Traffic-adaptive duty cycle adaptation in TR-MAC protocol for Wireless Sensor Networks

Sarwar Morshed, Mitra Baratchi, Geert Heijenk
University of Twente, The Netherlands
{s.morshed, m.baratchi, geert.heijenk}@utwente.nl

***Index Terms*—MAC protocol, TR-MAC, preamble sampling protocol, traffic-adaptive, duty cycle adaptation.**

***Abstract*—The Medium Access Control (MAC) layer can influence the energy consumption of a wireless sensor network (WSN) to a significant level. TR-MAC is an energy-efficient preamble sampling based MAC protocol for low power WSNs suitable for low data rate and low duty cycle scenario. However, low data rate is not always maintained in wireless sensor networks which often have to deal with event-driven scenarios where a sudden event rapidly increases traffic load within the network. In this paper we propose a traffic-adaptive duty cycle adaptation mechanism in order to provide responsiveness to traffic rate variations for TR-MAC protocol. This mechanism increases throughput and decreases packet delay while maintaining energy-efficiency without any extra information exchange among the sensor nodes in the network.**

I. INTRODUCTION

Energy-efficiency is an important criteria for MAC protocols to operate in wireless sensor networks. To achieve energy-efficiency, MAC protocols employ transceiver duty cycling to keep the sensor node's radio transceiver in sleep mode as much as possible to minimize the overall energy consumption. Therefore, the protocols that use transceiver duty cycling achieve energy-efficiency as a tradeoff between important performance parameters for example, lower throughput or higher delay. Asynchronous preamble sampling protocols tend to be the most energy-efficient category of MAC protocols for low data rate traffic scenario because they allow the nodes to sleep most of the time without network wide wake up synchronization [1]. Thus the nodes can independently wake up periodically to check any ongoing activity in the channel. In this way, basic preamble sampling protocols lack traffic-adaptivity due to their fixed time interval in consecutive listenings (check interval). Hence these protocols achieve energy-efficiency at an expense of higher per packet delay and lower throughput in presence of an event-driven higher traffic load scenario since improving both energy-efficiency and quality of service parameters is difficult.

We proposed a preamble sampling based energy-efficient MAC protocol called TR-MAC in [2], [3] and also proposed optimizations in [4]. The analytical model in these papers proposed check interval optimization to achieve energy-efficiency based on traffic load and is useful for low duty cycle and low traffic rate scenario. However, one node has to measure several parameters in order to use the analytical model to optimize the check interval. Furthermore, one node does not have any knowledge about its surroundings in the beginning. Moreover, some parameters often change during the network lifetime. Thus at a certain point of time one node might misestimate the parameters, e.g., packet arrival rate or the number of neighbors that in turn causes the node to miscalculate the optimized check interval using the analytical model. In addition, one node has to inform its pair node after changing its check interval to an optimum value when operating in synchronized link state by remembering each others future wake up time. This brings an overhead of extra information exchange because the sender node can no longer calculate the receiver node's next wake up time using the reception time of the last acknowledgement packet. All in all, maintaining energy-efficiency and at the same performing better in the quality of service parameters without extra overhead is very difficult for a MAC protocol to deal with variations in offered traffic load for event-driven scenarios. Therefore, a dynamic traffic-adaptive approach needs to be investigated that would simultaneously provide better performance for quality of service parameters without any extra information exchange and would minimize overall energy consumption in a WSN.

The contributions presented in this paper are: i) We presented a traffic-adaptive duty cycle adaptation algorithm to be used in the energy-efficient TR-MAC protocol for varying traffic, ii) We combined our traffic-adaptive duty cycle adaptation algorithm together with request based burst traffic transfer to improve performance in the quality of service parameters, iii) We implemented two other reference protocols in the OMNeT++ simulator using MiXiM simulation framework and compared TR-MAC protocol using duty cycle adaptation approach with those reference protocols.

