International Journal of Engineering and Techniques - Volume 1 Issue 3, May -June 2015

RESEARCH ARTICLE OPEN ACCESS

Receiver Initiated Asynchronous Duty-Cycle MAC protocol for Burst Traffic

Inchara K[1], Shreenath K N[2],Dr.Srinivasa K G[3]
1(Student M.tech, CSE, SIT, Tumkur)
2 (Associate Professor, CSE,SIT,Tumkur)
3(Professor, CSE,MSRIT,Banglore)

Abstract:

Many energy-efficient Receiver Initiated Asynchronous Duty-Cycle MAC protocol for wireless sensor networks (WSNs) have been proposed. Most nodes suffer from significant performance Degradation for burst traffic, due to randomly waking up to communicate with each other. The proposed protocol is new receiver initiated asynchronous duty-cycle MAC protocol for burst traffic . By adaptively adjusting beacon time of the receiver and it schedules the sender listening time based on scheduled period, by this high energy efficiency and low end-to-end packet delivery latency for burst traffic is achieved. We have evaluated the performance of MAC through detailed ns- 2 simulation. The simulation results show that this protocol reduce end-to-end packet delivery latency and energy consumption under various data rates in different topologies compared with RI-MAC.

Keywords— Wireless sensor networks, duty-cycle, receiver-initiated, low latency, energy-efficient

I. .INTRODUCTION:

In wireless sensor networks (WSN), energy consumption is one of the most important factors because it is difficult to recharge or replace the battery of each sensor node. Therefore, in wireless sensor network most MAC protocols employ the duty cycling technique, to save energy the sensor nodes turn their radio on and off repeatedly.

There are two types of duty cycle MAC protocols: synchronous and asynchronous. In synchronous duty cycle MAC protocols such as S-MAC [1] and T-MAC [2], sensor nodes repeatedly wake up and sleep at the same time. Thus time synchronization leads to control message overhead and makes sensor nodes Expensive and complex. On the other hand, in asynchronous duty cycle MAC protocols, each sensor node wakes up and sleeps independently. Thus, time synchronization is not necessary. Most asynchronous duty cycle MAC protocols adopt a random wake-up interval in order to avoid repeated collisions.

Idle listening[1] is one of major energy waste sources in WSNs. Duty-cycle mechanism[1][2] is used to reduce energy consumption of idle listening. However, it suffers from sleep latency problem because the next hop node may be in sleeping state when data arrival. Sleep latency results in high end-to-end packet delivery latency with packet delivery hops increasing. It cannot be tolerant for some real-time applications.

Event detection is one of major applications in WSNs. The node commonly transmits the data packets with burst once detecting the event. The existing receiver initiated asynchronous duty-cycle MAC protocol suffer from serious performance degradation under burst traffic.

The contributions of this paper are as follows:

- Each receiver adaptively adjusts its beacon time, so that the packet can be forwarded as soon as possible and low end-to-end packet delivery latency is achieved.

- End-to-end packet delivery latency is decomposed into how to control single hop latency.
- In order to reduce idle listening and save energy consumption, the sender schedules its listening time according to its receiver's beacon time.

II. RELATED WORKS

The idea of receiver-initiated transmission in a MAC protocol is not new, but we make the first attempt to combine this idea together with duty cycling in the context of MAC protocols.Contention-based duty-cycle MAC protocols in WSNs can be classified into two categories: synchronous and asynchronous. Synchronous duty-cycle MAC protocols, such as SMAC[1][3],T-MAC[4], RMAC[5], DWMAC[6] and PRMAC[7], need the neighbor nodes to synchronize for communicating with each other. In contrast, asynchronous duty-cycle MAC protocols, such as BMAC, WiseMAC[8], XMAC[9], RI-MAC[10] and PW-MAC[11], do not require any synchronization between the neighbor nodes. The proposed MAC is an asynchronous duty-cycle MAC protocol, so we focus on asynchronous MAC protocols in this section.In these existing asynchronous duty-cycle MAC protocols, each node always wakes up randomly. This random wakeup introduces extra packet delivery latency. Especially for burst traffic, they suffer from serious performance degradation. Furthermore, because the sender does not know when the receiver wakes up in some MAC protocols (e.g. BMAC, XMAC and RI-\ MAC), it has to immediately wake up when it wants to transmit data packets, which wastes a great deal of energy.

