2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover

A Workload-Aware Link Scheduling for Heterogeneous Wireless Sensor Networks

Bo Zeng*, Yabo Dong†, Zhidan Liu‡, Dongming Lu§
College of Computer Science and Technology, Zhejiang University, China

*Email: wbzeng.hn@gmail.com †Email: dongyb@zju.edu.cn ‡Email: danielliu@zju.edu.cn §Email: ldm@zju.edu.cn

***Abstract*—We focus on the problem of links scheduling in heterogeneous wireless sensor networks. Links scheduling becomes more difficult due to the diversity of workload among sensor nodes. An improper links scheduling strategy will lead to high end-to-end delay and then diminish the data availability. We assume the energy cost of state transition cannot be ignored, especially for wireless multimedia sensor node which equips with camera and higher performance processor compared to scalar sensor node. In this paper, we use the conflict graph to model the conflicts among nodes and then propose a workload-aware heuristic scheduling algorithm. In the scheduling, each node can obtain consecutive time-slots to transmit all data at a schedule transmission, and the number of time-slots is calculated based on node's workload. Hence, it is possible for our scheduling to reduce the end-to-end delay and save energy consumed by node's state-switching in heterogeneous wireless sensor networks. In additional, a metric is proposed to measure the frequency of node's state-switching. The numerical results show that our scheduling can reduce the end-to-end delay and meanwhile improve energy efficiency by reducing the energy cost of state-switches.**

***Keywords*-heterogeneous, centralized, conflict graph, workload-aware**

I. Introduction

A wireless sensor network (WSN) consists of a large number of low cost, low power sensor nodes to perform monitoring task. Sensor nodes may generate different amounts of data (e.g., they are equipped with different types sensors) and then send data to the sink node periodically. This type of data collection can be formulated as heterogeneous data collection, where each sensor node generates different amounts of data in one operation.

Time Division Media Access (TDMA) [1] usually is used to eliminate conflict among sensor nodes and achieve high wireless capacity. With TDMA protocol, links are allowed to use whole link bandwidth when they are scheduled to access the wireless channel. Scheduling TDMA is a valid method to achieve the optimal throughput of a WSN. Many previous literatures formulated scheduling TDMA as a graph coloring problem. Based on the communication graph of a WSN, a conflict graph consists of communication links as vertices and conflicts of communication links as arcs. The vertices connected with arc cannot transmit simultaneously due to mutual interference in conflict graph. Therefore, the result of coloring conflict graph is a conflict-free link schedule. Different graph coloring algorithms are proposed to obtain a conflict-free link schedule [2].

However, most of these algorithms assume that each sensor node generates exactly the same amount of data at the same rate, and use uniform slots in scheduling, such as one time-slot for a assignment. In many applications of heterogeneous WSN, uniform slots allocation of existing scheduling algorithms cannot adapt to highly variable data traffic on sensor nodes because discontinuous time-slots are allocated to each node if a node has a large amount of data needed to be transmission, resulting in a great degradation in performance of network, e.g., latency. In addition, each node needs frequently state switching which also causes large energy cost for sensor node, e.g., wireless multimedia sensor node.

This paper presents a workload-aware heuristic scheduling algorithm for heterogeneous data collection in WSNs. Taking the workload of sensor nodes into account, our scheduling employs a workload-aware time-slots allocation mechanism to minimize influence of heterogeneous workload in WSNs while guaranteeing low delay. In addition, our scheduling also considers the energy $E_{A,B}$ consumes by transiting from state A to state B for a sensor and other control units. A metric is proposed to measure the frequency of node's state-switching, and it also can be use to denote the consecutiveness of node's schedule.

The rest of the paper is organized as follows: Section III describes the heterogeneous network and physical interference model. Section IV shows the detail of workload-aware link scheduling algorithm and section V analyzes the performance of scheduling algorithm. Section VI collects some conclusions.

II. Related work

The main task in determining a TDMA schedule is to find a conflict-free schedule for all links in the network. A related problem is to find the interference information on a set of links, which shows these links whether or not can active simultaneously - they can be scheduled, while this problem has been proven to be NP-hard [2]. Many scheduling mechanisms based on TDMA adopt the protocol interference model [3], [4], [5]. Compared to protocols interference model, physical interference model has better accuracy on the degree of interference among nodes [6]. Therefore, in

978-0-7695-4810-4/12 $26.00 © 2012 IEEE
DOI 10.1109/CyberC.2012.66

order to improve the effective of links scheduling, we use the latter to represent the relationship among links.

