JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Protocol Design and Implementation

Rajeev Kumar, Research Scholar
Dr. Manisha Kumari Deep, Professor
School of Computer Science and IT
YBN University, Ranchi-834003, India
Corresponding author: - rajeevkumarybn@gmail.com

Abstract: - Data generated by sensors is processed and communicated in a store and forward manner. Based on this scheme, each node sets a wakeup timer and goes to sleep for a particular period of time. When the timer expires, the node wakes up compulsorily and listens into the medium to find if it needs to communicate with any other node.

Keywords: - Dynamic Power Management, SYNC, node

Overview of the protocol

A large network comprising of many sensor nodes which have limited storage, communication and processing capabilities is assumed. Configuration of nodes is as in the form of an ad hoc, self-organized and self-managed wireless network. Data generated by sensors is processed and communicated in a store and forward manner. The applications supported by the network are assumed to swing between long idle periods during which there is no event occurrence and bursty active periods in which data flows to the sink through message exchange among peer sensor nodes.

AEMAC uses the bursty profile of sensor applications to inherit high energy savings. Along with schedule selection, co-ordination, synchronization and adaptive listening strategies, it also adopts Traffic Aware Dynamic Power Management, adaptation to link quality and channel quality conditions and cross layer interaction of the Physical layer, the Data Link layer and the Network layers. These aspects are detailed below.

Periodic sleep and listen operations

It was given in above Chapter that by establishing a low duty cycle operation for sensor nodes, the energy consumption can be reduced since idle listening gets avoided. In this protocol, all the sensor nodes move into a sleep state during which their radio interfaces are turned OFF. Nodes in sleep state consume very little energy compared to those in listen state.

Based on this scheme, each node sets a wakeup timer and goes to sleep for a particular period of time. When the timer expires, the node wakes up compulsorily and listens into the medium to find if it needs to communicate with any other node.

A complete listen-sleep period forms a frame. Duty cycle which characterizes each frame denotes the ratio of the listen interval to the frame length.

Nodes have their freedom to schedule their own listen and sleep intervals. In order to reduce the control overhead needed to achieve communication between the nodes, the schedules should be coordinated.

Schedule selection and coordination

The listen and sleep schedules of the neighboring nodes are coordinated. Each node selects a schedule and exchanges it with its neighbors during the synchronization period. A schedule table having the schedule of all its known neighbors is maintained by each node.

In order to select a schedule, a node listens to the channel for a fixed amount of time equal to the synchronization period. If a node does not hear a schedule from another node, on the expiration of the synchronization period, it chooses its own schedule. After performing carrier sensing the node announces its schedule by broadcasting a synchronizing (SYNC) packet. Carrier sensing rules out the possibility of SYNC packet collisions. If a node happens to receive a schedule from a neighbor before announcing its schedule, it accepts the received schedule as its schedule. But the node has to wait till the next synchronization period, in order to announce its schedule to its neighbors. Sometimes a node may receive a different schedule after it chooses and announces its own schedule. This is due to the SYNC packet getting corrupted by collision or channel interference. In this case, it checks whether any other neighboring node is sharing the same schedule as its one. If not so, it discards its own schedule and accepts the new one. But if any other neighboring node is sharing the same schedule as its schedule, the node adopts both the schedules as it is aware that other nodes have adopted its schedule. Hence it has to wake up at the listen intervals of both the adopted schedules.

In some cases, neighboring nodes may fail to know each other due to the loss of a SYNC packet or delay. So nodes should perform neighbor discovery frequently. For this, a node listens periodically for the entire synchronization period. Those nodes that do not have neighbors presently should perform neighbor discovery frequently.

