Volume 6, Issue 6, June 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A New Queuing Method for Differentiated Service Wireless Sensor Networks Based Event QOS Parameters

Gorre. Narsimhulu
Research Scholar,
JNTU, Hyderabad, India

Dr. D. Srinivasa Rao
Professor in ECE Department,
JNTU, Hyderabad, India

Abstract-In Wireless sensor networks (WSN), the sink is only interested in to collect collective information of sensor nodes within the event radius and not in their individual data. It is insufficient for end-to-end network QoS parameters to measure the QoS support in WSNs. We propose event QoS parameters into WSN for non-end-to-end applications. The event QoS parameters reflect the actual need of applications in WSN. Moreover, we present improved differentiated service queuing service (IDQS)- a queuing algorithm to provide service differentiation for WSNs based on event QoS parameters by setting different packet's lifetime. We find the relationship between packet's lifetime and event QoS parameters. By using the strategy the sink can detect the urgent event reliably and in time. The result of simulation verifies the effectiveness of our approach by using network simulator(NS2)

Keywords-event Qos parmeters; IDQS; WSN

I. INTRODUCTION

Wireless sensor networks (WSNs) are revolutionizing the way people interact with the physical world. A large volume of sensor nodes are deployed in an ad hoc fashion and to collect data from the environment. For information dissemination in WSNs, it is important to send critical information back to the sink with high reliability and low delay [1]. Consider an application that monitors temperature in a forest so as to detect forest fires. The sensor has to differentiate between temperature readings of 70 F on a normal day from another reading of 1000' F due to forest fire. From the point of view of the network, the packet reporting 1000' F is much more important as it could indicate a forest fire. Hence, this data should reach the base station with high reliability and low latency. In this paper, we aim to provide different QoS for different kind of data in WSNs.

There are some differences in application requirement between WSNs and traditional networks [2] [4]: First of all, applications in WSNs are no longer end-to-end applications. Traffic mainly flows from a large number of sensor nodes to a small subset of sink nodes. Second, bandwidth may be an important concern for a group of sensors for certain time periods due to burst nature of sensor traffic. Third, packet losses in traffic generated by one single node can be tolerated to a certain extent since there always exists much redundancy in data. As a result, we are convinced that it is insufficient for end-to-end network QoS parameters to measure the QoS support in WSNs. We thereby need to propose some new non end- to-end parameters to measure the QoS support in WSNs. Collective identification is firstly proposed in ESRT [3]. The sink is only interested in the collective information provided by numerous sensor nodes and not in their individual reports. In accordance with this, ESRT does not require individual node IDs for operation. Collective QoS parameters are summarized in [2] [6]: collective latency, collective packet loss, collective bandwidth and information throughput. However, they do not provide a precise definition of each collective QoS parameter and there are not solutions for QoS support in WSNs.

There are many solutions of providing service differentiation for WSNs. SAR [12] is the first protocol for WSNs that includes a notion of QoS. A queuing model is designed for the case of coexistence of real-time and non-real-time traffic in each sensor node in [5]. UMIUSI [7] presents the methodology for designing WSN architecture for fast and reliable transmission of urgent information. A simple max-min fairness bandwidth allocation technique is proposed in [10]. SPEED [11] provides soft real-time end-to-end guarantee.

However, the above solutions are still based on end-to-end QoS parameters. In this paper, we propose to provide different QoS for different kind of data in WSN s based on a set of new non-end-to-end Event QoS parameters. To best of our knowledge, there has not been too much research work done for providing service differentiation based on non-end-to-end QoS parameters for WSNs.

What's more, the above solutions have a common characteristic, i.e., the service granularity is per-flow or per class based so that the scheduler must know exactly per-flow or per-class QoS requirements accordingly. Such per-flow treatment with storing voluminous information in networks units is the major cause of the scalability problem and will complicate the implementation. To avoid this problem, [8] proposed a new scheduling algorithm called Differentiated Queuing Service (DQS) to provide end-to-end QoS cost effectively according to the QoS requirements and end-to-end situation of applications such as path lengths. However, DQS was originally proposed for wired networks. It was improved for wireless mesh networks by using a cross-layer implementation in [9]. In this paper, we will propose an improved DQS (IDQS) for WSNs based on non-end-to-end Event QoS parameters.