Jain et al. [7], formulate the problem of links scheduling under protocol and physical interference model as an LP problem. Unfortunately, the solution requires computing all possible sets of conflict-free transmissions to achieve the optimal link schedule, which needs exponential time. In this paper, the authors provide theoretical performance base on MATLAB and CPLEX, and propose a column generation approach which also requires complexity computation and exponential time.

In [8], Wang et al., study links scheduling problem under the RTS/CTS interference model and protocol interference model. Although they present both centralized and distributed algorithms in the paper, the results of links scheduling may be not optimal solution under some networks, such as tree-based network. Similarly, there are some literatures have considered physical interference for links scheduling in wireless networks [9], [10], [11], [12].

In [13], Mao et al., proposes an optimization framework where genetic algorithm and particle swarm optimization algorithm are hybridized to enhance the searching ability. However their algorithm does not provide any guarantees on the performance. Djukick and Valaee[1] design an algorithm that finds the transmission order with the minimum delay on overlay tree topology. Combined it with a modified Bellman-Ford algorithm, the author wanted to find minimum delay schedule in polynomial time. However, the algorithm has possibility of failure in searching the minimum number of slots required to schedule all links. In [14], the author proposed a heuristic algorithm that schedules as many independent set of links as possible to increase the degree of parallel transmission. In [2], Ergen and Varaiya provided another heuristic algorithm based on scheduling the levels in the routing tree and whose performance depends on the distribution of the nodes across the levels in routing tree.

As the previous scheduling algorithms assign fixed number of time-slots to a set of conflict-free links at a schedule transmission, these algorithms unable to adapt to the heterogeneous workload in wireless heterogeneous sensor networks, and lead to large end-to-end delay and waste energy. Hence, the need of an efficient scheduling which provides adaptability for workload motivates us to design our scheduling algorithm.

III. Network Model and Assumption

We consider a network that is composed of a single access point (AP) and several sensor nodes. Each sensor node has different initial workload. Links among the nodes are assumed to be bidirectional, so sensor node can receive the link-layer acknowledgement when data is delivered successfully to destination. Besides, we assume all sensor nodes transmit at the same power.

A physical interference model is used in our scheduling. According to the model shown in [15], successful reception of a packet that is sent from node u to node v depends on the cumulative signal-to-interference and noise ratio (SINR) at v. To be specific, $P_v(i)$ denotes the received power at node v, and the signal transmitted by node i, so a packet along with link (u,v) is correctly received if and only if:

$$\frac{P_v(u)}{N_0 + \sum_{i \in (V'/u)} P_v(i)} \geq \beta$$

where N_0 is the background noise, $V^{'}$ denotes the subset of nodes in V that are transmitting simultaneously, and β is a constant which depends on the modulation scheme for channel, data rate, etc. Note that we do not assume any specific radio propagation model in our scheduling.

We represent the network with the physical communication graph G = (V,E), where V is the set of sensor nodes, $N = |V|$ is the number of vertices in G. The directed edge (u,v) $\in$ E exists if and only if a link between node u and v exists in network. Each edge e in E is labeled with w_e which is equivalent to the initial workload of source end of e. The graph G forms a data collection tree T for a WSN at last, and AP is the destination of all data.

To build interference graph $G_I = (V_I, E_I)$ according to the interference model is necessary for links scheduling in network. An edge e = (u,v) $\subseteq$ I is labeled with weight I_e which represents the signal strength at node v when node u is transmitting and is able to use dBm to express. Note that I is the subset of edges V x V. Interference graph was also referred in [10]. We assume interference graph G_I is known in this paper. However, it is worth pointing out that some interference detection approaches have been proposed for wireless sensor networks [16]. These approaches can be effectively integrated with our scheduling algorithm.