Schedule synchronization

Schedule synchronization should be performed periodically by neighboring nodes to prevent long term clock drift. Transmission of SYNC packet is done to update a schedule. The listen interval is divided into two subintervals for a node to receive both data packets and sync packets. Channel access by contending nodes during these intervals is regulated using a multi slotted contention window. The former time subinterval is dedicated for the transmission of SYNC packets. The latter time subinterval is dedicated for the transmission of data packets. In

both these time intervals, a time slot is randomly selected by the contending station. Then it performs carrier sensing and transmits its packet only if it finds the channel idle. Thus energy consumption due to unwanted collisions can be reduced. Data transmission uses the Ready To Send/ Clear To Send (RTS/CTS) handshake to achieve a secure exclusive access to the channel during data transmission. Both RTS/CTS packets have different fields which have information regarding the

identity of source and destination and the time duration for the whole transmission. So the neighboring nodes can delay their transmissions till this packet transmission ends, thus preventing un-necessary energy wastages due to collisions and idle listening. Moreover the neighboring nodes are made to sleep till the end of the packet transmission, thus preventing energy losses due to over hearing and thus increasing the overall bandwidth utilization.

Adaptive listening

If all the nodes follow the sleep schedule strictly, it may lead to an increased delay since the packets will get delayed at every hop due to the store and forward mechanism. To improve the latency, adaptive listening is used. During the listen period of a node, if it happens to overhear an RTS/CTS packet exchange between its neighboring nodes, it assumes that it may be the next hop node. So it ignores its wake up schedule and schedules an extra listening period for the time duration as overheard from the RTS/CTS packet till the end of that packet transmission. If the overhearing node receives an RTS it can continue with the transmission. The unnecessary transition between the two power levels is avoided thus improving the energy transmission. If the overhearing node does not receive an RTS, it enters the sleep state till the next scheduled listen interval.

Access control and exchange of data

To regulate the access to the shared channel, this protocol uses CSMA/CA based procedure, which includes both physical and virtual carrier sensing. It also uses the RTS/CTS handshake to get rid of the hidden and exposed terminal problem. Network Allocation Vector (NAV) is used for virtual carrier sensing.

NAV is a variable which has information about the remaining time till the end of the current packet transmission, stored in it. Initially, NAV is set to the packet duration field as indicated in the transmitted packet. On the passage of time, its value is decremented till zero. A node can only initiate a packet transmission if its NAV value reaches zero. Physical carrier sensing is by listening to the channel for some ongoing transmission as in ordinary CSMA. To avoid collisions and starvation, carrier sensing is randomized within a contention window. Only if both the virtual and physical carrier sensing indicate a free channel, a node is allowed to transmit.

To prevent overhearing, nodes are allowed to sleep on hearing an RTS/CTS exchange from the other nodes. Nodes set its NAV to the duration of the packet transmission and sleep till its NAV becomes zero. This leads to significant energy savings.

Packet transmission:

Node first senses the channel. If the channel is busy, it sleeps till the channel becomes free again. On sensing a channel idle, it sends an RTS packet and waits for the CTS packet from the destination node. On getting the CTS packet, it transmits the data packet. Transmission ends on receiving an acknowledgement packet. The communicating nodes will not sleep till the end of the transmission.

Message passing

A long message is divided into small fragments that a node can process. These fragments are sent in a burst after a single RTS/CTS exchange between the communicating nodes. The medium is reserved for the time needed to transmit an entire message. Each fragment is transmitted, on getting an acknowledgement from the receiving node, for the previous fragment. This process improves the application level performance.

Channel adaptive mac protocol

Packet transmission through a link of high quality consumes less energy than that through a "bad" link. Based on this observation, in this proposed scheme, each sensor node should possess the ability to decide the state of its communication unit with respect to the current condition of the wireless link between it and the sink. Every node estimates the channel state and link quality for each contending flow and its energy status. To represent the channel state and link state at the LLC queue, a flag is initiated. The flag can take three values: Good, Bad or Probe. The proposed protocol calculates a combined weight value based on these flags and the energy status. Then transmission is allowed only for those nodes with a weight value greater than a minimum threshold value. Nodes attempting to access the wireless medium with a weight value less than the threshold value will be allowed to transmit again when their weight becomes high.

