ISSN:0975-9646

Avoidance of Collision and Overhearing in Wireless Sensor Networks

P.SRIVALLI, N.NAGAKUMARI, G.ANUPAMA, V.SRUJANA

KONERU LAKSHMAIAH UNIVERSITY, COMPUTER SCIENCE DEPARTMENT
VADDESWARAM, VIJAYAWADA, ANDHRAPRADESH

***ABSTRACT*-In wireless sensor network, we have large number of sensor nodes. The main task of a sensor node in a sensor network is to detect events, perform data processing within the network, and then transmit the data. Wireless sensor node has limited power source and the replacement of power source might be impossible. Energy efficiency is one of the important issues in wireless sensor networks. We can increase the energy efficiency by using MAC protocols. S-MAC protocol reduces the listen time by letting node go into periodic sleep mode. For collision avoidance S-MAC follows virtual carrier sense, physical carrier sense and RTS/CTS mechanism. S-MAC uses (Network Allocation Vector) NAV vector in virtual carrier sense. Maintenance of NAV vector throughout the transmission is difficult. In this paper, source node broadcasts transmission time to the neighboring nodes, to let the neighboring nodes go to sleep mode. It can reduce the collisions and overhearing.**

KEYWORDS: Energy efficiency, Collision, Overhearing, Medium Access Control (MAC), Wireless Sensor Network (WSN).

I.INTRODUCTION

The MAC sub layer acts as interface between logical link layer and the physical layer. The addressing mechanism of the Mac layer is called physical address. This MAC address is not modifiable.

The fundamental tasks of any MAC protocols are MAC creates the network infrastructure, since thousands of nodes are scattered in the field. MAC must establish communication links for data transfer. By using MAC, the network fairly and efficiently share communication resources between sensor nodes.

There is a duration field in each transmitted packet that indicates how long the remaining transmission will be. So if a node receives a packet destined to another node, it knows how long it has to keep silent. The node records this value in a variable called the network allocation vector (NAV) [1] and sets a timer for it. Every time when the NAV timer fires, the node decrements the NAV value until it reaches zero. When a node has data to send, it first looks at the NAV. If its value is not zero, the node determines that the medium is busy. This is called virtual carrier sense. Physical carrier sense is performed at the physical layer by listening to the channel for possible transmissions. The randomized carrier sense time is very important for collision avoidance. The medium is determined as free if both virtual and physical carrier sense indicates that it is free.

II.RELATED WORK

S-MAC protocol is designed for wireless sensor networks. The primary goal of this MAC protocol is reducing the energy consumption. It achieves good scalability and collision avoidance by utilizing a combined scheduling and contention scheme.

Periodic Listen and Sleep:

In sensor networks, sensor nodes sense the events and transmit the information to the sink node. In many sensor network applications, nodes are in idle for a long time if no sensing event happens. The data rate during this period is very low, it is not necessary to keep nodes listening all the time. This protocol reduces the listen time by letting node go into periodic sleep mode. For example, if in each second a node sleeps for half second and listens for the other half.

Each node goes to sleep for some time, and then wakes up and listens to see if any other node wants to talk to it. During sleep, the node turns off its radio, and sets a timer to awake itself later. The duration of time for listening and sleeping can be selected according to different application scenarios. This scheme requires periodic synchronization among neighboring nodes. All nodes are free to choose their own listen/sleep schedules. However, to reduce control overhead, we prefer neighboring nodes to synchronize together. That is, they listen at the same time and go to sleep at the same time. It should be noticed that not all neighboring nodes can synchronize together in a multi-hop network. Nodes exchange their schedules by broadcasting it to all its immediate neighbors. This ensures that all neighboring nodes can talk to each other even if they have different schedules.