The paper is organized as follows. Section II describes the concept of event group and event QoS parameters, showing the precise definition of each parameter. Section III discusses the principle of IDQS, showing the relationship between the packet's lifetime and the event QoS parameters. Section IV discusses the experiment performed, the simulation scenarios, metrics used and results obtained. Finally, the paper is summarized in Section V.

II. EVENT QOS PARAMETERS

End-to-end QoS parameters for traditional networks are unfit for WSNs. We thereby need to propose some new non end- to-end QoS parameters. As a whole, we term such parameters Event QoS parameters. In WSN s, when an event generated, the application needs to detect the event as quickly as possible and as reliably as possible. Every event has an affecting region. All the sensors covered by the event will report this event to sink, thus sink can reliably detect the event if it receives number m packets about the event in t seconds. The number m and t is application-dependent. If the packets of an event received by sink less than m in t seconds, the sink will think the event die out. The application itself is not end-to-end, i.e., one end of the application is the sink, the other end is not a single sensor node, but a group of sensor nodes. The sink is only interested in collective information of sensor nodes within the event radius and not in their individual data. Data flow from sensors which detect the same event are likely to be highly correlated and thereby containing much redundancy.

A. Event Group

So we suggest that all the sensors that detect the same event make up an event group (EO). Every node has an event table. There are three fields in the event table: EO identification (EID), event observation (EO), and registering time (RT). Each EO has an EID. Different EO has different EID. All the sensor nodes of a same EO have the same QoS requirements. EO is the observations of the event. For example the temperature is higher than a specified threshold. RT is the time when the item is registered in the event table.

When a node detects an event, it will: (see Fig.1). When the neighbour receives the broadcast message, it will consult its event table. If there is not a same event observation in the event table, the neighbour will register EID, EO from the broadcast message and current time as RT in a new item of its event table, and broadcast to its neighbours too.

From this way, all the nodes that detect same event have same EID and EO. Thus all the nodes that detect same event make up of an event group. We add a new packet header named event QoS header (EQH). There is a new field in EQH: EID. All the packets come from an EO have same EID. Thus, one end of the application is the sink node , the other end is the EO. Every packet's source address is its EID and destination is the sink.

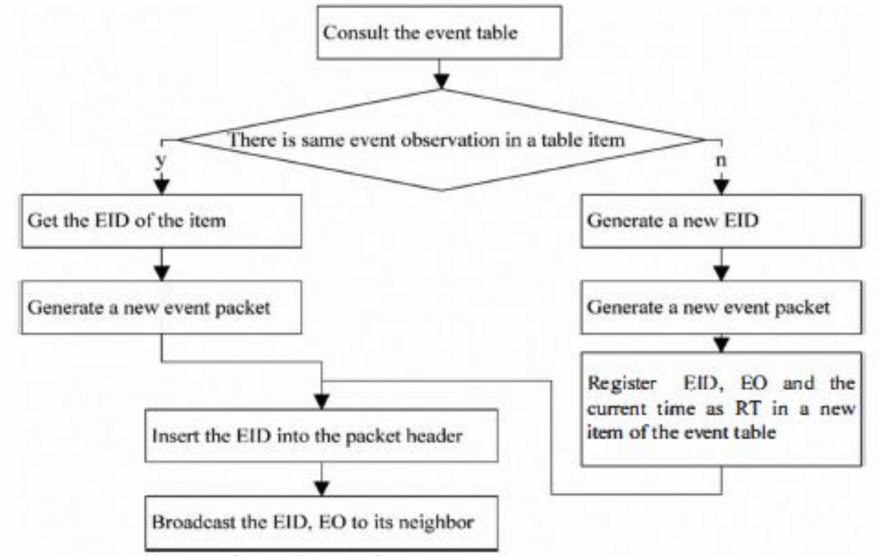

Fig.1.Generating an event group

B. Group Parameters

Now, we can define the Event QoS parameters:

Event latency (EL): The difference between the time at which a packet of an EO is generated and the time at which the sink receives number m packets of the EO after the packet's generating. Let g denote the time when a packet of an EO is generated, and a_{sm} denote the time when m packets of the EO arrives at the sink after g. Thus: EL= a_{sm} -g.

Event packet loss (ES): the number of packets of an EO lost during information delivery. Let N_l denote the number of packets of an EO lost during information delivery, and Ns denote the number of packets of the EO generated during information delivery. Thus: ES=N_l / N_s.

Event bandwidth (EB): the number of bits of an EO generated by the source sensor nodes per second.

Event throughput (ET): the number of bits of an EO successfully delivered by the networks per second.

Different kind of events requires different EL, ES and EB. The urgent information, a fire alarm for example, requires lower EL, lower ES and higher EB than other non-urgent information. The main theme of our scheme is to satisfy the EL, ES and EB requirements of different events.

III. DIFFERENTIATED QUEUING SERVICE (DQS) FOR WSNs

The original DQS was proposed to provide more granular QoS in wired networks with the following assumptions. Any application should have a maximum end-to-end delay that can be defined exactly. Therefore this delay is used as the lifetime of the associated packets. When a node receives a packet, it will check whether the packet's lifetime has expired. If so, the node should drop the packet immediately or put it into a best effort service queue.

A. Network models

For convenience, we consider WSNs as shown in Fig.2. N nodes are randomly distributed in a disk of radius R. The sink node is located at the canter of the disk. The transmission range of a node is r. All the nodes are the same. Sink transmits a hop value packet to its neighbours periodically. These neighbour nodes store the received hop value, increment it and transmit it to its neighbour nodes and so on until the whole sensor network is configured with different levels of hops. When a node receives a hop from its neighbour, it checks this value against its local hop value. If the local hop value is greater than the received one, the node updates its hop, increment this value and retransmit it to its neighbours. As illustrated in Fig.2, we divide the disk into several regions. Z_k means there are k hops between sink and the nodes in the region. Every node knows which region it belongs to. In order to eliminate unnecessary energy consumption, if a node receives a packet, it will choose a node that is in its transmission range and is closest to the sink to be the next hop [14].

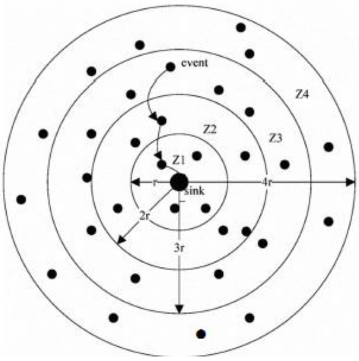

Fig.2. Network model

B Improved DQS (IDQS)

IDQS is based on the following assumption. Any packet should have a lifetime (D). If a packet can have an arbitrary D, then it can be set to certain values artificially. All the packets come from same EO have same D. The following notations are used in the discussion.

- a: packet's arrival time.
- e: latest time for the departure of a packet from a node subject to sink.
- d: packet's maximum delay at a node.
- g: packet's generate time.
- T: packet's maximum remaining lifetime upon its arrival at a node.
- D: packet's lifetime.