The conflict graph corresponding to G = (V,E) and $G_I = (V, I)$ is called $G_{CI} = (V_{CI}, E_{CI})$ which is constructed by having a node for each link in G, and adding the undirected edge(e_1, e_2) if link e_1 and e_2 conflict or interference with each other (according to G_I described above). G_{CI} is defined as follows:

- Transferring links into vertices: for each link in G, we builds a quaternion to represent it. Each quaternion includes vertex id, source sensor node, destination, and initial workload of link, we use (ID, S, D, W) to describe the vertex.
- Building conflict edges: if ((i,j) $\cup$ (j,k)) $\in$ E, vertex x $(ID_{ij}, S_{ij}, D_{ij}, W_{ij})$ and vertex y $(ID_{jk}, S_{jk}, D_{jk}, W_{jk})$ are connected with an edge. As a parent node and a child node cannot transmit at the same time, that is (x,y) $\in E_{CI}$.
- Building interference edges: if (i,j) $\in$ I or (i,j) $\in$ E and c_j is a child of j in G, (x,y) $\in E_{CI}$, note that i or c_j is a source or destination of vertex x and it has the same

meaning for vertex y, and the edge is an interference arc. Because i and j interference each other, while i is transmitting, the data transmission from child c_j to j may fail because j would hear from both j and c_j. The physical interference model can be used to determine whether x and y can transmit at the same time.

Given a links set M=$e_1, e_2, \cdots, e_n$, M is a feasible schedule if and only if the corresponding nodes in G_{CI} construct an independent set, in other words, M is a matching in G. Hence, a necessary condition for links set M to be feasible is that M is a match on the graph G.

In this paper, time is divided into time-slots. A time-slot is long enough to transmit one packet and an acknowledgement between two sensor nodes. The length of scheduling frame is the amount of time-slots used by sensor nodes to transmit all data to AP.

Let G to be the tree rooted at AP, and G_{CI} is a conflict graph under physical interference model. A schedule S composed of N slots $t_1, t_2, \cdots, t_N$ is feasible for G if and only if the following conditions are satisfied:

-the selected set of vertices in G_{CI} form an independent set in each time slot $t_i \subseteq$ N

-each link (i,j) in G is scheduled for at least W_{ij} slots

-W_{ij} = $W_s^{'}$ + w_{ij}, where $W_s^{'}$ is the total workload of subtree s which roots at i in G and w_{ij} is initial workload of link (i,j).

IV. Workload-aware greedy scheduling algorithm (WAGS)

A. The detail of algorithm

Our scheduling is a workload adaptive greedy algorithm. At the beginning of scheduling, the nodes are ordered according to ascending order of degree since low-degree vertices have less conflict constraints and so are more likely to obtain largely independent set of nodes for each time-slot. In this paper, each time-slot represents a schedule transmission. The scheduling then allocates time-slots to nodes based on the workload of nodes in that order. After determining the first node for a time-slot, additional nodes are added if the node set for time-slot is conflict-free. Note that the member of node set for each time-slot is dynamical because nodes have different initial workload in heterogeneous WSNs. The advantage of our scheduling is that node can transmit all data to next-hop once it is scheduled, and this can remarkably reduce the frequency of state transition. Energy is saved when the amount of energy wasted on state transition cannot be neglected, such as wireless multimedia sensor node which equips with camera, meanwhile, the end-to-end delay is reduced. The scheduling algorithm is reported in Algorithm 1. Table I shows some notations and terminologies used in Algorithm 1.

When WAGS initializes completely, the function $UPDATE\_DEGREE(set_{os})$ (line 16) is called to update the degree of candidate vertices in G_{CI}, and then the candidate vertices are scheduled in ascending order of degree. Note that G_{CI} consists of conflict edges and

Table I
Notations and terminologies

Symbol	Meaning
set_p	set of vertices that prior to schedule in current time-slot
set_s	set of vertices already scheduled in current time-slot
set_{os}	set of candidate vertices can be scheduled in current time-slot
set_{last}	set of vertices scheduled at last time-slot
W_i	the workload of vertice i
T	set of time-slots

