

Node scheduling in heterogeneous IoT systems with cluster topology

Xu Luo¹, Jun Yang²

¹ Department of Information Engineering, Zunyi Medical University, Zunyi, Guizhou Province, China.

² Engineering Research Center of Metallurgical, Automation and Measurement Technology, Wuhan University of Science and Technology, Wuhan, Hubei Province, China

Abstract. IoT systems are heterogeneous network systems which consist of sensor terminals monitoring different objects. Terminal sensors have different sampling frequencies determined by minimum Nyquist sampling requirements. It is difficult to ensure synchronization when monitoring data of different sensors is transmitted to the data center. In consideration of sampling requirements of different terminals and in order to upload the sampling data timely and avoid data congestion, an applicable node scheduling scheme is necessary. In this paper node scheduling algorithms are proposed in hierarchical IoT systems with cluster topologies. The IoT systems include IoT alarming systems and IoT monitoring systems. In the simulation test, node scheduling methods are tested and the validity is demonstrated quantitatively.

1 Introduction

IoT(Internet of Things) refers to distributed collaborative networks which connect the Internet and huge smart objects such as RFID(Radio Frequency Identification) devices, infrared sensors, global positioning systems, laser scanners, etc[1]. The fundamental part of an IoT system is the sensing network which consists of sensor nodes, and the end terminal is the data center. Sensor nodes in IoT systems can connect with each other through both wireless and wired networks. Most IoT systems are used to monitor different objects and have many subsystems. For example, in an IoT system for health monitoring, there are different numbers and types of sensors which monitor different health information, such as temperature, pulse, heart rate, blood pressure and posture[2].

Information sampling is a fundamental problem in IoT systems, and due to multiple monitoring objects and different sampling frequency requirements, a multi-rate sampling problem is involved in IoT systems [3]. In most IoT systems, uploading the sampling data to the data center in real time is an important requirement, and applicable network scheduling schemes are necessary to ensure this requirement under constraints of sampling rules of different sensor terminals. And network congestion should also be avoided in scheduling.

In this paper, the node scheduling problem is discussed in hierarchical cluster IoT networks [4], in which there are heterogeneous sensor nodes with different sampling rates. And the research objects are the IoT alarm network system in which only the interest data is reported, and the IoT monitoring network system in which the monitoring data is reported in real time.

The following organization is: in Section 2 the related work is introduced, Section 3 is the problem statement, in Section 4, nodes scheduling algorithms are proposed, Section 5 is the simulation part, and the last is the conclusion part.

2 Related works

Some node scheduling problems have been discussed in the network control system (NCS). There are the RM (Rate Monotonic) scheduling algorithm [5], the EDF (Earliest Deadline First) scheduling algorithm [6] and the MEF - TOD (Maximum Error First - Try Once the Discard) scheduling algorithm [7]. In the RM algorithm, the priority of a task is assigned according to the duty cycle, and the smaller the duty cycle is, the higher the priority is. The algorithm is a static one, and priorities of different tasks are fixed. EDF is a dynamic task scheduling algorithm, and in the algorithm the priority is assigned according to the deadline of the task. The longer the deadline of the task is, the lower the priority of the task is. The task with the highest priority is always the first one to be executed. In MEF-TOD algorithm, the priority of a task is calculated by the MEF algorithm firstly. When multiple nodes compete for the authority in the network communication, the node with the highest priority sends the data and the other nodes discard current data packets.

Scheduling problems are also discussed in sensor networks. In wireless sensor networks, node energy is limited, so energy is the primary factor which is considered in scheduling algorithms. In scheduling algorithms in sensor networks, most nodes are divided

into sleeping nodes and non-sleeping nodes, the node is scheduled to switch between the two states. An overview on the node scheduling in wireless sensor networks is given in [8].

As the problem background is different, the above works cannot be adopted to solve the problems in this paper directly. The problems to be solved in this paper are node priorities assignment considering realtime data requirements and data integrity, node task and layer task scheduling algorithms, and the scheduling coordination between the node layer and the cluster layer. In this paper IoT systems are not divided into wired and wireless ones. Different from scheduling problems in NCS, realtime data and continuous scheduling between layers are considered in this paper. Different from scheduling problems in wireless sensor networks, network performance is taken more consideration, and energy problem is not considered.