The remainder of this paper is organized in the following manner: Section II presents related works. A brief description of TR-MAC protocol is given in Section III. Section IV deals with the duty cycle adaptation techniques of TR-MAC protocol. Afterwards, Section V represents the results and analysis. Finally Section VI provides the concluding remarks of our work and suggests the future work.

II. RELATED WORKS

There exists different strategies in the literature to adapt the duty cycle of a preamble sampling protocol based on different requirements to enhance the protocol performance. For example, BEAM [5] allows the nodes to adapt the duty cycle based on different requests from the neighborhood.

MaxMAC [6] enables duty cycle adaptation based on varying traffic load. EA-ALPL [7] performs the adaptation of duty cycle based on available topology information. Although these protocols provide duty cycle adaptation based on specific application scenario, they still do no provide adaptability in terms of energy availability on individual nodes. The preamble sampling protocol with a goal to handle increased traffic load can be categorized mainly in two branches: i) Request based methods, and ii) Traffic based methods.

i. Request based methods: Request based burst packet transfer was first proposed to be used in WiseMAC [8]. By assigning a bit representing more data needs to be sent when a sender has more packets to send, WiseMAC sender indicates the receiver that more packets are coming. The receiver then continues to listen by halting its periodic duty cycling for a moment. This approach can deliver many packets from a single sender to a single receiver. However, this approach suffer when traffic load increases from multiple sender nodes by creating a bottleneck scenario when many nodes want to send data to a single node. Later WiseMAC *more bit* [9] enhanced this technique by proposing a solution for bottleneck node scenario where a *Stay awake promise* flag was added in the acknowledgement indicating the receiver will be awake for at least one complete check interval. For a scenario with two nodes, only the first node that wins the contention can send data. The second node transmits right after the first node finishes transmission by overhearing all the subsequent acknowledgements sent from the receiver to the first sender. However, the potential receiver node has to remain awake for the rest of the check interval duration for this case, thus drains energy in continuous empty listening. In addition, the second node also spends extra energy by continuous overhearing of the acknowledgements sent from the receiver until the first node finishes its transmission. Another similar technique is used in BEAM [5] where the sender marks a bit in the data packet header to signal the receiver to double its duty cycle. Thus the sender can effectively shorten its preamble duration from the maximum one and can increase throughput.

ii. Traffic based methods: Traffic load based adaptation algorithms focus on estimating the traffic load and adapt the duty cycle based on application layer requirements. MaxMAC [6] is built on top of WiseMAC protocol with a threshold based duty cycle adaptation where a pair of synchronized nodes increase their duty cycle when the received packet rate crosses a certain threshold. The receiver node estimates the incoming traffic rate using a sliding window. From the beginning base state, the nodes move to the first state by doubling their duty cycle after exceeding the first threshold. Subsequently, the nodes move to the second state by doubling their duty cycle again after exceeding the second threshold. Finally, the nodes move to the third state and operate in CSMA after exceeding the third threshold. The change of duty cycle of the receiver node is always transmitted back using the acknowledgement. However, this extra information exchange is beneficial only for the sender in the synchronized node pair. Other neighboring nodes can gain the advantage only if they overhear the acknowledgement. Furthermore, operating in CSMA after crossing the third threshold sacrifices energy-efficiency to increase throughput. X-MAC [10] is also an asynchronous preamble sampling protocol that packetizes the preamble with destination address information. X-MAC outlines a mathematical model for traffic load estimation in the network, but it requires a certain minimum check interval to operate. Furthermore, X-MAC generally has a high per packet overhead that constraints the maximum achievable throughput. SCP-MAC [11] proposes a traffic load based duty cycle adaptation where the receiver node adds few extra listening just after receiving a packet from the sender. Once these extra listening cycles are not used, the receiver node deletes them and goes back to the basic preamble sampling.