The energy consumed in an idle state is less than that in Active state, but significantly greater than in the sleep state. Hence, intelligently switching to sleep state whenever possible will generally create significant energy savings. In this work, a Traffic Aware Dynamic Power Management scheme (TA-DPM) is designed. The design goal of the Dynamic Power Management scheme is, to minimize energy consumption by continuously turning OFF the radio interface of unnecessary nodes that are not included in the routing path. For this, the nodes are categorized into three types depending upon the state defined by the data transmission as Current Transmitting Node (CTN), Future Transmitting Node (FTN), and No Transmitting Node (NTN). A state may dynamically change whenever data traffic is transmitted. Then, only the CTN and FTN nodes are asked to wake up, while other NTN nodes can continuously remain in their sleep states.

Link quality estimation

The link quality or link metrics such as: Bandwidth, Delay, Loss rate are measured by the MAC layer. Link metrics are then introduced to the IP routing protocol. The link metrics are taken into account in order to calculate

the path for the new incoming flow by the routing algorithm. Each node computes the Residual bandwidth and Residual energy.

Residual bandwidth Rbw is estimated as

Rbw = Cbw-Ubw (1)

Where

bw C - channel bandwidth,

bw U - used or consumed bandwidth,

Similarly the residual energy *e R* is estimated as

Re = (Ce – Ue) (2)

where,

e C – Initial energy,

U - Used or consumed energy,

A radio model defines:

Eelec J/bit - Energy consumption of a transmitter-receiver

Eamp J/bit - Energy consumption of a transmitter amplifier for a signal to noise ratio

Etx - Transmitting energy of a radio beam

The following equation is the factor for sending a *k* (bit) message to a node with a distance of *d* (meters)

Z = (Etx / (Eelec + Eamp * *k* *d*d)) (3)

Now the Link Weight LQ can be calculated as the combined sum of residual bandwidth and residual energy

LQ = Rbw + Z * Re (4)

Channel condition estimation

Every node estimates the channel conditions for each contending flow. To represent the channel state at the Link Layer (LL) queue, a flag is initiated. The flag can take three values as GOOD, BAD and PROBE

GOOD: A flag is set by the node as GOOD, when it receives from the following flow: (i) a MAC-layer acknowledgment in response to a data frame, (ii) a CTS frame in response to an RTS frame, or (iii) an error-free data frame or RTS.

BAD: The node sets the flag to BAD after a transmission failure. If the collisions or channel errors cause a transmission failure, then the values of the Short Retry Limit (SRL) and the Long Retry Limit (LRL) are selected. If a transmission takes place without receiving an acknowledgement, it is noted by a Long Retry Counter (LRC) and a Short Retry Counter (SRC). By monitoring the values of LRC and SRC, the transmission failure and collisions can be detected.

PROBE: The node switches the flag from BAD to PROBE when a configurable timeout, named as Ptimer, expires. Ptimer starts to run whenever the channel state switches to BAD, and its initial value is doubled when a transition from PROBE to BAD occurs. The duration of Ptimer is reset to its initial value upon a transition from PROBE to GOOD.

Each node {Ni, i=1,2…} maintains a queue {Qi, i=1,2…}. Each queue has associated with it a weight {Wi ,i=1,2….} such that

Wi = LQi * 1, if flag = GOOD,

= LQi * -1, if flag = BAD.

Cross layer interaction

The link layer metrics like residual bandwidth and residual energy and the channel state information depicted in the Flag is conveyed into the Network layer to make intelligent routing decisions. The residual bandwidth and residual energy information are determined by every wireless sensor network node. Each node estimates a weight value based on its residual bandwidth and residual energy stores a copy of it and is also broadcast across the network during the Neighbor Discovery process. Each node on receiving the neighbor's packet, checks for the lesser value. The lowest weight value in the network is selected as the minimum weight value. This minimum weight value will be determined at run time during every Neighbor Discovery process, which is performed periodically. So at the end of the Neighbor Discovery process, all nodes become aware of the weight information of its neighbors and the minimum weight value of the network. During the Route Discovery process, the standard Route Request packet of the routing protocol is modified by including the following fields like host address, weight information and timestamp. This is broadcast and any node on receiving this, computes a weight value as the sum of its weight value and the incoming weight value; and this is appended in its Route Request packet and again broadcast. In this way, it reaches the sink node, which gets the total weight value of all nodes in its path. This is sent back to the source in the Route Reply packet. Source selects the route having the highest weight value as its transmission route.