3 Problem statements

Firstly, there are below symbols and assumptions:

Denote:

T_1 : which is the cost time of a node requiring to communicate with a cluster;

T_2 : which is the cost time of a cluster responding to communicate with a node;

T_3 : which is the cost time of a node uploading sampling data to a cluster in an acknowledge cycle;

Z_j^i : which is the minimum sampling interval of the node N_j^i in the subsystem S_i ;

t_j^i : which is the data updating time of the node N_j^i in the subsystem S_i , and it is also the prescheduling time.

Assumptions:

1. Time synchronization can be ensured in nodes, clusters, and the data center.

2. The network is a three-level system, including nodes, clusters and the data center.

3. IoT systems are centralized data acquisition systems, it is required that the data is uploaded to the data center.

4. In each sampling period, the data item is uploaded to the superior node, and there is no data fusion in each node.

4 Scheduling algorithms

Scheduling algorithms are discussed in the two IoT systems as below: 1. When the monitoring data is greater than a given threshold, the data is uploaded to the data center from the sensor terminal, otherwise discarded. Such a system is an alarm network system. 2. The information is sampled according to the minimum Nyquist sampling law, and uploaded in real time. Such a system is a monitoring network system.

4.1. Scheduling in an alarm IOT system

Layer 1: Scheduling between nodes and the cluster

A node has a fixed sampling frequency, discards the monitoring data when the data value is smaller than a given threshold, gives an alarm and uploads the data when the monitoring value is greater than a given threshold. Take the case of the communication between the node N_1 and the cluster C_1 as an example to illustrate the scheduling procedure.

Step 1: When the sensing value of the node N_1 is larger than a given threshold, the importance of the request information is set to be $P=1$. Node N_1 gives a communication request to the cluster C_1 .

Step 2: Cluster C_1 detects whether there are nodes which have given out communication requests. If there are multiple nodes, choose the node which has the largest information importance to response. If the cluster chooses the node N_1 , go to Step 7, otherwise go to Step 3.

Step 3: Delay $\Delta t = T_1$, and set the request information importance as $P=P+0.5$. If the node N_1 does not receive the response information from the cluster, go to Step 4, otherwise go to Step 7.

Step 4: Delay $\Delta t = \Delta t + T_2$, and set the request information importance as $P=P+0.5$. Node N_1 gives a communication request to the cluster again, if the cluster gives a response, go to Step 7, and otherwise go to Step 5.

Step 5: Delay $\Delta t = \Delta t + T_3$, and set the request information importance as $P=P+0.5$. Node N_1 gives a request to the cluster again, if the cluster gives a response, go to Step 7, and otherwise go to Step 6.

Step 6: Delay $\Delta t = \Delta t + T_1$, and set the request information importance as $P=P+0.5$. Node N_1 gives a request to the cluster again, if the cluster gives a response, go to Step 7, and otherwise go to Step 4.

Step 7: Cluster C_1 gives a response to the node N_1 , and then the node N_1 uploads the monitoring data to the cluster. When the communication is ended, set the request information importance of the node N_1 as $P=1$ again. Cluster C_1 and the node N_1 go to another communication cycle.

In the procedure above, to avoid that the communication request of a node is not responded for a long time, P , which is the importance of the request information is given. The specific communication procedure between the node and the cluster above is as follows: A node N_1 requires communicating and uploading the monitoring value to the cluster C_1 firstly, and this sub-procedure costs time T_1 . Then the cluster C_1 responds and gives a reply to the node N_1 and this sub-procedure costs time T_2 . At last, if the cluster C_1 gives a positive response, the node N_1 uploads the monitoring data to the cluster, and this sub-procedure costs time T_3 , otherwise the communication between the node and the cluster ends immediately.

The above communication procedure between C_i and N_i is an example, and can be extended to the communication between any node N_j and any cluster C_i .

Layer 2: Scheduling between clusters and the data center D .