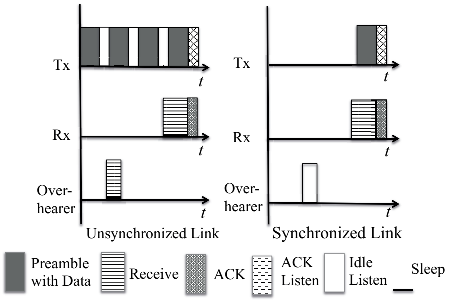

Fig. 1: TR-MAC protocol operation

Based on the literature studies, we proposed an energy-efficient traffic-adaptive duty cycle adaptation approach combined with request based burst packet transfer that will be suitable for varying traffic load scenarios to increase throughput and decrease delay for the TR-MAC protocol. Furthermore, the energy-efficiency can still be maintained using preamble sampling approach without any extra information dissemination within the network.

III. BACKGROUND: TR-MAC PROTOCOL OPERATION

TR-MAC is an energy-efficient preamble sampling protocol that enables the nodes to maximize sleeping time and wake up periodically independent of other nodes to sample the channel for any ongoing activity [2], [3]. The protocol has two operating states: unsynchronized and synchronized link states. In the beginning the sender node operating in unsynchronized link state sends preamble accompanying a small data packet together and waits for the acknowledgement from the receiver. The sender node repeats this procedure until it receives an acknowledgement. Adding a small data packet together with the preamble becomes feasible because of the fast synchronization of the underlying transmitted reference (TR) modulation. After receiving the preamble-data packet, the receiver node sends an acknowledgement back to the sender. However, the transmitter using transmitted reference modulation requires more transmission power than a normal transmitter [12], thus sending is more costly in TR-MAC than other protocols.

After this communication in unsynchronized link state, the nodes can remember each others next wake up time and move to the synchronized link state. If the sender remembers the receiver's next wake up, then it is called receiver-driven synchronization and the sender holds its packet transmission till the receiver's next wake up time. Thus the sender node successfully eliminates many extra iterations of the data and acknowledgement listen cycles to save energy. Figure 1 represents a sender (Tx), a receiver (Rx) and an overhearer for both the unsynchronized and synchronized link state operation of the TR-MAC protocol.

IV. TRAFFIC ADAPTATION IN TR-MAC PROTOCOL

The energy-efficient TR-MAC protocol was originally designed for low data rate and low duty cycle WSNs. Therefore, the MAC protocol has to adapt to the changes in traffic caused by a sudden event-driven scenario within the network that needs to be disseminated fast while maintaining its energy-efficiency. In order to provide traffic adaptiveness, we propose a combination of two approaches for TR-MAC protocol, namely: i) Request based burst packet transfer, and ii) Traffic-adaptive duty cycle adaptation. Request based burst transfer mainly focuses on maximizing the number of packet deliveries within a pair of nodes by the sender node signaling the receiver that it has more packets to transfer to the same receiver. Alternatively, traffic-adaptive duty cycle adaptaiton increases duty cycles of the receiver node depending on incoming traffic. Thus combining these two approaches would provide better performance for a varying traffic rate scenarios in WSNs. In the following sections, we elaborate each of these approaches.

A. Request based burst packet transfer

In this section, we propose a request based burst packet transfer methodology that enables a node to send multiple packets from its queue destined to a single node. In this case, the sender sets a flag, called *More Bit,* in the header section of the packet indicating that one more data packet is coming towards the receiver. Since this is a request based traffic adaptation approach, the receiver always accepts the request. Therefore, the receiver node sends the acknowledgement for the reception of this data packet back to the sender node, then continues to listen to receive the next data packet without adding any extra information in the acknowledgement packet. On the other side, the sender node receives the acknowledgement, then sends the next packet from its queue and waits for the acknowledgement from the receiver. Once again the receiver sends an acknowledgement back to the sender after reception of the new packet. In case the sender wishes to send more data packets to the same receiver, it again repeats the same procedure by setting the *More Bit* and the receiver individually acknowledges each data packet. When the sender has no more packet to send to the same receiver then it resets the *More Bit* to zero, which indicates that the receiver node can return to sleep after sending the last acknowledgement.