Algorithm 1 Workload-aware Greedy Scheduling Algorithm

Require: a communication graph G and the conflict graph G_{CI}

Ensure: a feasible schedule S for G under physical interference model

1: set $S = \phi$
2: ordering vertices according to ascending order of degree using depth-first-search (DFS)
3: **while** (at least one packet has not reached AP) **do**
4: set_s = NULL, set_p = NULL
5: **while** $W_i > 0$ **do**
6: $set_{os} \leftarrow i$
7: **end while**
8: **while** $W_j > 0$ and j $\in set_{last}$ **do**
9: $set_p \leftarrow j$
10: $set_{os} \leftarrow set_{os} - j$
11: **end while**
12: **if** set_p != NULL **then**
13: set_s = set_p
14: **end if**
15: **if** $set_{os}! = NULL$ **then**
16: $UPDATE\_DEGREE(set_{os})$
17: **end if**
18: **for** (set_{os} != NULL) **do**
19: CHECK:
20: **if** $\kappa \in set_{os}$ and κ has the minimum degree **then**
21: **if** $(\kappa, \psi) \notin ECI$ and $(\forall \psi \in set_s)$ **then**
22: **for** each vertice k $\in set_s$ **do**
23: **if** $SINR(\kappa, k) \leq \beta$ **then**
24: goto CHECK
25: **end if**
26: **end for**
27: $set_s \leftarrow \kappa$, $set_{os} = set_{os} - \kappa$
28: **end if**
29: **end if**
30: **end for**
31: set_{last} = set_s, $T = T + 1$, $S \leftarrow set_s$
32: update workload of vertices
33: **end while**
34: return S, and schedule length $|S|$

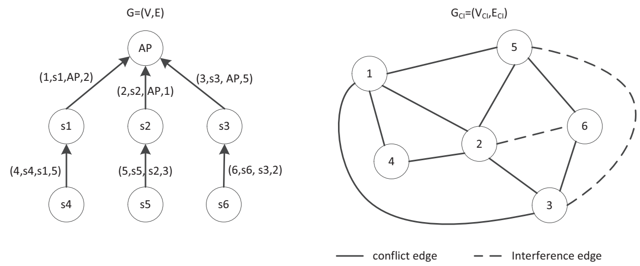

Figure 1. Communication graph $G = (V, E)$ and Conflict graph G_{CI}=(V_{CI}, E_{CI})

interference edges. Vertices cannot work simultaneously when there are conflict edges, however, if they only have interference edge, SINR must be checked by calling function $SINR(e_1, e_2)$ (line 23), and if $SINR(e_1, e_2) \geq \beta$ is true, vertice will be included into simultaneous set which belongs to the current time-slot, or else it should be give up in current time slot. When set_{os} is empty, there is no more vertex can be included into current time-slot. At last, if some packets still cannot reach to AP, additional time-slots are needed to transmit these packets.

B. The degree of spread of heterogeneous workload (χ)

In WAGS, the degree of spread of heterogeneous workload (χ) is used to describe the difference of initial workload of sensor nodes. The value of χ is larger, the difference of initial workload jumps over bigger. Considering a heterogeneous wireless sensor network which has n sensor nodes, and the initial workload of node i is W_i. WAGS uses equation (1) to calculate χ.

$$\chi = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(W_i - \bar{W})^2} \quad (1)$$

$$\bar{W} = \frac{1}{n}\sum_{i=1}^{n} W_i \quad (2)$$

C. The dispersion of schedule (φ)

A large number of discrete time-slots can lead to frequent state transition, and hence waste considerable energy such as wireless multimedia sensor nodes. However, assigning consecutive time-slots to sensor node if possible can reduce energy cost caused by node's state transition. In WAGS, the dispersion of schedule (φ) is used to measure the consecutiveness of data transmission. The value of φ is larger, the discrete degree of schedule is higher and the number of inconsecutive time-slot is larger so that each sensor node needs more frequent state transition. WAGS uses the same equation as χ to calculate φ, the only difference between χ and φ is that W_i is replaced by the number of inconsecutive time-slots included in the schedule of node i.

D. Example analysis

We give an example of a typical tree topology of heterogeneous wireless sensor networks, and χ=1.5. $G = (V, E)$ and $G_{CI} = (V_{CI}, E_{CI})$ are shown in Fig.1. There are two interference edges: (2,6) and (3,5) in G_{CI}. In the scheduling example, we assume that two interference edges cause serious interference between these vertices and cannot be scheduled in the same time-slot. The scheduling results are shown in Fig.2. Table II shows the performance data of example.

Table II
THE PERFORMANCE RESULTS OF EXAMPLE

Algorithm	Total workload	χ	φ	State transition	Delay
node-based	18	1.5	1.2	3.0	37
WAGS	18	1.5	1.0	1.0	20

As we can see from Fig.2, WAGS has the same scheduling length when compared with Node-based scheduling. However, as WAGS allocates consecutive time slots to vertices using workload-aware method, WAGS can remarkably reduce the state transition of sensor nodes, and it is clearly shown in Table II. We consider node s1, its state transition reach up to 5 times under the node-based scheduling, however, it can decrease to only 1 times according to WAGS. For some sensor nodes which has high circuit power consumption, WAGS owns obvious advantage on energy efficiency, meanwhile it can reach the similar performance on schedule. Besides, WAGS does not need coloring conflict graph which can increase the complexity of scheduling, and this is different from node-based scheduling.