In RI-MAC [7], as shown in Fig. 1, when a packet arrives at a sender, it wakes up and simply waits for a base beacon from its intended receiver. When the receiver wakes up, it sends a base beacon as an invitation for data transmission. once the sender receives the beacon message then the data transmission will start When the data is successfully received, the receiver sends a beacon as an acknowledgement (also used as an invitation for new data transmission). If a collision occurs, the receiver sends a beacon which includes the backoff window size (W).

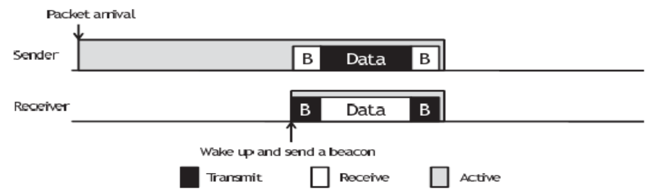

Fig. 1. Operation of RI-MAC.

III. DESIGN

A.Overview

Like RI-MAC, proposed MAC is an asynchronous duty-cycle receiver-initiated MAC protocol. Each node wakes up randomly to broadcast a beacon to notify that it is ready to receive data. When a node has queuing data packets receives the beacon from its intended receiver, it transmits data packets immediately. If no packet is received after broadcasting its beacon, the node will go to sleep to save energy. Time is divided into random period (RP) and scheduled period (SP) as in figure 2. During RP, each node randomly wakes-up to broadcast its beacon like RI-MAC.

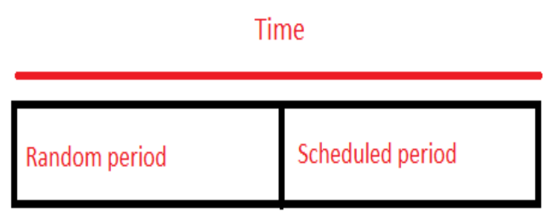

Figure 2:Timing Diagram

B.Latency Problem Description

We describe end-to-end packet delivery latency problem. Figure 3 represents the deployment of nodes. The network is composed of *N* nodes, including one or more than one source nodes, some relay nodes and one sink node. When any node detects the event, it generates data packets that are transmitted to the sink node through multiple relay nodes with burst. The interval between when the source node generates the data packet and when the sink node receives it is defined as end-to-end packet delivery latency *D*.

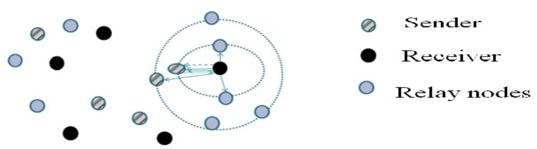

Figure 3: Network Model

Figure 4 illustrates single hop latency in Receiver Initiated Asynchronous Duty-Cycle MAC protocol. Node S transmits its beacon at *t*1 and receives the data packets. Then S keeps listening to the channel for the beacon from receiver R. Once S receives R's beacon successfully at *t*2, R receives the queued data packets from S. Thus the sender S needs to wait till the receiver wake-up, it's just a waste of time. If we ignore the packet propagation delay, single hop latency from S to R can be presented as:

$$d_{S,R}=t_2-t_1$$

If the duration of operational cycle is *Tcycle* and each node randomly wakes up, average single hop latency is *Tcycle/2*, thus we find that single hop latency can be reduced significantly if R wakes up just after S wakes up, but not randomly chooses its beacon time.