If multiple neighbors want to talk to a node, they need to contend for the medium when the node is listening. The contention mechanism is using RTS (Request To Send) and CTS (Clear To Send) packets. The node that first sends out the RTS packet wins the medium, and the receiver will reply with a CTS packet. After they start data transmission, they do not follow their sleep schedules until they finish transmission. Neighboring nodes are free to talk to each other no matter what listen schedules they have. This scheme is quite easy to adapt to topology changes. The downside of the scheme is that the latency is increased due to the periodic sleep of each node. So the latency requirement of the application places a limit on the sleep time.

Choosing and Maintaining Schedules:

Before each node starts its periodic listen and sleep, it needs to choose a schedule and exchange it with its neighbors. Each node maintains a schedule table that stores the schedules of all its known neighbors. It follows some steps to choose its schedule and establish its schedule table.

1. The node listens for a certain amount of time. If it does not hear a schedule from another node, it randomly chooses a time to go to sleep and immediately broadcasts its schedule in a SYNC message, indicating that it will go to sleep after t seconds. We call such a node a synchronizer, since it chooses its schedule independently and other nodes will synchronize with it.

2. If the node receives a schedule from a neighbor before choosing its own schedule, it follows that schedule by setting its schedule to be the same. We call such a node as follower. It then waits for a random delay td and rebroadcasts this schedule, indicating that it will sleep in t−td seconds.

The random delay is for collision avoidance; so that multiple followers triggered from the same synchronizer do not systematically collide when rebroadcast the schedule.

3. If a node receives a different schedule after it selects and broadcasts its own schedule, it adopts both schedules. It broadcasts it own schedule before going to sleep.

We expect that nodes only rarely adopt multiple schedules, since every node tries to follow existing schedules before choosing an independent one. On the other hand, it is possible that some neighboring nodes fail to discover each other at beginning due to collisions when broadcasting schedules. Consider a network where all nodes can hear each other. The timer of one node will fire first and its broadcast will synchronize all of its peers on its schedule. If instead two nodes independently assign schedules, those nodes on the border between the two schedules will adopt both. In this way, a node only needs to send once for a broadcast packet. The disadvantage is that these border nodes have less time to sleep and consume more energy than others. Another option is to let the nodes on the border adopt only one schedule, which is the one it receives first. Since it knows another schedule that some other neighbors follow, it can still talk to them.

Maintaining Synchronization:

The listen/sleep scheme requires synchronization among neighboring nodes. Although the long listen time can tolerate fairly large clock drift, neighboring nodes still need to periodically update each other their schedules to prevent long-time clock drift. The updating period can be quite long. Updating schedules is done by sending a SYNC packet. The SYNC packet is very short, and includes the address of the sender and the time of its next sleep. The next-sleep time is relative to the moment that the sender finishes transmitting the SYNC packet. Receivers will adjust their timers immediately after they receive the SYNC packet. A node will go to sleep when the timer fires. In order for a node to receive both SYNC packets and data packets, we divide its listen interval into two parts. The first part is for receiving SYNC packets, and the second one is for receiving RTS packets. Each part is further divided into many time slots for senders to perform carrier sense. For example, if a sender wants to send a SYNC packet, it starts carrier sense when the receiver begins listening. It randomly selects a time slot to finish its carrier sense. If it has not detected any transmission by the end of the time slot, it wins the medium and starts sending its SYNC packet at that time. The same procedure is followed when sending data packets.

Collision and Overhearing Avoidance :

Collision avoidance is a basic task of MAC protocols. This MAC protocol adopts a contention-based scheme. It is common that all its neighbors receive any packet transmitted by a node even though only one of them is the intended receiver. Overhearing makes contention-based protocols less efficient in energy than TDMA protocols. So it needs to be avoided.

