,2}, \delta_{i,3}$ and $\delta_{i,4}$ of nodes in different area. All nodes in B broadcasts $\delta_{i,1}, \delta_{i,2}, \delta_{i,3}$ and $\delta_{i,4}$ messages in WSNs and get the max values of $\delta_{i,1}, \delta_{i,2}, \delta_{i,3}$ and $\delta_{i,4}$ that separately denote $\delta_1, \delta_2, \delta_3$ and δ_4 . The nodes in the $area_{k,k}$ are assigned corresponding time slot t deferred to the conflict-free in δ_k and are sleep in the rest of the time. For example, the first time slot is assigned in Fig. 3. Every dominator has the same scheduling period $\delta_1 + \delta_2 + \delta_3 + \delta_4$. The max value of the scheduling period is $4\delta - 1$. The dominates in all nodes of B periodically send data to b_j if and only if they are in the corresponding time slot and are in sleep mode in the rest of time. In $\delta_1 - \delta_{i,1}, \delta_2 - \delta_{i,2}, \delta_3 - \delta_{i,3}$ and $\delta_4 - \delta_{i,4}$ interval, the b_j node is in sleep state.

Let C be the set of connectors. The scheduling in C is as the following. Observe that every dominator,

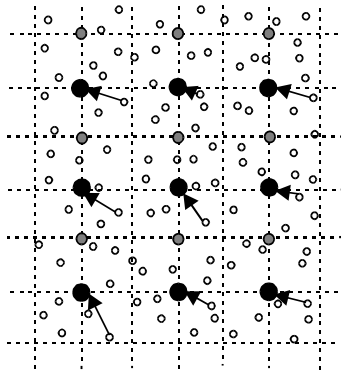


Fig. 3: Scheduling in the first time slot

except the root of the data aggregation tree, connects to at least one dominator in the upper level within 2-hops. We now determine the max number of time slots that a dominator node will use to connect to all dominators within 2 hops. Consider any dominator u , let $D_2(u)$ be the set of dominators within 2-hops of u in the original communication network G . Based on a technique lemma implied from lemmas proved, u requires at most 12 connectors to connect $D_2(u)$. We use $C(u)$ to denote the set of connectors used to connect all dominators in $D_2(u)$. According to Wegner Theorem, there can be at most 5 dominant nodes in a unit disc centered at any node u in $C(u)$. Each connector only receives the data from lower-level dominant nodes, so u can receive the data from all neighboring dominators $D_2(u)$ in at most 15 time slots. All dominators within 2 hops execute parallel scheduling by connectors.

Theorem 1: By using distributed parallel aggregation scheduling, the sink can receive all the aggregated data in at most $4\delta + 15R - 16$ time-slots.

Proof: Every dominator can aggregate its neighboring dominates' data in at most $4\Delta - 1$ time-slots within corresponding time-slot by parallel manner. Then connectors ensure that every dominator's data can be aggregated upwards the root finally. Every dominator u including the root of data aggregation tree can collect aggregated data from all dominators in $D_2(u)$ within at most 15 time-slots. Since there is no dominator in level R , after at most $15(R - 1)$ time-slots, every dominator's data can be aggregated to the root. Therefore within time-slots, all the data can be aggregated to the sink node.

SIMULATION RESULTS

In our simulation we randomly deploy sensors into a region of 200×200 m. All sensors have the same transmission radius. We compare the performance of our algorithm SAS with the algorithm for aggregation scheduling DAS.