This request based burst packet transfer technique towards a single receiver node reduces the packet queueing delay at the sender since the sender node does not need to wait for the next duty cycle of the receiver node to transmit packets. Therefore, this technique increases throughput significantly. However, this request based burst packet transfer using *More Bit* could improve the traffic adaptivity only for a pair of nodes where one sender can send multiple packets to one receiver. Only one node that wins the first contention can send multiple packets to a potential receiver node for its one periodic listening. The rest of the nodes need to contend again and wait for at least the check interval duration for next periodic listening of the potential receiver node. As a result, a different mechanism is needed to provide a fruitful solution where many nodes want to send multiple packets to a single node.

B. Traffic-adaptive duty cycle adaptation

In this section, we propose a traffic-adaptive duty cycle adaptation algorithm to be used in preamble sampling based TR-MAC protocol in synchronized link state where a potential receiver node increases its duty cycle for increasing traffic and decreases its duty cycle for decreasing traffic. Moreover, after increasing its duty cycle by decreasing its check interval, one node does not need to send its current check interval value to its neighbors in the acknowledgement. The extra infromation exchange can be eliminated because the sender knows the receiver's duty cycle adaptation pattern and implicitly can estimate the next wake up time based on the difference between a number of consecutive acknowledgement reception times while operating in synchronized link state. In this way, this algorithm avoids the overhead of adding the updated duty cycle adaptation information in the acknowledgement packet and abstain from conveying this information to its neighbors.

In a WSN scenario, the preamble sampling approach achieves energy-efficiency by infrequent periodic listenings. Thus one sender node might experience extra delay if the receiver node does not increase the frequency of periodic listening depending on the offered traffic load. Therefore, the receiver node needs to have empty periodic wake ups frequently to accommodate new sender nodes or to ensure enough wake ups to match the traffic load offered to it possibly from multiple sender nodes. For a multiple access scenario when multiple sender nodes try to follow the future wake up time of a single receiver, there might exist a possibility of collision because of the time synchronization. However, the TR-MAC protocol has an inherent mechanism where one node always checks for an empty channel before sending. Hence only the sender node that wins the contention can send to the potential receiver node.

We propose a state-based duty cycle adaptation algorithm where one node adapts to the traffic rate by increasing its duty cycle if it realizes the traffic towards itself is increasing. Figure 2 illustrates the stage based mechanism and the decision making process to switch between these states using a timeline. Initially the node operates in state S_1 with duty cycles separated by base check interval (T_w). The first duty cycle S_1 in the figure represents the last duty cycle of the successive packet reception. When one node receives packets in K successive duty cycles, that node increases its duty cycle and moves to state S_2 by waking up more frequently at T_w/M time instances ($M = 3$ in this case) to accommodate more traffic instead of waiting till the base check interval. This same procedure is repeated to move to state S_3 by adding two extra wake ups at $T_w/9$ time instances and then to state S_4 by adding two more extra wake ups at $T_w/27$

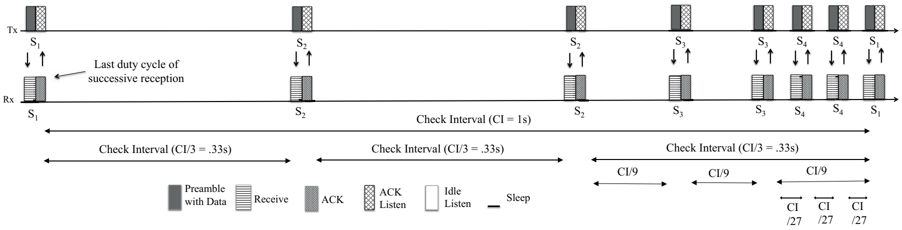

Fig. 2: Traffic-adaptive duty cycle adaptation for TR-MAC protocol

time instances consecutively to accommodate more traffic. As a result, the node can quickly reduce its check interval to adapt with the increasing traffic by a factor of the system parameter M having values $\{M, M^2, ... M^n\}$ where n is the number of levels. We chose $n = 3$ that results in four operational states $\{S_1, S_2, S_3, S_4\}$ where the first state is based on the base check interval and the following states depend on the successive values of M.