Collision avoidance:

Since multiple senders may want to send to a receiver at the same time, they need to contend for the medium to avoid collisions. Among contention based protocols, the 802.11 does a very good job of collision avoidance. S-MAC protocol follows similar procedures, including both virtual and physical carrier sense and RTS/CTS exchange. We adopt the RTS/CTS mechanism to address the hidden terminal problem. There is a duration field in each transmitted packet that indicates how long the remaining transmission will be. So if a node receives a packet destined to another node, it knows how long it has to keep silent. The node records this value in a variable called the network allocation vector (NAV) and sets a timer for it. Every time when the NAV timer fires, the node decrements the NAV value until it reaches zero. When a node has data to send, it first looks at the NAV. If its value is not zero, the node determines that the medium is busy. This is called virtual carrier sense. Physical carrier sense is performed at the physical layer by listening to the channel for possible transmissions. The medium is determined as free if both virtual and physical carrier sense indicates that it is free. All senders perform carrier sense before initiating a transmission. If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again. The packet transmission follows the sequence of RTS/CTS/DATA/ACK between the sender and the receiver.

Overhearing avoidance:

In 802.11 each node keeps listening to all transmissions from its neighbors in order to perform effective virtual carrier sensing. As a result, each node overhears a lot of packets that are not directed to it. This is a significant waste of energy, especially when node density is high and traffic load is heavy. Our protocol tries to avoid overhearing by letting interfering nodes go to sleep after they hear an RTS or CTS packet. Since DATA packets are normally much longer than control packets, the approach prevents neighboring nodes from overhearing long DATA packets and the following ACKs.

S-MAC efficiently transmits a long message in both energy and latency. A message is the collection of meaningful, interrelated units of data. It can be a long series of packets or a short packet, and usually the receiver needs to obtain all the data units before it can perform in-network data processing or aggregation. The disadvantages of transmitting a long message as a single packet are the high cost of re-transmitting the long packet if only a few bits have been corrupted in the first transmission. However, if we fragment the long message into many independent small packets, we have to pay the penalty of large control overhead and longer delay. It is so because the RTS and CTS packets are used in contention for each independent packet. This protocol is to fragment the long message into many small fragments, and transmit them in burst. Only one RTS packet and one CTS packet are used. They reserve the medium for transmitting all the fragments. Every time a data fragment is transmitted, the sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment. If a neighboring node hears a RTS or CTS packet, it will go to sleep for the time that is needed to transmit all the fragments.

III.PROPOSED WORK

In collision avoidance, S-MAC protocol maintains the NAV vector for virtual carrier sense. Maintenance of this NAV vector through out the transmission is difficult. In this paper we introduce a new mechanism. In this the source node broadcasts transmission time to the neighboring nodes, to let the neighboring nodes go to sleep mode. It can reduce the collisions and overhearing.

Collision and Overhearing Avoidance:

Collision avoidance is a basic task of MAC protocols. This MAC protocol adopts a contention-based scheme. It is common that all its neighbors receive any packet transmitted by a node even though only one of them is the intended receiver. Overhearing makes contention-based protocols less efficient in energy than TDMA protocols. So it needs to be avoided.

Collision Avoidance:

Since multiple senders may want to send to a receiver at the same time, they need to contend for the medium to avoid collisions. Among contention-based protocols, the 802.11 do a very good job of collision avoidance. This protocol follows similar procedures, including both virtual and physical carrier sense and RTS/CTS exchange. We adopt the RTS/CTS mechanism to address the hidden terminal problem. In this mechanism source node broadcast transmission time to the neighboring nodes. After receiving this packet all the neighboring nodes except nodes those are participate in the current transmission go to sleep mode. Then the node determines that the medium is busy. This is called virtual carrier sense. Physical carrier sense is performed at the physical layer by listening to the channel for possible transmissions. The medium is determined as free if both virtual and physical carrier sense indicates that it is free. So the current transmission will be carried out without collisions. All senders perform carrier sense before initiating a transmission. If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again. Broadcast packets are sent without using RTS/CTS. Unicast packets follow the sequence of RTS/CTS/DATA/ACK between the sender and the receiver.