If a node has data to transmit, it checks whether its weight value is greater than the minimum weight value. The nodes with high weights are allowed to transmit. Nodes attempting to access the wireless medium with a low weight value will be allowed to transmit again when their weight becomes high, till then the data is buffered at the node.

Traffic aware dynamic power management

The nodes are grouped into the following categories:

Current Transmitting Node (CTN): Any node currently participating in the actual data transmission (Like nodes 1 and 2 in Figure 1).

Future Transmitting Node (FTN): Any node to be involved in the actual data transmission (Like node 3 in Figure 1).

No Transmitting Node (NTN): Any nodes that are not included in a routing path and hence not involved in the actual data transmission at all (Like nodes 4-11 in Figure 1).

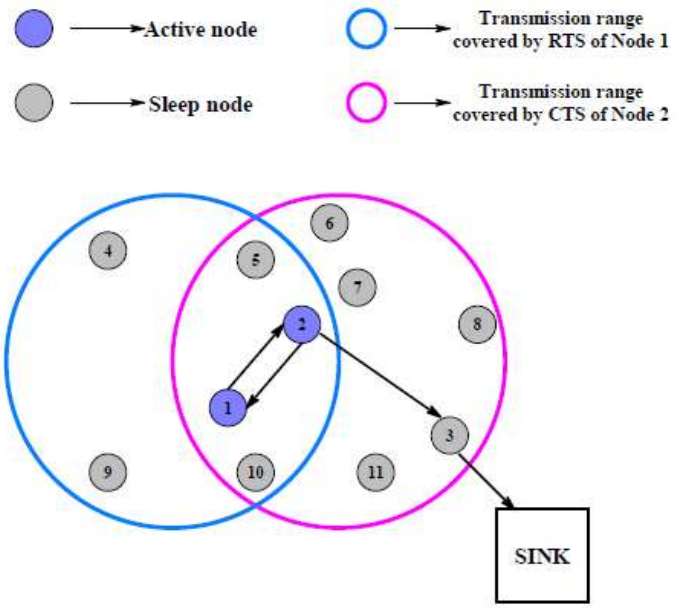

Figure 1 Classification of node states

The format of RTS/CTS control frames needs to be slightly modified from their original MAC protocol. This modification is for informing a node the fact that its state is changed to CTN or FTN in the corresponding period. The new RTS and CTS packet add only one field to the original packet. The newly added field in RTS is the final destination address, by which the receiver's routing agent can search for the next hop address. The new field of CTS is the next hop node address and it informs which node needs to be CTN to its neighbors.

Referring back to Figure 1, when node 2 receives 1's RTS packet including the final destination address of sink, its routing agent refers to the routing table for getting the next hop node (node 3) and informs back to its own MAC. The MAC agent of node 2 then transmits CTS packet including the information of node 3.

After receiving the CTS packet from node 2, node 3 changes its state to CTN and the other neighboring nodes become aware of the fact that they are FTN nodes.

Otherwise, if no such information about FTN is available in node 2's routing agent, it means the routing path is broken or has not yet been established..

Complete algorithm

The steps involved in the proposed MAC protocol are summarized as:

1. If the category of n_i is NTN, then
   1.1 Put n_i into sleep mode
   1.2 Terminate
2. Else
   2.1 Put n_i into awake mode
3. End if
4. For each node n_i, in awake mode then,
   4.1. $R_{bwi} = C_{bwi} - U_{bwi}$
   4.2. $R_{ei} = (Ce_i - Ue_i)$
   4.3. $LQ = R_{bwi} + Z * R_{ei}$.
   Where $Z = (Etx / (Eelec + Eamp * d^*d))$
   Eelec J/bit - Energy consumption of a transmitter-receiver
   Eamp J/bit - Energy consumption of a transmitter amplifier for a signal to noise ratio
   Etx - Transmitting energy of a radio beam
   4.4. LQ exchanged with neighbors during Neighbor Discovery.
   4.5. W_{min} the minimum threshold value is found based on minimum LQ values of all nodes.
   4.6. Path having highest weight value selected as the routing path.
   4.7 Nodes having to transmit, check:
      4.7.1. If node n_i receives ACK or CTS or Data from MAC layer, then
         Flag = GOOD
      4.7.2. Else If node n_i observes collision or transmission errors, then
         Flag = BAD
      4.7.3. Else
         Flag = PROBE
      4.7.4 End if
   4.8. If Flag = GOOD, then
      4.7.1 $W_i = LQ_i * 1$
   4.9. Else If Flag = BAD, then
      4.8.1 $W_i = LQ_i * -1$
   4.10. End if
5. End For
6. If $W_i > W_{min}$, then
   6.1 Transmission is allowed
7. Else
   7.1 Transmission is rejected.
9. End if.

Performance evaluation

Performance of this protocol is evaluated, for different performance metrics like Energy consumption, Delay, Delivery ratio and Throughput with respect to number of nodes, number of flows and Transmission Rate in the wireless sensor network, using the Network Simulator NS2.

Conclusion

This chapter proposed a Channel Adaptive MAC protocol with Traffic Aware DPM and competent packet scheduling and queuing in a wireless sensor network. The different aspects adopted as in SMAC to have efficient power management like periodic sleep-listen operations, schedule selection, co-ordination, synchronization, access control, data exchange and message passing were elaborated. Moreover the significant and unique aspects of the proposed scheme like Channel Adaptive MAC protocol, channel quality estimation, link quality estimation, cross layer interaction and Traffic Aware DPM were discussed in detail. Finally the complete proposed algorithm and the performance evaluation factors were also detailed.

Bibliography

[1] Rajeev K Shakya, Yatindra N Singh, and Nishchal K Verma. A correlation model for MAC protocols in event-driven wireless sensor networks. IEEE Region 10 Conference TENCON 2012. 1–6, IEEE, 2012.

[2] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. Ad hoc Networks, 3(3):325–349, 2005.

[3] Mano Yadav, Vinay Rishiwal, and KV Arya. Routing in wireless Ad hoc networks: A new horizon. Journal of Computing, 1(1):204–208, 2009.

[4] Stephen Makonin, William Sung, Ryan Dela Cruz, Brett Yarrow, Bob Gill, Fred Popowich, and Ivan V Bajic. Inspiring energy conservation through open source metering hardware and embedded real-time load disaggregation. IEEE PES Asia- Pacific Power and Energy Engineering Conference, APPEEC 2013. 1–6, IEEE, 2013.

[5] Amit Sinha and Anantha Chandrakasan. Dynamic power management in wireless 176
sensor networks. IEEE Design & Test of Computers, 18(2):62–74, 2001.

[6] Jeongyeup Paek, Ben Greenstein, Omprakash Gnawali, Ki-Young Jang, August Joki, Marcos Vieira, John Hicks, Deborah Estrin, Ramesh Govindan, and Eddie Kohler. The Tenet architecture for tiered sensor networks. ACM Transactions on Sensor Networks (TOSN), 6(4):34, 2010.

[7] Mohamed Younis and Kemal Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. Ad hoc Networks, 6(4):621–655, 2008.

[8] Arpit Tripathi, Pulkit Gupta, Aditya Trivedi, and Rahul Kala. Wireless sensor node placement using hybrid genetic programming and genetic algorithms. Organizational Efficiency through Intelligent Information Technologies. 125–144, IGI Global, 2013.

[9] Chris Knight, Joshua Davidson, and Sam Behrens. Energy options for wireless sensor nodes. Sensors, 8(12):8037–8066, 2008.

[10] Winston KG Seah, Zhi Ang Eu, and Hwee-Pink Tan. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP)-Survey and challenges. 1st International Conference on Wireless Communication,

Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1–5, IEEE, 2009.