The protocol adapts to the decreasing traffic rate by deallocating the added extra wake ups and moving to a lower state when it does not receive any packets in L number of successive duty cycles. Hence the node deallocates the extra two duty cycles added previously and move one state down. For example, if a node is operating in state S_4 and does not receive packets in L consecutive duty cycles, then this node deallocates two extra wake ups and ramps down to state S_3. After successfully delivering the instantaneous increased traffic load, the node eventually moves down to initial base state S_1 to continue waking up after base check interval by deallocating all the previously added extra ups. Later in Section V, we further investigate optimum values for the system level up parameter K to allocate extra wake ups to move to a higher state and level down parameter L to deallocate the previously allocated extra wake ups to return back to a lower state.

At any point in time for a pair of nodes in synchronized link state, the sender node can determine the current check interval of the receiver from the reception time between two consecutive acknowledgements. Subsequently, the sender node can realize the operating duty cycle state of the receiver node. As a result, one node does not need to notify its neighbors about its adapted extra duty cycles and can avoid network-wide communication overhead. The fact that a potential receiver node adds extra wake ups in synchronized link state based on increasing traffic eventually allows the sender to deliver more packets. Therefore, the per packet delay is minimized from the base check interval duration to a factor of the base interval. This approach also enables other potential senders operating in unsynchronized link state to send packets faster to the same receiver using preamble sampling approach since it wakes up more frequently now. As a result, the potential receiver can overcome a bottleneck scenario. The receiver node might experience few empty listenings when it moves to a higher state with more wake ups, but it is insignificant in the long run because of the benefits achieved by faster multiple access, more throughput and less per packet delay.

In case of data packet loss, the receiver node may deallocate few extra wake ups for the absence of few data packets due to packet loss and may return to a lower state without notifying back to the sender. Nevertheless, the inherent mechanism of the preamble sampling approach together with duty cycle adaptation algorithm ensures that the sender can deliver the packets to the receiver with a minimized delay possibly less than or equal to the base check interval duration at the expense of few extra data-listen iterations.

V. RESULTS AND ANALYSIS

We implemented the TR-MAC protocol with traffic-adaptive duty cycle adaptation combined with request based burst transfer in the OMNeT++ simulator using MiXiM simulation framework. We also implemented and compared our protocol with the preamble sampling protocols X-MAC [10] and WiseMAC [8]. We chose X-MAC as it represents the basic preamble sampling protocol and only operates in unsynchronized link where nodes do not remember each others wake up time. And we chose WiseMAC because of its packet based burst transfer approach and its capability to operate in synchronized link, which is similar to TR-MAC protocol. We used two nodes to experiment with the duty cycle adaptation algorithm and evaluated its performance using system parameters mentioned in [4]. We considered data rate of 25 kbps, a queue length of 20 and Poisson distribution for traffic arrival. In the rest of this section, we compare these protocols in terms of throughput and delay per packet. Afterwards, we investigate into finding the optimum level up and down parameters for our proposed proposal for varying traffic loads.