Subscript 'i' is often associated with these parameters to indicate that they are for node i. To simplify discussion, the propagation delay is not addressed since it is a constant for a given path. Therefore, for a path consisting of n nodes, we can have the following relations between the above parameters for a packet arriving at node i.

$$T_i = D - \sum_{l=1}^{i-1} \tilde{d}_l = D - (a - g) - - - - (1)$$

With T_i=D, where d_i is the actual delay that the packet experiences at the node i. The effective packet's maximum delay (which is defined as the interval between the arrival of the first bit and the departure of last bit from the node for the same packet) allowed at node i subject to its D, is given by

$$\hat{d}_i = T_i - \sum_{j=i+1}^{n} \tilde{d}_j \text{-------------}(2)$$

If the upstream nodes have bounded the delay as promised, then

$$e = a + \hat{d}_i \text{---------------------------}(3)$$

From (2), $\sum_{j=i+1}^{n} \tilde{d}_j$ should be available for node i to determine e_i with (3). However, this is difficult since $\tilde{d}_j$ and n will be only available in the future rather than upon the packet's arrival at node i. [9] proposed to send a probe packet to estimate the delay for the remaining journey of a packet. However this method is not fit for WSNs, because the probe packet will consume more energy. In our network model, all the packets' destination is the sink. Sink broadcast a hop value packet to its neighbours periodically. Every node knows how many hops between sink and itself. The hop value packet brings the timestamp (t_s) of the packet's generating time. Let t_j denotes the time when node i firstly receives the hop value packet. Then the node will get the delay between sink and itself: $t_{si} = t_i - t_s$. Although it is difficult to get the exact delay for the remaining journey of a packet, this delay can be estimated according to t_{si} :

$$\sum_{j=i+1}^{n} \tilde{d}_j \approx t_{si} \text{ --------------------------------(4)}$$

$$\text{Then, } e = a + Ti - t_{si} \text{--------------------- (5)}$$

from (l), (5),

we can get: $e = a + [D - (a - g)] - t_{si}$

$$\text{So, } \quad e = D + g - t_{si} \text{ --------------(6)}$$

We queue packets which arrive at a node according to e. Packets with smaller e are at or closer to the head of the queue and will be serviced earlier accordingly.

C. Improved DQS (IDQS) Packet's Lifetime

From (6), we can make a packet to be serviced earlier by setting smaller D. Let PL denotes the end-to-end delay of a packet. We get:

$$D = \max(PL) \text{-----------------(7)}$$

Let l denotes packet length, RES denotes ES requirement, t denotes EL requirement. As mentioned in Section 2, the sink is required to receive m packets of the event in t seconds. If m packets arrived at the sink, at least *m/ (1-RES)* packets must be generated. If EB is satisfied, generating *m/ (1-RES)* packets need $\frac{ml * 8/(1 - RES) + PL}{\text{EB}}$ seconds at least. We find the EB relationship between PL and EL as shown in Fig.3:

What's more EL<t, from (7), (8), we can get:

$$D \le t - \frac{ml * 8/(1 - RES)}{\text{EB}} \text{------------(9)}$$

If we choose D that satisfies (9) as lifetime of an event packet, the sink will receives m packets about the event in t seconds. If the sink can receives m packets about the event in t seconds, it will detect the event reliably. If the event is more urgent, the lower t, higher EB and lower ES are asked. Thus the smaller D can be set. From this way, we can service urgent events preferentially.

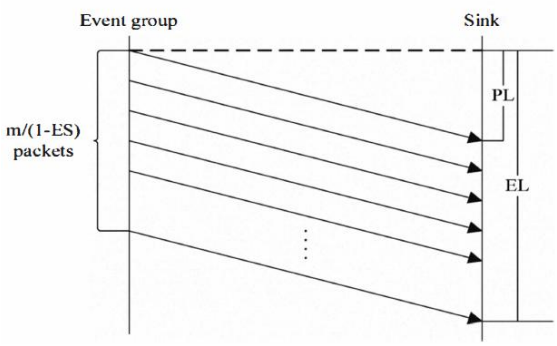

Fig.3 relationship between PL and EL

IV. PERFORMANCE EVALUATION

The IDQS algorithm was programmed in C and implemented using NS-2 simulator [13]. We choose a group of parameters for IDQS based on discussion above to implement it. Our simulation is designed to address the following concerns: Given an event's EL requirement (t) and m, we choose D that satisfies (9). We compare the simulation results with the requirements in order to invalidate our approach. We also compare performance between IDQS and Droptail in order to invalidate that IDQS improves the performance of urgent event and at the same time it does not worsen the performance of other events. We employ a typical WSN scenario that contains several occurrences of monitored events. The simulation time is 100s. We choose m=20. Event configurations are listed in TABLE.I.