Replace the cluster C_i with the data center D , and the node N_j with the cluster C_i , the scheduling algorithm in Layer 1 can be extended to be adopted in Layer 2.

4.2. Scheduling in an alarm IOT system

Sampling periods satisfy Nyquist minimum sampling requirements. The purpose of the monitoring task is to upload realtime samples to the data center. The scheduling algorithm is also divided into Layer 1 and Layer 2 two part.

Layer 1: Scheduling between nodes N_j^i and the cluster C_i

There is relevant information of the sensor nodes shown in Table 1.

Table 1. Node Information

Node	Node N_1^i	Node N_2^i	Node N_3^i	...	Node N_N^i
Sampling intervals	Z_1^i	Z_2^i	Z_3^i	...	Z_N^i
Pre-scheduling time	t_1^i	t_2^i	t_3^i	...	t_N^i

The pre-scheduling time is renewed according to

$$t_j^i = t_j^i + Z_j^i, j = 1, 2, \dots, N \quad (1)$$

Let the clock time be τ and assume that nodes obtain data successfully at each sampling time. Clusters traverse each node according to priorities $y_j^i, j = 1, 2, 3, \dots, N$ which are determined by the formula as below:

$$y_j^i = \tau - t_j^i \quad (2)$$

The higher the priority of the node is, the earlier the node is visited by the cluster.

Step 1: Calculate scheduling priorities according to (2). Sort the nodes by scheduling priorities, and let the orders be $y_1^i \geq y_2^i \geq y_3^i \geq \dots \geq y_N^i$.

Step 2: If $y_1^i < 0$, the cluster visits the corresponding node N_1^i after waiting for time $|y_1^i|$. If $y_1^i > 0$, the cluster visits the corresponding node N_1^i immediately. When the visiting procedure is over, time τ is renewed, $\tau = \tau + T_1 + T_2 + T_3$, and go to Step 3.

Step 3: Renew scheduling priorities according to (2) and go to Step 1.

The specific communication procedure between the node and the cluster is: The cluster C_i gives a communication requirement to node N_1^i firstly, and this sub-procedure costs time T_1 . Then the node N_1^i gives a feedback to the cluster C_i , and this sub-procedure costs time T_2 . Finally, if the node N_1^i has data to be uploaded, uploads the data and the renewed pre-scheduling time to

the cluster C_i , and this sub-procedure costs time T_3 , otherwise visiting procedure ends immediately.

Remark 1. Node number 1, 2, 3, ..., N change with priorities in ordering procedures.

Layer 2: Scheduling between clusters and the data center.

Assign the initial priority of the cluster according to the minimum sampling interval of the nodes in the cluster. If a node has the minimum sampling interval, the cluster where the node is has the highest initial priority. The data center traverses clusters according to priorities. If there are M clusters, let the initial visiting tag be $k=1$, counting tag be $c=0$. The scheduling procedure is as follows:

Step 1: The data center gives a communication request to the cluster C_k . When the cluster C_k has data to be uploaded, $c=c+1$, and go to Step 2. When no data should be uploaded, let the visiting tag be $k=k+1$. If $k > M$, go to Step 3, otherwise loop this step.

Step 2: The cluster C_k sends the data to the data center. Renew the priority of the cluster C_k as $M+c$; that is marking the cluster C_k as C_{M+c} . Make the visiting tag be $k=k+1$. If $k > M$, go to Step 3, otherwise go to Step 1.

Step 3: Sort the clusters by priorities again and remark the clusters from small priorities to large priorities. The clusters are remarked as $C_1, C_2, C_3, \dots, C_M$ again. Go to step 1.

The specific communication procedure between the data center and the cluster above is: The data center gives a communication requirement to a cluster firstly, and this sub-procedure costs time T_1 . Then the cluster gives a feedback to the data center, and this sub-procedure costs time T_2 . Finally, if the cluster has data to be uploaded, the data is transmitted to the center and this sub-procedure costs time T_3 , otherwise the visiting procedure ends immediately.