i. Throughput: The throughput comparison is presented in Figure 3 using logarithmic scale for varying traffic load with 1s base check interval. The TR-MAC protocol uses level up parameter K set to 5 and level down parameter L set to 10. The reasons for choosing this combination is given in subsection *iii* of this section. Here we see that TR-MAC with the combination of duty cycle adaptation (DCA) together and *More Bit* provides significantly more throughput for higher traffic load. Firstly, the duty cycle adaptation mechanism enables the receiver node to wake up more frequently depending on the traffic. Secondly, the request based approach enables the sender to flush its queue every time it gets an acknowledgement from the receiver. Thus the TR-MAC protocol with a combination of both approaches can successfully deliver the offered load faster with very less packet drop. WiseMAC protocol also enjoys higher throughput because of the request based approach using *More Bit* but the performance gets limited for higher traffic due to the lack of duty cycle adaptation mechanism. In contrast, X-MAC

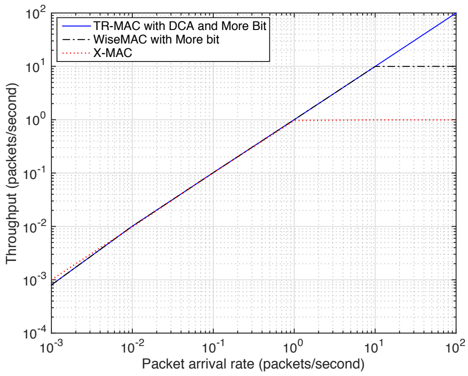

Fig. 3: Throughput with varying traffic

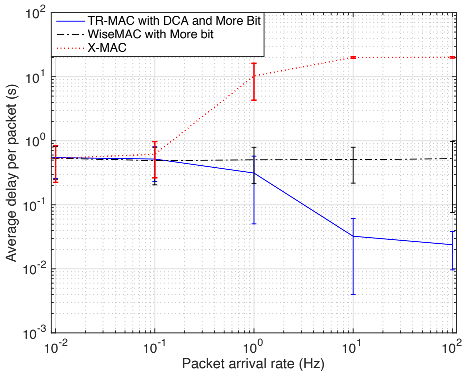

Fig. 4: Average delay per packet with varying traffic

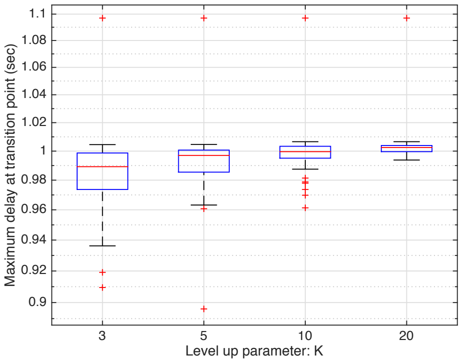

Fig. 5: Maximum delay during traffic load increase

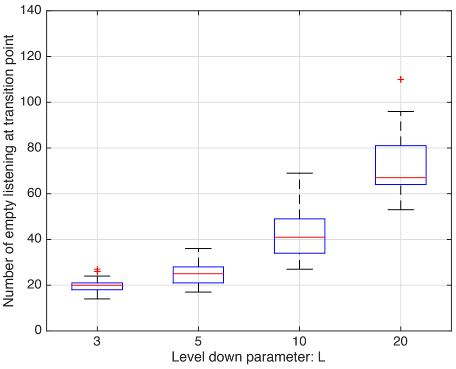

Fig. 6: Number of empty listening during traffic load decrease

protocol experiences a lower throughput and higher packet drop for higher traffic since it does not employ any traffic or request based adaptation mechanism.

ii. Delay: The average delay per received packet is represented in Figure 4 using logarithmic scale for varying traffic load. Here we see that TR-MAC protocol with DCA and *More Bit* experiences significantly lower per packet delay for increasing traffic because of the inherent mechanism of the protocol combining the sender-driven request based approach and receiver-driven traffic-adaptive duty cycle adaptation approach, as explained in the previous subsection *ii* of this Section and also in Section IV. The per packet delay remains constant for WiseMAC because of its usage of sender-driven request based *More Bit* approach to deliver all the packets in the queue whenever the sender gets access to the receiver. WiseMAC cannot achieve the per packet delay like TR-MAC protocol because of the fixed check interval duration of the receiver. X-MAC protocol experiences higher per packet delay for higher traffic as it lacks any traffic adaptation mechanism.