Table. I: event configurations

Event No	1	2	3
EL requirement (sec)	10.0	1.0	5.0
ESrequirement(sec)	50%	50%	50%
EB(bits/sec)	1800	24000	3600
D	1.0	0.1	0.5

Event 1 represents the kind of application with low real-time requirement that can always be satisfied by network. Event 2 represents a sort of urgent applications. Event 3 represents applications requiring medium EL.

Fig.5 shows the actual ELs of IDQS and Drop tail. For IDQS, the actual ELs of event 2 remains steady and are all less than 1.0 seconds (EL requirement) during the simulation period. The actual ELs of event 1 and event 3 are all satisfy the EL requirements, too. For Drop tail, the actual ELs of event 2 fluctuate greatly according to the time and are more than 1.0 seconds many times during the simulation periods. The average actual EL of event 3 is longer than IDQS. The average actual EL of event 1 is almost the same as the IDQS.

Fig.6 shows the event throughput (ET) of IDQS and Drop tail. From the figure we find that the ET of event 2 of IDQS is much better than that of Drop tail. The ETs of event 1 and event 3 of IDQS are almost same with that of Drop tail. The results can validate our scheme. First, if we set D that satisfies (9), IDQS can make the sink receive m packets of an event in t seconds. Second, IDQS ensure the urgent event (event 2) to be serviced prior to other events and avoid other

V. CONCLUSION

In this paper, we normally bring event QoS parameters into WSN. The event QoS parameters only pay attention to the collective information of an event group rather than an individual node. They are much more tolerant than traditional end-to-end QoS parameters. They concern about whether the sink can receive m packets in t seconds and the throughput of an event group. The event QoS parameters reflect the actual need of applications in WSN. According to the event QoS parameters, we propose IDQS to provide different service for different event. Our schemes queue packets that arrive at a node according to the latest time for the departure of the packet from the node subject to sink. We can make a packet to be serviced earlier by setting smaller packet's lifetime. We find the relationship between the packet's lifetime and the event QoS parameters. The scheme is quick, stable and effective, which is validated by the simulation.