V. PERFORMANCE EVALUATION

We conduct extensive simulations to compare the performance of WAGS with some methods in the literature. All algorithms are implemented in Matlab. The size of network topology is 300m x 300m. We take the arithmetic mean of 50 runs. We assume a range based interference model and regulate that every sensor's communication range is 1 hop (about 50m) and its interference range is 2 hop. The first set compares the frequency of state transition. Reducing frequency of state transition can saved energy when the energy cost of state transition cannot be ignored, especially when referring to the wireless multimedia sensor networks. Meanwhile, the dispersion of schedule is also evaluated in our simulations. The second set of results are related to end-to-end delay. In all simulations, we compare WAGS with the node-based scheduling algorithm provided in [2]. The node-based scheduling algorithm assigns time-slots to each node according to the coloring of conflict graph. Each time only one time-slot is assigned. Besides node-based scheduling algorithm, we also implement another two greedy scheduling algorithm referred to maximum- and minimum-first scheduling algorithm (Max-prior and Min-prior). Two

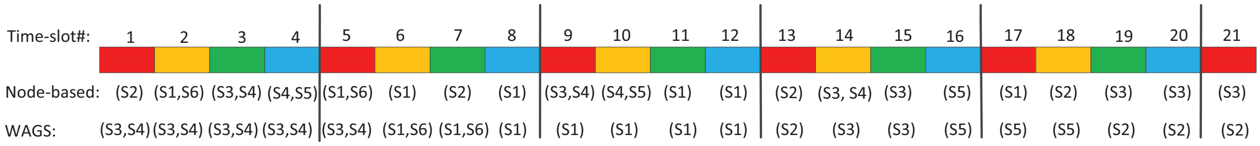

Figure 2. The scheduling results of example

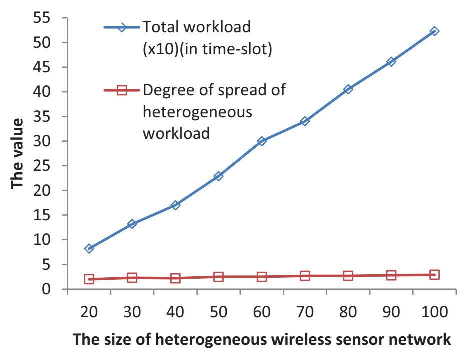

Figure 3. The total initial workload of network and the degree of spread of heterogeneous workload (χ)

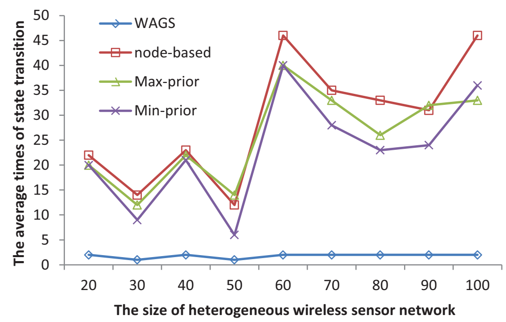

Figure 4. The average frequency of state transition under different sensor node densities

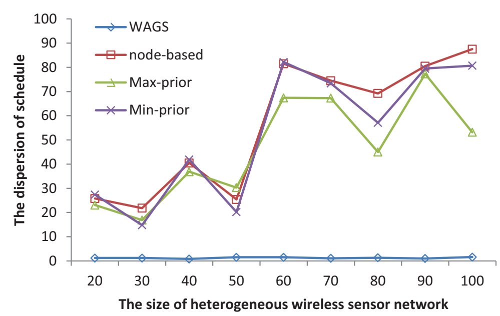

Figure 5. The degree of spread of time-slots under different sensor node densities

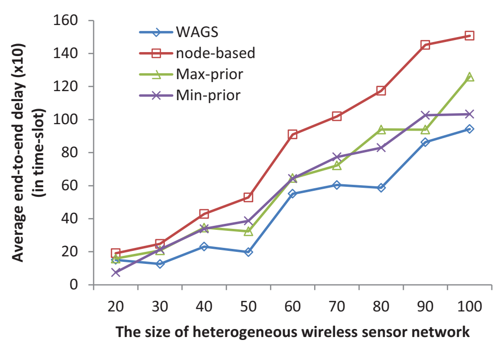

Figure 6. The average end-to-end delay under different sensor node densities

scheduling assign time-slots to each sensor node according to the descending and ascending order of degree respectively, and they have not use the workload-aware mechanism.