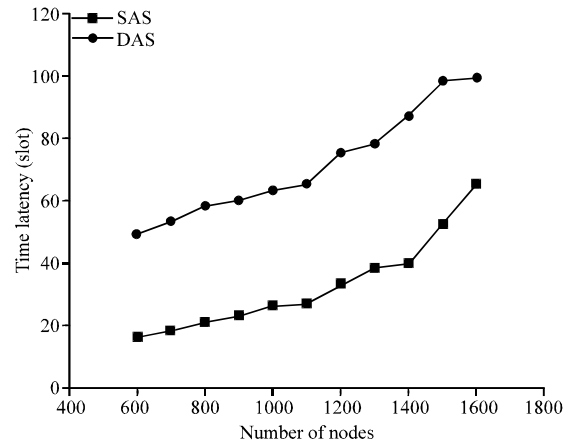


Fig. 4: Latency with different number of nodes

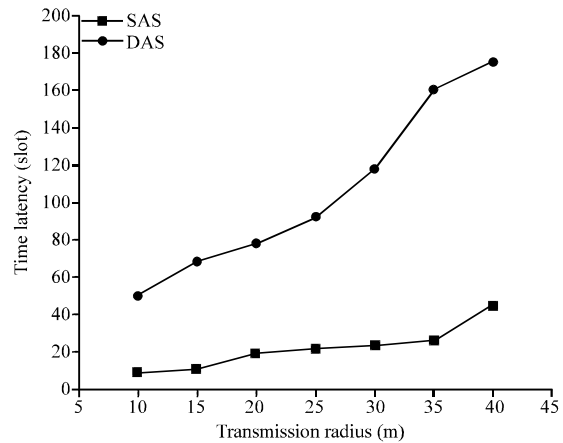


Fig. 5: Latency with different transmission radius

We use matlab to implement SAS and DAS algorithm. In Fig. 4, the transmission radius of each sensor is fixed to 25 m. The figure shows the number of the time slots needed to aggregate data from the leaves to the sink by running the two algorithms while the number of nodes increases. Fig. 5 compares the number of slots to aggregate data using the two algorithms when the transmission radius varies. Here the number of nodes is fixed to 1600. It can be seen from the Fig. 4 and 5 that SAS outperforms DAS algorithm with much lower latencies. From the figures, SAS's latencies are 1.52 to 3.06 times smaller than DAS's latencies and with the increment of the number of nodes and the transmission radius, the improvement of the latency will be larger. The reason is that we choose the max degree node as a dominant node which causes the nodes to uniformly distribute in different levels, thereby, reduce the latency.

Next we compare the running time of SAS and DAS in Fig. 6 when the number of sensor nodes and the

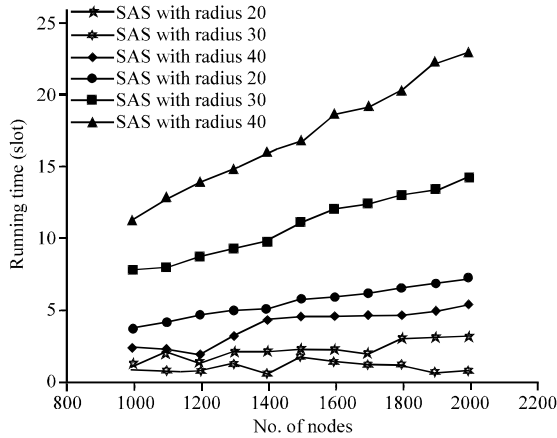


Fig. 6: Running time with different number of nodes

transmission radius vary. The running time refers to the time required to generate the aggregation scheduling. We set 1/20 of a second for each time slot when running two algorithms which is greater than the time required to transmit one packet. Though the time complexity is $O(n)$ in DAS we can see that the running time is rather small. We can see that the running time of SAS is 2 to 8 sec on average. For instance, for 2000 nodes with transmission radius of 30 m, the running time of DAS is 14.6 sec and the running time of SAS is only 3.2 sec. The reason is that many nodes can schedule simultaneously in SAS and the occasion where only one node can schedule at each time maybe happens in DAS. As the number of the nodes or the transmission radius increases, the average size of the nodes' competitor sets also increases and thus each node has to compete with more competitors that cost more time.

CONCLUSIONS

In this study we make a study of the data aggregation problem and its latency. We use the techniques of maximal independent sets and connected dominating set to construct the aggregation tree. We propose a distributed synchronous aggregation scheduling algorithm. The latency of the aggregation schedule generated by our algorithm is at most $4\delta+15R-16$ time-slots. Analysis and simulation results show that the SAS algorithm has smaller latency and better performance by comparing with the DAS algorithm. A distributed asynchronous aggregation scheduling algorithm will be proposed which makes the algorithm suitable for networks with dynamic topology changes in our future work.

ACKNOWLEDGMENTS

We would like to thank Yanchao Zhao for helpful discussions and insightful comments. He reviewed the

draft of the paper and made further modifications that improved the quality of the study. Yanchang Liu devoted herself to designing some graphs and other art works in this study. We also thank the anonymous reviewers for their insightful and valuable hints.

REFERENCES

Annamalai, V., S. Gupta and L. Schwiebert, 2003. On tree-based convergecasting in wireless sensor networks. IEEE WCNC., 3: 1942-1947.

Chen, X., X. Hu and J. Zhu, 2005. Minimum data aggregation time problem in wireless sensor networks. Proceedings of the 1st International Conference on Mobile Ad-hoc and Sensor Network-MSN (MASN'05), Springer-Verlag, pp: 133-142.