[11] Mummadi Veerachary, Tomonobu Senjyu, and Katsumi Uezato. Neural network based maximum power point tracking of coupled inductor interleaved-boost- converter-supplied PV system using fuzzy controller. IEEE Transactions on Industrial Electronics, 50(4):749–758, 2003.

[12] Sujesha Sudevalayam and Purushottam Kulkarni. Energy harvesting sensor nodes: Survey and implications. IEEE Communications Surveys & Tutorials, 13(3):443– 461, 2011.

[13] Abha Rajoriya and Eugene Fernandez. Hybrid energy system size optimization and sensitivity evaluation for sustainable supply in a remote region in India. International Journal of Sustainable Energy, 32(1):27–41, 2013.

[14] Olga Moraes Toledo, Delly Oliveira Filho, Antonia Sonia AC Diniz, Jose Helvecio Martins, and Maria Helena Murta Vale. Methodology for evaluation of grid-tie connection of distributed energy resources-case study with photovoltaic and energy storage. IEEE Transactions on Power Systems, 28(2):1132–1139, 2013.

[15] Aurobinda Panda, Mukesh kumar Pathak, and Satya Prakash Srivastava. Solar direct torque controlled induction motor drive for industrial applications. International Journal of Renewable Energy Research (IJRER), 3(4):794–802, 2013.

[16] Stephen Makonin, Laura Guzman Flores, Robyn Gill, Roger Alex Clapp, Lyn Bartram, and Bob Gill. A consumer bill of rights for energy conservation. IEEE Canada

International Conference on Humanitarian Technology Conference, IHTC 2014. 1– 6, IEEE, 2014.

[17] Chung-Yang Sue and Nan-Chyuan Tsai. Human powered MEMS-based energy harvest devices. Applied Energy, 93:390–403, 2012.

[18] Rudolf Heer, Jurgen Wissenwasser, M Milnera, L Farmer, C Hopfner, and M Vellekoop. Wireless powered electronic sensors for biological applications. Annual International Conference on Engineering in Medicine and Biology Society (EMBC), 2010. 700–703, IEEE, 2010.

[19] Pooja Lakhlan and Aditya Trivedi. Energy harvesting-based two-hop D2D communication in cellular networks. International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016. 328–332, IEEE, 2016.

[20] Soumyajit Mandal, Lorenzo Turicchia, and Rahul Sarpeshkar. A low-power, battery-free tag for body sensor networks. IEEE Pervasive Computing, 9(1):71– 77, 2010.

[21] Fei Fei, John D Mai, andWen Jung Li. A wind-flutter energy converter for powering wireless sensors. Sensors and Actuators A: Physical, 173(1):163–171, 2012.

[22] Aoife M Foley, Paul G Leahy, Antonino Marvuglia, and Eamon JMcKeogh. Current methods and advances in forecasting of wind power generation. Renewable Energy, 37(1):1–8, 2012.

[23] Saurabh S Soman, Hamidreza Zareipour, Om Malik, and Paras Mandal. A review of wind power and wind speed forecasting methods with different time horizons. North American Power Symposium, NAPS 2010. 1–8, IEEE, 2010.

[24] Y Mishra, S Mishra, Fangxing Li, and ZY Dong. Eigenvalue analysis of a DFIG based wind power system under different modes of operations.Wind Power Systems. 191–213, Springer, 2010.

[25] Nicholas S Hudak and Glenn G Amatucci. Small-scale energy harvesting through thermoelectric, vibration, and radiofrequency power conversion. Journal of Applied Physics, 103(10):5, 2008.

[26] Xin Lu and Shuang-Hua Yang. Thermal energy harvesting for WSNs. IEEE International Conference on Systems Man and Cybernetics, SMC 2010. 3045–3052, IEEE, 2010.

[27] Sahar Ayazian, Vahid A Akhavan, Eric Soenen, and Arjang Hassibi. A photovoltaic- driven and energy-autonomous CMOS implantable sensor. IEEE Transactions on Biomedical Circuits and Systems, 6(4):336–343, 2012.