A. *Impact of number of nodes*

In this simulation, we study the relationship among the number of sensor nodes, the end-to-end delay and the frequency of state transition. We vary the number of sensor nodes from 20 to 100 in the network. The workload of each sensor node is w time-slots which lies in the interval [1,10]. Only one packet is transmitted in a time-slot. The total workload of heterogeneous network and corresponding χ are shown in Fig.3. The total workload increases remarkably when the number of nodes increases, and χ changes among the interval [2.0, 3.0].

We found that under WAGS, sensor nodes need to wake up less times to transmit or receive , while many sensor nodes need to wake up numerous times under the schedule by node-based method. This is because the total traffic load in data collection application increases when the number of nodes increases so that it will increase the dispersion of schedule. Figure 4 and 5 also show that WAGS has the lowest state-switches and the smallest dispersion of schedule when compared with node-based method, max-prior, or min-prior. This is because we assign the consecutive time-slots to each node using workload-aware method, meanwhile only one time-slot can be assigned for each coloring using node-based method so that each sensor node needs more than one discrete time-slot to transmit traffic load.

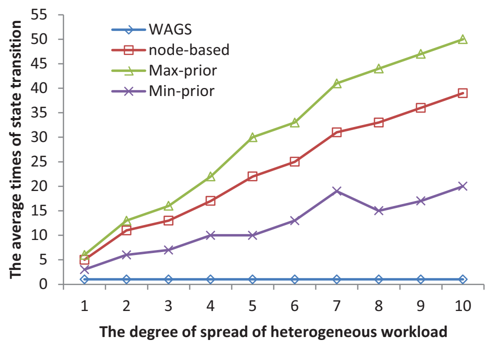

Figure 7. The average times of state transition with different χ, N=50

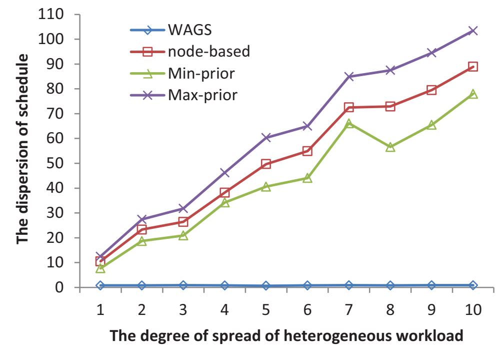

Figure 8. The dispersion of schedule with different χ, N=50

Fig.6 shows the end-to-end delay under different node densities. We found that the average end-to-end delay increases along with the number of nodes increases. This is because adding sensor nodes lengthen the scheduling length and thus increase the queue delay of each packet. We observe that WAGS has smaller delay than node-based method.

B. *Impact of heterogeneous workload*

We then study the impact of the degree of spread of heterogeneous workload on network performances. We vary the value of χ from 1 to 10 in a sensor network with 50 nodes.

Figure 7 shows that the trend of state-switches increases when χ increases, except WAGS. This is because the workload increases when χ increases and sensor nodes need more discrete time-slots to transmit data. Under the WAGS, each node can use finite times of state transition to accomplish the data transmission by workload-aware method, and the times of state-switches depends on the position of node in data collection tree which is formed by the sensor network. As a result, the average time of state transition almost unchanged for WAGS under different χ. The continuity of schedule is shown in figure 8. WAGS also has the best performance when compared with node-based method, max-prior, and min-prior. From figure 9, we observe that the delay of WAGS is far less than node-based, and the delay of node-based scheduling increases obviously with χ increases. The max- and min-prior methods have similar delay performance compared with WAGS. This is because some sensor nodes may have consecutive time-slots so that they will be scheduled sequentially.