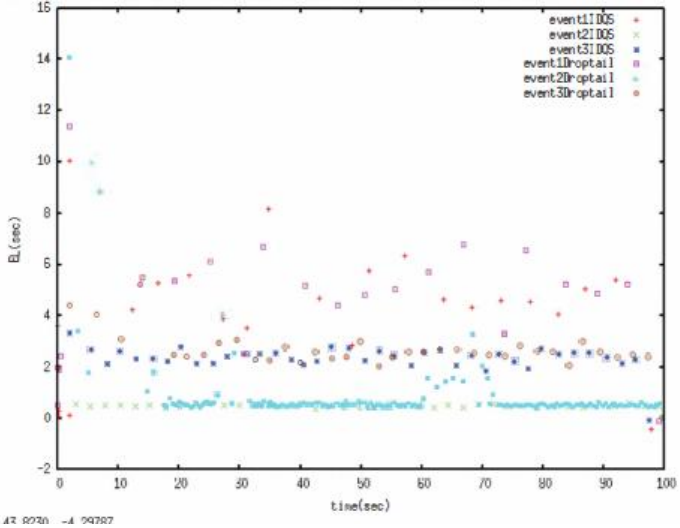

Fig.5. ELs of three events

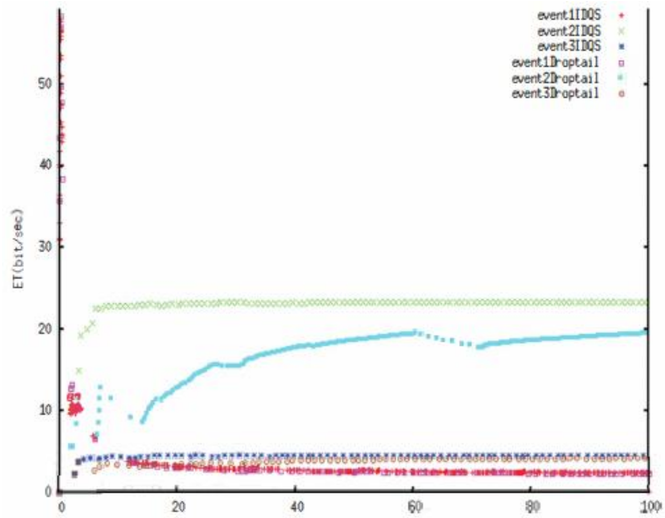

Fig.6. ETs of three events

REFERENCES

[1] Wang, V.M. and Vokkarane, "Optimal remote homing for providing service differentiation in information-aware multi-layered wireless sensor networks," Proceedings of IEEE International Conference on Communications, vol. 1, pp. 361-366, June 2006.

[2] Chen 0 and Varshney P K. "QoS support in wireless sensor networks:a survey," Proceedings of the International Conference on Wireless Networks, Bogart,USA:CSREA Press, pp.227-233, 2004.

[3] Sankarasubramaniam, Y., Akan, 0.8., and Akyildiz, I.F., "ESRT: Eventto- Sink Reliable Transport in Wireless Sensor Networks," in Proc. ACM MobiHoc'03, Annapolis, Maryland, USA, June 2003.

[4] Wang YL,Liu X H, Yin J P, "Requirements of Quality of Service in Wireless Sensor Network". In the Proc. of the International Conference on Networking, International Conference on Systems and Technologies(ICNICONSMCL' 06) , 2006(4)

[5] K. Akkaya, M. Younis. "An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks," Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003), Providence, Rhode Island, May 2003.

[6] Lin, X.Z., Zhou, J.J., Mu, C.D.: "Collective Real-Time QoS in Wireless Sensor Networks," Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (Wi COM 2006), pp. 1-4, 2006.

[7] T. Kawai, N. Wakamiya, M. Murata, "Designing a Sensor Network Architecture for Transmission of Urgent Sensor Information," Telecommunication Networks and Applications Conference, (A TNAC 2007): pp.169-174, 2007.

[8] S.M. Jiang, "Granular differentiated queuing services for QoS: structure and cost model", ACM SIGCOMM Computer Communication Review, v.35 n.2, April 2005

[9] X.F. Teng, S.M. Jiang, Gang Wei, Guikai Liu, "A Cross-Layer 'I Implementation of Differentiated Queueing Service (DQS) for Wireless Mesh Networks", Vehicular Technology Conference, (VTC Sprmg 2008), pp.2233-2237, 2008.

[10] H. Z. Abidin, F. Y. A. Rahman, "Provisioning QoS in Wireless Sensor o 20 40 60 80 100 Networks Using a Simple Max-Min Fair Bandwidth Allocation", Computer Science and Information Engineering, 2009 WRI World Congress, pp.44-48

[11] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," Proceedings of ICDCSW'03, 2003.

[12] K. Sohrabi, J. Gao, V. Ailawadhi and G. Pottie, "Protocols for SelfOrganization of a Wireless Sensor Networks," IEEE Personal Communications, pp. I 6-27, October 2000

[13] The Network Simulator ns-2. www.isi.edu/nsnamlns.

[14] Zhao Q, Tong L. "Energy efficiency of large-scale wireless networks: proactive versus reactive networking," IEEE Journal on Selected Areas in Communications 2005.23, pp.IIOO -1112.

[15] Jianliang Zheng , Myung J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4", IEEE Press Book, 2004.