5 Simulation test

In scheduling algorithms, the main performances are network delay and data acquisition ability. In this paper, the data acquisition ability is measured by non-effective data sampling time which is the total time loss in failing communication between sensor terminals and clusters, as well as clusters and the data center. The larger the non-effective data sampling time is, the lower the data acquisition ability is.

Table 2. Time delay and non-effective data sampling time in an alarm IOT system

A	Average Time delay(s)	Non-effective data sampling time(s)
50	11.51	1448
80	9.75	2752
100	9.29	3169
140	8.78	3716
180	8.21	4064

Table 3. Time delay and non-effective data sampling time in a monitoring IOT System

Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Average Time delay(s)	Non-effective data sampling time(s)
50	50	50	50	50	50	429.50	658
150	150	150	150	150	150	49.50	810
30	60	90	120	150	180	121.56	759
10	60	110	160	210	260	53.77	589
50	100	150	200	250	300	27.94	868

The initial clock time is $\tau_0=0$ s. In the cluster network there are a data center, 5 clusters and 6 sensor nodes in each cluster. The total running time of each node is 1000s. Time parameters T_1, T_2, T_3 are all set to be 1s.

The sampling interval of a sampling node is a random integer between $[0, A]$ s, and the performances of an IOT alarm system with proposed scheduling algorithms in the simulation are shown in Table 2.

The performances of an IOT monitoring system with proposed scheduling algorithms in the simulation are shown in Table 3, and in the table $Z_1, Z_2, Z_3, Z_4, Z_5, Z_6$ are the sampling intervals of the nodes in each cluster.

In the tables above, “Average time delay” refers to the average time delay of each data item in data uploading from a sensor node to the data center.

The results show that the proposed scheduling algorithms can be executed successfully. From Table 1 and Table 2, it can be seen that the longer the sampling intervals are, the shorter the average time delays are, but the non-effective data sampling time becomes larger.

6 Conclusions

In this paper, node scheduling algorithms are proposed in the alarm IOT system and the monitoring IOT system in consideration of sampling rules of different terminal nodes and real-time data. Performances of the algorithms are test in the simulation part.

In the algorithms, the data of a node in scheduling is uploaded only after the previous node having uploaded the monitoring data. If limitations of the communication bandwidth and the induced delay are neglected, there is no network congestion in theory.

The algorithms are proposed in ideal network environment and the research is only a frame work. In the future work the node scheduling problem will be discussed under more relaxed assumption constraints, and mathematical optimization problems in scheduling will also be studied.

Acknowledgments

This work is supported by National Natural Science Foundation (NNSF) of China under grant 61463053, Natural Science Foundation of Guizhou Province in China under Grant LH[2015]7549, Funding for the Scientific Research Project of Zunyi Huichuan District under Grant ZHKH201711, and Funding for the Scientific Research Program of the Hubei Provincial Education Department under Grant D20171102.

References

1. P. Ray. A survey on Internet of things architectures. *Journal of King Saud University - Computer and Information Sciences* (2016)
2. Y. Ma, Y. Wang, J. Yang, Y. Miao. Big health application system based on health Internet of things and big data. *IEEE Access* (2016)
3. J. Jin, M. Palaniswami and B. Krishnamachari. Rate control for heterogeneous wireless sensor network: characterization algorithms and performance. *Computer Networks*, **56** (2012)
4. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless micro-sensor networks. *IEEE Transactions on Wireless Communication*, **1** (2002)
5. C. L. Liu and J. W. Layland. Scheduling algorithm for multi-programming in a hard-real-time environment. *Journal of the Association for Computing Machinery*, **20** (1973)
6. J. Y. Wang, T. Lin T, J. D. Wang, *et al.* Research on preemptions of preemptive EDF and improvement on its performance. *ACTA ELECTRONICA SINICA*, **32** (2004)
7. G. C. Walsh and H. Ye. Scheduling of networked control systems. *IEEE Control Systems Magazine*, **21** (2001)
8. J. J. Niu, Z. D. Deng and C. Li. Distributed Scheduling Approaches in Wireless Sensor Network. *ACTA AUTOMATICA SINICA*, **37** (2011)