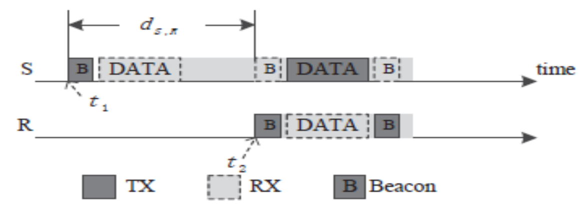

Figure 4: Single hop latency

C. Adaptive Beacon Time of Receiver

Adaptive beacon time mechanism of receiver is proposed to control single hop latency, by reducing this end-to end packet delivery latency also can be reduced significantly. When the sender transmits the data packet, it piggybacks its next beacon time in the data packet. Once the data packet is received successfully by the receiver, the receiver adaptively adjusts its next beacon time based on the information piggybacked.

Table I lists the variables defined:

Variable	Definition
$T_{beacon,i}$	next beacon time of sender i piggybacked in the data packet
T_{guard}	the interval between sender's beacon time and receiver's beacon time
$T^{j}_{beacon,i}$	next beacon time of receiver j based on next beacon time of sender i
T^{j}_{beacon}	receiver j's scheduled next beacon time based on each sender i
$T_{listen,i}$	sender's scheduled listening time for receiver i
T_{thre}	the duration of scheduled period in each node

Table I: VARIABLES DEFINITION

Initially, each node i is in Random period and chooses its beacon time *Tibeacon* randomly. Each node randomly wakes up at its beacon time to notify that it is ready to receive the data packets by broadcasting a beacon. Once the nodes detect the event it transmits the data packets to the sink node through multi-hop relaying. When sender S transmits the data packets to its receiver, S's next beacon time *Tbeacon,S* is piggybacked in the data packet. Upon receiver R receives the data packets, R updates its next beacon time corresponding to S based on *Tbeacon,S*:

$$T^{R}_{beacon,S}=T_{beacon,S}+T_{guard}$$

where *Tguard* can ensure that S receives all queued packets from its previous hop.

Especially, because R may have multiple senders, R should choose minimum next beacon time as its scheduled next beacon time:

$$T^{R}_{beacon}=\min_i(T^{R}_{beacon,i}>=TIME\_NOW)$$

D. Scheduled Listening Mechanism of Sender

In MAC, the sender plays a very important role. Firstly, the receiver schedules its next beacon time based on the sender's beacon time information piggybacked in the data packet. On the other hand, in order to reduce energy waste of idle listening, the sender schedules its listening time based on the receiver's beacon time. The most important difference between MAC and RIMAC for a sender is how the sender wakes up to listen to the channel when it wants to transmit the data packet. In RI-MAC, the sender immediately wakes up to listen to the channel when it has queued data packets.

In MAC, according to adaptive beacon time of receiver, the sender schedules its listening time during Schedule period to reduce idle listening and save energy. Because the node always immediately wakes up to listen to the channel when it wants to transmit the data packet during Random period, the sender doesn't miss its receiver. However, the sender starts to listen to the channel only just before its receiver wakes up during SP.

Several different Cases determine how the sender schedules its listening time as follows:

Case I:The sender S is in RP now. *Tlisten,R* is set as the time of the data arrival. It means that S starts to listen to the channel once the data packets arrival. This can guarantee that sender S doesn't miss the beacon of receiver R.

Case II:S transmits the data packet to R successfully. R will calculate its next beacon time as

$$TRbeacon,S = Tbeacon,S + Tguard$$

S knows when R will wake up to send its next beacon so that it can schedules its listening time at

$$Tlisten,R = TRbeacon,S.$$

Case III:As shown in Figure 5, S doesn't receive any data packet at *TSbeacon*, so R also cannot receive anything at *TRbeacon,S*. Because both of S and R are in SP, R schedule its next beacon time:

$$T^{IR}_{beacon,S}=T^{R}beacon,S+T_{cycle}$$

and S calculates its next scheduled listening time:

$$T^{1}_{listen,R}=T_{listen,R}+T_{cycle}=T^{R}_{beacon,S}+T_{cycle}$$

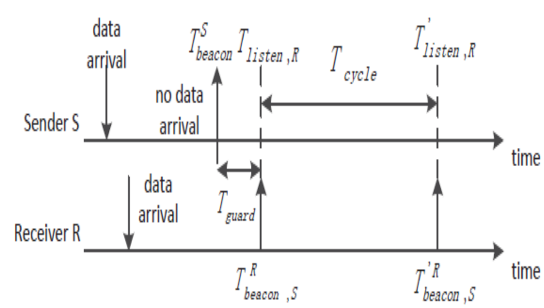

Figure 5: Case III

Case IV: R receives the data packet successfully and sends its ACK, but S doesn't receive ACK from R, just like Figure 6. R receives the data packet successfully but not knows that S doesn't receive ACK, so it calculates its next beacon time based on *Tbeacon,S* piggybacked in the data packet:

$$T^{1R}_{beacon,S} = T_{beacon,S} + T_{guard}$$

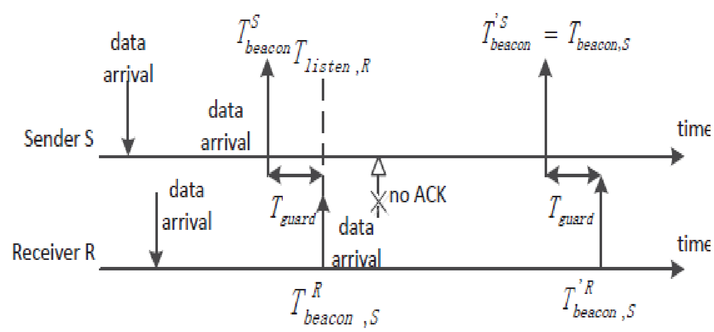

Figure 6: Case IV

Consequently, in order to guarantee that S and R can rendezvous, S should calculate its next scheduled listening time as:

$$\mathbf{T^{1}_{listen,R} = \min(T_{listen,R} + T_{cycle}, T_{beacon,S} + T_{guard})}$$

IV. CONCLUSIONS

In this paper we have proposed a new Receiver Initiated Asynchronous Duty-Cycle MAC protocol For Burst Traffic. The proposed MAC protocol will guarantee that it will Increase performance and this can be achieved by reducing end-to-end packet delivery latency, Increases the life time of sensor network and Saves energy.

The enhancement can be done to the burst traffic based on the priority concept.

REFERENCES

1. *W. Ye, J. S. Heidemann and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies, June 2002, pp. 1567-1576.*

2. *J. Polastre, J. Hill and D. Culler. Versatile low power media access for wireless sensor networks. In Proceedings of the 2nd International Conference On Embedded Networked SensorSystems, November 2004, pp. 95-107.*
3. *W. Ye, J. S. Heidemann and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Transactions on Networking, vol. 12(3), pp. 493-506, 2004.*

4. *T. V. Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the 1st International Conference On Embedded Networked Sensor Systems, November 2003, pp. 171-180.*

5. *S. Du, A. K. Saha and D. B. Johnson. RMAC: A routingenhanced duty-cycle MAC protocol for wireless sensor networks. In Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies, May 2007, pp. 1478-1486.*

6. *Y. J. Shu, S. Du, O. Gurewitz and D. B. Johnson. DWMAC: A low latency, energy efficient demand-wakeup mac protocol for wireless sensor networks. In Proceedings of the ACM international Symposium on Mobile Ad Hoc Networking and Computing, 2008, pp. 53-62.*

7. *T.Canli, A. Khokhar. PRMAC: Pipelined routing enhanced MAC protocol for wireless sensor networks. In Proceedings of the International Conference on Communication, June 2009, pp. 1-5.*

8. *A.El-Hoiydi and J. Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In Proceedings of International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, July 2004, pp. 18-31.*

9. *M. Buettner, G. V. Yee, E. Anderson and R. Han. XMAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In Proceedings of the 4th International Conference On Embedded Networked Sensor Systems, November 2006, pp. 307-320.*

10. *Y. Sun, O. Gurewitz and D. B. Johnson. RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In Proceedings of the 6th International Conference On Embedded Networked Sensor Systems, 2008, pp. 1–14.*

11. *L. Tang, Y. J. Sun, O. Gurewitz and D. B. Johnson. PWMAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In Proceedings of the 30th Annual Joint Conference of the IEEE Computer and Communications Societies, April 2011, pp. 1305-1313.*