iii. Parameter analysis: We analyzed different combinations of system level up parameter K and down parameter L to find out the optimum combination ensuring both adaptability and stability. One node needs to adapt quickly to move to a higher state by decreasing the check interval after realizing an increase in traffic towards it. After moving to a new state, the node needs to operate in a stabilized manner by not moving back and forth to upper or lower states. Stability can be achieved by setting the parameters to a higher value, which in turn provides lower adaptability. Therefore, a tradeoff between adaptability and stability exists to decide for a combination of optimum level up and down parameter. To analyze this tradeoff, we evaluated the traffic rate variation from three perspectives: the traffic increasing part, the stable traffic part, and the traffic decreasing part. We did an experiment using 1s base check interval where initially we offered .5 packets/s of traffic for first 50s, then increased the traffic to 10 packets/s and maintained this rate for 1000s, afterwards the traffic was decreased to initial .5 packets/s.

Firstly, we analyzed the adaptability for fast state changing during traffic load increase by measuring the maximum delay at the transition point. Figure 5 represents a box-plot of the distributions for maximum delay experienced for different level up parameters K over 100 simulations. Here we see that a smaller level up parameter value provides relatively smaller delay to moveto the next state and eventually offers quick adaptivity. However, we don't see a significant difference in delay among the parameter values since the initial traffic was increased to a significantly higher traffic rate. Thus the node could move quickly to a higher state after receiving consecutive packets in K number of successive wake ups. Secondly, we analyzed the effect of adaptability during traffic load decrease for different level down parameters L. If one node takes too long to decrease to a lower working state, then it consumes extra energy because of empty listenings

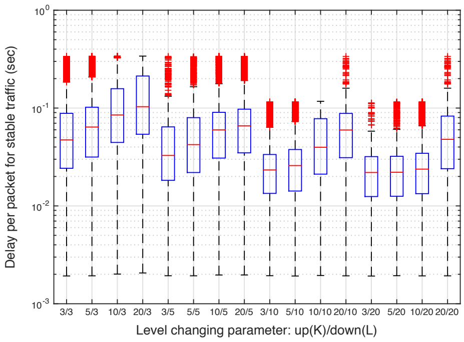

Fig. 7: Delay per packet for stable traffic

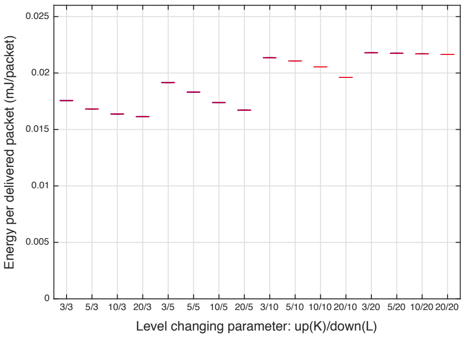

Fig. 8: Energy per packet for stable traffic

allocated earlier. For different level down parameters L, Figure 6 represents the number of empty listening that occurred for 20s after the traffic rate decreased. We see that a higher level down parameter implies higher number of empty listenings before moving down to a lower state. Thus a lower level down parameter provides more energy saving and more adaptability.

Finally, we analyzed the stability of the algorithm for a long stable traffic scenario from both delay and energy consumption perspectives. Figure 7 represents average delay per packet for various asymmetric combinations of level up and down parameters for stable traffic. Here we see that delay per packet is more with smaller L because the node switches quickly to a lower state after missing packets in subsequent wake ups, thus a higher level down parameter performs better with less delay per packet. Alternatively, a smaller K provides less delay because of quick adaptability. Energy per delivered packet is given in Figure 8 where we see that a combination of smaller K and higher L consumes more energy per delivered packet because this combination eventually delivers less number of packets as more delay is experienced.