C. *Energy efficiency*

In this paper, node's energy can be consumed by state switching from one to another, such as from sleeping to transmitting. Table III summarizes some symbols for energy computation. We calculate the number of saved energy using equation (3). As there are many types of sensor nodes which have different energy cost for state transition, it is very hard to measure the effect of energy saving under WAGS. However, according to figure 4 and 7, we have $E_s > 0$ which means WAGS can save energy because of less frequency of state-transition.

$$E_s = N(\gamma P_H + (1-\gamma)P_S)(F_{node} - F_{WAGS}) \quad (3)$$

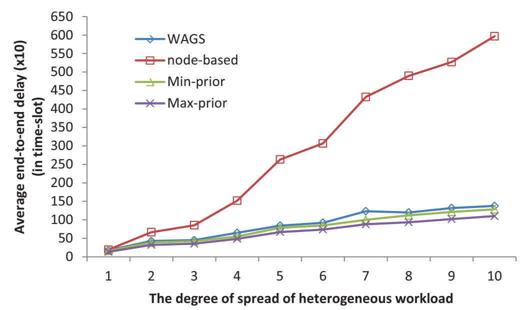

Figure 9. The average end-to-end delay with different χ, N=50

Table III
SYMBOLS FOR ENERGY COMPUTATION

Symbol	Meaning	Value
P_H	Energy consumption in state-switch of non-scalar node	α
P_S	Energy consumption in state-switch of scalar node	β
γ	The ratio of non-scalar sensor nodes to the size of network (e.g., N)	[0,1]
F_{WAGS}	The average times of state switching under WAGS	
F_{node}	The average times of state switching under node-based method	

VI. CONCLUSION

Wireless sensor networks are characterized by limited hardware resource, bandwidth and variable channel capacity. In this paper, we propose workload-aware greedy scheduling algorithm (WAGS) for heterogeneous WSNs. A physical interference model is used to describe the interference among sensor nodes. We use conflict graph which includes interference edges as the basis of adaptive greedy scheduling.

Under the workload-aware greedy scheduling, consecutive time-slots are assigned to each sensor node, and this can remarkably reduce the frequency of state transition which can cause energy consumption for some complicated sensor node, such as wireless multimedia sensor node. In addition, the degree of spread of initial workload is considered in WAGS. The numerical results show that WAGS is energy efficient and has lower end-to-end delay than node-based scheduling.

REFERENCES

[1] V. S. Djukic P., "Delay aware link scheduling for multi-hop tdma wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, 2009.

[2] S. C. Ergen and P. Varaiya, "Tdma scheduling algorithms for wireless sensor networks," *Wirel. Netw.*, vol. 16, pp. 985–997, 2010.

[3] T. Moscibroda, Y. Oswald, and R. Wattenhofer, "How optimal are wireless scheduling protocols?" in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1433 –1441.

[4] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," *Computer Networks*, vol. 52, pp. 610 – 629, 2008.

[5] C.-Y. Hong, A.-C. Pang, and P.-C. Hsiu, "Approximation algorithms for a link scheduling problem in wireless relay networks with qos guarantee," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1732–1748, 2010.

[6] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 141–154.

[7] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Networks*, vol. 11, pp. 471–487, 2005.

[8] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware tdma link scheduling for static wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, ser. MobiCom '06. New York, NY, USA: ACM, 2006, pp. 262–273.

[9] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric sinr," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '07. New York, NY, USA: ACM, 2007, pp. 100–109.

[10] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, ser. MobiCom '06. ACM, 2006, pp. 2–13.

[11] S. Gomez, O. Gras, and V. Friderikos, "Fast randomized stdma link scheduling," in *Mobile Lightweight Wireless Systems*. Springer Berlin Heidelberg, 2009, vol. 13, pp. 15–24.

[12] F. V. Papadaki K., "Interference aware routing for minimum frame length schedules in wireless mesh networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2008, 2008.

[13] W. X. Mao J., Wu Z., "A tdma scheduling scheme for many-to-one communications in wireless sensor networks," *Computer Communications*, vol. 30, no. 4, pp. 863–872, 2007.

[14] H. Choi, J. Wang, and E. Hughes, "Scheduling on sensor hybrid network," in *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, 2005, pp. 503 – 508.

[15] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388 –404, 2000.

[16] J. Huang, S. Liu, G. Xing, H. Zhang, J. Wang, and L. Huang, "Accuracy-aware interference modeling and measurement in wireless sensor networks," in *International Conference on Distributed Computing Systems*, 2011, pp. 172–181.