Overhearing Avoidance:

In 802.11 each node keeps listening to all transmissions from its neighbors in order to perform effective virtual carrier sensing. As a result, each node overhears a lot of packets that are not directed to itself. This is a significant waste of energy, especially when node density is high and traffic load is heavy. In proposed mechanism, if any node wants to transmit data packet to the other node then source node calculates transmission time. Source node broadcast transmission time to neighboring nodes then all the neighboring nodes except the nodes those are participate in the current transmission go to sleep mode. Therefore overhearing is avoided because at the time of current transmission neighboring nodes are in sleep mode. This approach prevents neighboring nodes from overhearing long data packets and the fallowing ACKs.

IV. CONCLUSION

In proposed mechanism, the drawback of NAV vector maintenance is replaced. By using this NAV (network allocation vector) vector the neighboring nodes can know how much time they are in sleep mode, i.e at each time NAV vector tells the remaining transmission time of the packet. In this mechanism, broadcast the transmission time to the neighbors and the neighboring nodes go to sleep mode until the end of the current transmission.

REFERENCES

[1] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the IEEE Infocom, pages 1567– 1576, New York, NY, June 2002. IEEE.

[2] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the First ACM SenSys Conference, pages 171–180, Los Angeles, California, USA, November 2003. ACM.

[3] Sha Liu, Kai-Wei Fan and Prasun Sinha , "An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks", IEEE SECON,2007

[4] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In The Second ACM Conference on Embedded Networked Sensor Systems (SenSys), pages 95–107, November 2004.

[5] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: A short preamble MAC protocol for duty-cycled wireless sensor networks. In Proc. Sensys'06, 2006.

[6] A. El-Hoiydi and J. Decotignie. Low power downlink MAC protocols for infrastructure wireless sensor networks. ACM Mobile Networks and Applications, 10(5):675–690, 2005.

[7] Surjya Ray, Ilker Demirkol and Wendi Heinzelman, "Advertisement-based MAC Protocol for Wireless Sensor Networks".

[8] Kyle Jamieson, Hari Balakrishnan, Y. C. Tay, "Sift: A MAC protocol for Event driven sensor networks," May. 2003.

[9] Zhenzhou Tang, Qian Hu, "ALLEE-MAC: An Adaptive Low Latency and Energy Efficient MAC protocol for Wireless Sensor Networks" .

[10] Ilker Demirkol, Cem Ersoy, and Fatih Alagöz,"survey: MAC layer protocols for wireless sensor networks".

[11] Michael Buettner, Gary V. Yee, Eric Anderson, Richard Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks"

[12] Kyung Tae Kim, Won Jun Choi, Myung Jin Whang, and Hee Yong Youn, "Context Adaptive MAC Protocol for Wireless Sensor Networks".

[13] Rajgopal Kannan, Ram Kalidini, S. S. Iyengar "Energy and rate based MAC protocol for Wireless Sensor Networks" SIGMOD Record, Vol.32, No.4, December 2003.

[14] Anirudha Sahoo, Prashant Baronia "An Energy Efficient MAC in Wireless Sensor Networks to Provide Delay Guarantee", Local & Metropolitan Area Networks, 2007. LANMAN 2007. 15th IEEE Workshop on.
[15] Barroso A, Roedig U, and Sreenan C. μ-MAC: an energy efficient medium access control for wireless sensor networks[C]//Proc. of the Second European Workshop on Wireless Sensor Networks, Istanbul, 2005:70 – 80
[16] An Energy-Efficient MAC Protocol for Wireless Sensor NetworksWei Ye, JohnHeidemann, Deborah Estrin.

P.Srivalli received the E.Tech degree from the , Loyola Institute of Technology and Management, in 2009 and the M.Tech degree in Computer Science and Engineering from the KLUNIVERSITY, vaddeswaram, in 2011. She is
Working as Assistant Professor in computer science and Engineering KLUniversity. She is currently working toward the Research in the Wireless networks.
KLUniversity. Her research interests include computernetworks,wireless communication networks ,data mining.