Goldenberg, D.K., P. Bihler, Y.R. Yang, M. Cao, J. Fang, A.S. Morse and B.D.O. Anderson, 2006. Localization in sparse networks using sweeps. Proceedings of the ACM Mobicom, December 2006, USA., pp: 110-121.

Guo, L., B. Wang, Z. Liu and W. Wang, 2010. An energy equilibrium routing algorithm based on cluster-head prediction for wireless sensor networks. Inform. Technol. J., 9: 1403-1408.

He, T., C. Huang, B.M. Blum, J.A. Stankovic and T. Abdelzaher, 2003. Range-free localization schemes for large scale sensor networks. Proceedings of the 9th ACM International Conference on Mobile Computing and Networking, Sept. 14-19, San Diego, CA, USA., pp: 81-95.

Hu, L. and D. Evans, 2004. Localization for mobile sensor networks. Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, Sept. 26-Oct. 1, Philadelphia, USA., pp: 45-57.

Huang, S.C.H., P.J. Wan, C.T. Vu, Y. Li and F. Yao, 2007. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. Proceedings of the 26th International Conference on Computer Communications, May 6-12, IEEE., pp: 366-372.

Idris, M.Y.I., E.M. Tamil, N.M. Noor, Z. Razak and K.W. Fong, 2009. Parking guidance system utilizing wireless sensor network and ultrasonic sensor. Inform. Technol. J., 8: 138-146.

Intanagonwiwat, C., D. Estrin, R. Govindan and J. Heidemann, 2002. Impact of network density on data aggregation in wireless sensor networks. Proceedings of the 22nd International Conference on Distributed Computing Systems (DCS'02), IEEE., pp: 457-457.

- Kesselman, A. and D. Kowalski, 2005. Fast distributed algorithm for convergecast in ad hoc geometric radio networks. Proceedings of the 2nd Annual Conference on Wireless On-Demand Network Systems and Services, Jan. 19-21 IEEE., pp: 119-124.
- Krishnamachari, B., D. Estrin and S. Wicker, 2002. The impact of data aggregation in wireless sensor networks. The 22nd International Conference on Distributed Computing Systems Workshops, July 02-05, IEEE Computer Society, Washington DC., USA., PP: 575-578.
- Liu, Z., B. Wang and L. Guo, 2010a. A survey on connected dominating set construction algorithm for wireless sensor networks. *Inform. Technol. J.*, 9: 1081-1092.
- Liu, Z., B. Wang and Q. Tang, 2010b. Approximation two independent sets based connected dominating set construction algorithm for wireless sensor networks. *Inform. Technol. J.*, 9: 864-876.
- Madden, S., M.J. Franklin, J. Hellerstein and Wei Hong, 2002. Tag: A tiny aggregation service for ad-hoc sensor networks. Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Dec. 9-11, Boston, Massachusetts, pp: 131-146.
- McNeff, J.G., 2002. The global positioning system. *IEEE Trans. Microwave Theory Techniques*, 50: 645-652.
- Savvides, A., C.C. Han and M.B. Srivastava, 2001. Dynamic fine-grained location in ad hoc networks of sensors. Proceedings of the ACM Mobicom, July 2001, USA., pp: 166-179.
- Wan, P.J., C.H. Huang, L. Wang, Z.Y. Wan and X. Jia, 2009. Minimum-latency aggregation scheduling in multihop wireless networks. Proceedings of the 10th International Symposium on Mobile Ad-Hoc Networking and Computing, May 18-21, ACM., pp: 185-193.
- Wei, W., A. Gao, B. Zhou and Y. Mei, 2010a. Scheduling adjustment of mac protocols on cross layer for sensornets. *Inform. Technol. J.*, 9: 1196-1201.
- Wei, W., B. Zhou, A. Gao and Y. Mei, 2010b. A new approximation to information fields in sensor nets. *Inform. Technol. J.*, 9: 1415-1420.
- Xu, X.H., S.G. Wang, X.F. Mao, S.J. Tang and X.Y. Li, 2009. An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks. Proceedings of the 2nd International Workshop on Foundations of Wireless ad-hoc and Sensor Networking and Computing, May 18, ACM., pp: 47-56.
- Yu, B., J. Li and Y. Li, 2008. Distributed data aggregation scheduling in wireless sensor networks. Proceedings of the IEEE INFOCOM, April 19-25, IEEE., pp: 2159-2167.
- Yu, Y., B. Krishnamachari and V. Prasanna, 2004. Energy-latency tradeoffs for data gathering in wireless sensor networks. Proceedings of the 23rd Annual Joint Conference of the Computer and Communications, March 7-11, IEEE., pp: 1-12.