Combining these results, we propose to choose a combination of smaller level up parameter K and higher level down parameter L to minimize both delay per packet and energy-efficiency per delivered packet, for example, $K = 5$ and $L = 10$. Given this setting, we could trade delay for energy consumption, and adaptability for stability by optimally tuning the level changing parameters.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a traffic-adaptive duty cycle adaptation algorithm and combined it with request based burst packet transfer for TR-MAC preamble sampling protocol to maximize throughput and minimize per packet delay and at the same time maintain energy-efficiency without any explicit information dissemination within the WSN. This approach is useful to quickly adapt to sudden variation of traffic generated by event-driven scenarios. We evaluated our protocol in comparison with others and showed that our proposed solution achieves better results in both energy-efficiency and quality of service parameters like throughput and per packet delay in traffic varying scenarios. Scalability can be achieved by multiple pair of nodes. For our future work, we would like to evaluate the algorithm for scalability using multiple link scenario and to compare our protocol to some other traffic-adaptive reference protocols.

ACKNOWLEDGEMENT

This work is supported by the Dutch Technology Foundation (STW) through the WALNUT project (no. 11317).

REFERENCES

[1] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, “Low energy operation in WSNs: A survey of preamble sampling MAC protocols,” *Comput. Netw.*, vol. 55, pp. 3351–3363, Oct. 2011.

[2] S. Morshed and G. J. Heijenk, “TR-MAC: an energy-efficient MAC protocol for wireless sensor networks exploiting noise-based transmitted reference modulation,” in *2nd Joint ERCIM eMobility and MobiSense Workshop, St. Petersburg, Russia*, pp. 58–71, June 2013.

[3] S. Morshed and G. Heijenk, “TR-MAC: An energy-efficient MAC protocol exploiting transmitted reference modulation for wireless sensor networks,” in *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM ’14, pp. 21–29, ACM, 2014.

[4] S. Morshed and G. J. Heijenk, “Optimization and verification of the tr-mac protocol for wireless sensor networks.,” in *WWIC*, vol. 9071 of *Lecture Notes in Computer Science*, pp. 396–410, Springer, 2015.

[5] M. Anwander, G. Wagenknecht, T. Braun, and K. Dolfus, “BEAM: A burst-aware energy-efficient adaptive MAC protocol for wireless sensor networks,” in *Seventh International Conference on Networked Sensing Systems (INSS)*, pp. 195 –202, June 2010.

[6] P. Hurni and T. Braun, “Maxmac: A maximally traffic-adaptive MAC protocol for wireless sensor networks,” in *Wireless Sensor Networks, 7th European Conference, EWSN 2010, Coimbra, Portugal, February 17-19, 2010. Proceedings*, pp. 289–305, 2010.

[7] R. Jurdak, P. Baldi, and C. V. Lopes, “Energy aware low power listening for sensor networks,” in *Second International Workshop on Networked Sensing Systems (INSS05)*, 2005.

[8] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks.,” in *ALGOSENSORS*, vol. 3121 of *Lecture Notes in Computer Science*, pp. 18–31, Springer, 2004.

[9] P. Hurni, T. Braun, and M. Anwander, “Evaluation of wisemac and extensions on wireless sensor nodes.,” *Telecommunication Systems*, vol. 43, no. 1-2, pp. 49–58, 2010.

[10] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys ’06, pp. 307–320, ACM, 2006.

[11] W. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle mac with scheduled channel polling,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys ’06, (New York, NY, USA), pp. 321–334, ACM, 2006.

[12] A. Meijerink, S. Cotton, M. Bentum, and W. Scanlon, “Noise-based frequency offset modulation in wideband frequency-selective fading channels,” in *16th Annual Symposium of the IEEE/CVT*, (Louvain-la-Neuve, Belgium), IEEE/SCVT, 2009.