Simulation-based Evaluation of DMAMAC - A Dual-Mode Adaptive MAC Protocol for Process Control

A. Ajith Kumar S.
Bergen University College, Norway
University of Agder, Norway
aaks@hib.no

Lars M. Kristensen
Bergen University College
Bergen, Norway
lmkr@hib.no

Knut Ovsthus
Bergen University College
Bergen, Norway
kovs@hib.no

ABSTRACT

Control systems automation is widely used in many industrial domains and have strong requirements on delay, throughput, robustness, and reliability. In the domain of networked control systems, the medium of communication is increasingly involving wireless communication along-side conventional wired communication. Issues ranging from energy efficiency and reliability to low-bandwidth have to be addressed to enable the transition to increased use of wireless communication. In earlier work, we have proposed the Dual-Mode Adaptive MAC (DMAMAC) protocol relying on a combination of Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA). The DMAMAC protocol is able to dynamically adapt to the two main states found in process control: the steady state and the transient state. Key requirements to the DMAMAC protocol are energy efficiency, low probability of state-switch failures, and a low state-switch delay.

The contribution of this paper is a comprehensive simulation-based evaluation of the original DMAMAC protocol along with the evaluation of a new pure TDMA-based variant of the DMAMAC protocol. Our results show that for processes where the steady state dominates, both variants of the DMAMAC protocol can reduce energy consumption by up to 45% in comparison to the closely related single-mode GinMAC protocol. Among the two variants of DMAMAC, the pure TDMA-based variant has the better energy efficiency and higher reliability. The simulation results also show that the hybrid TDMA-CSMA variant of the DMAMAC protocol has a probability of less than 0.3% for a state-switch failure in a given MAC superframe. The simulation study has impacted the design of the DMAMAC protocol by providing insights that have led to design changes in the originally proposed DMAMAC protocol in order to further reduce the state-switch delay between the steady and the transient state.

SIMUTOOLS 2015 August 24-26, Athens, Greece

Keywords

Wireless Sensor Actuator Networks, Process Automation, Process Monitoring and Control, Medium Access Control, MAC protocols, Energy efficiency, Reliability

1. INTRODUCTION

A Wireless Sensor Actuator Network (WSAN) [1] consists of sensors and actuators that use radios to send, relay, and receive information. WSANs are used across multiple application domains, including process and factory automation. The advantage of WSANs lies in reducing operating costs, the size of the devices used, and in increasing automation. Feedback-based control-loop automation is one of the main applications, and control systems that use wired or wireless solutions for data transfer are known as Networked Control Systems (NCSs) [4]. A general control system consists of a reference input, plant output, sensors, actuators, and a controller providing control input as shown in Fig. 1. Previously, NCSs used only wired communication due to the high reliability, low delay, and high bandwidth. Now, NCSs are adopting wireless communication co-existing with wired solutions. Given the fact that nodes in WSAN are generally battery powered, energy efficiency is an important requirement in addition to the real-time requirements.

In this paper, we focus on NCS based on wireless communication designed for process control applications. Processes are generally modelled using mathematical process models, which represent the characteristic features of the process. The process model also describes the change in physical quantities (e.g. temperature, pressure, and level) over time. In process control, the process being controlled could be in one of the two states at a given time: *steady* and *transient*. In the transient state, the process changes rapidly, i.e., the physical quantities change rapidly. For the process control to operate properly, the sensors need to communicate the rapidly changing dynamics of the process (in transient state) to the sink (gateway), which then aggregates this data

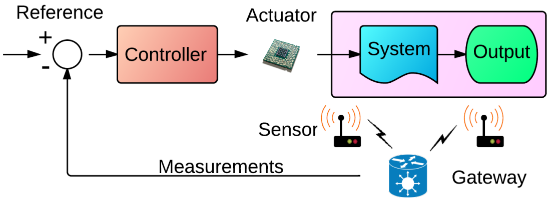

Figure 1: Control loop with wireless communication

and forwards processed control data to the actuators to act on this data. With time, the process starts stabilizing and the rate of change decreases. The process is then said to be in steady state. In steady state, the physical quantity is either constant or has minimal change within a given threshold interval. This threshold interval is also used to detect the need for a switch from steady to transient state. When the sensors measure values outside of the threshold interval, it notifies the sink to initiate a state-switch. This switch from steady to transient can be critical depending on the application requirements.

In [13], we proposed the Dual-Mode Adaptive Medium Access Control protocol (DMAMAC), designed to have two operational modes to cater for the two process control states. The protocol has a transient mode that supports the transient state, and a steady mode to support the steady state. The transient mode has data communication at a higher rate relative to the steady state. We designed DMAMAC for applications where steady state dominates the process operation. Generally, wireless sensor nodes are battery powered and the transceiver consumes more energy than the micro-controller and other parts [3]. Thus, the reduced communication in the steady mode results in energy savings. The transient mode still has energy consumption similar to single mode protocols. A further benefit of the DMAMAC protocol is that it has reduced interference to other operations in the vicinity due to a low duty cycle in the steady state. Previously, in [13] we proposed a hybrid protocol based on a combination of Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA). In this article, we simulate this protocol to study the performance in comparison to the GinMAC protocol [15] which is a well-established protocol for industrial monitoring and control. In addition, we develop and evaluate the performance of a new pure TDMA-based variant of DMAMAC (DMAMAC-TDMA) and compare it to the original DMAMAC protocol (DMAMAC-Hybrid) from [13].

The rest of the paper is organized as follows. Section 2 briefly introduces the design of the DMAMAC protocol and the related GinMAC protocol [15] which will serve as a baseline for comparison and design. The simulation setup based on the scenario in [13] is discussed in Sect. 3, along with the MiXiM [6] simulation model used. Both variants of the DMAMAC protocol are evaluated and compared with the GinMAC [15] protocol in Sect. 4. In Sect. 5, we sum up the conclusions and discuss future work. In the rest of the paper we use DMAMAC to refer to our protocol in general and append either Hybrid or TDMA to designate the specific variant of the protocol in question. We assume that the reader is familiar with the basic concepts of MAC protocols.

2. RELATED WORK AND DMAMAC

In this section we briefly discuss related work, and describe the DMAMAC protocol with focus on the changes made to the previous version [13]. In related work, we focus specifically on the TDMA-based GinMAC protocol [15] as the design of DMAMAC is rooted in this protocol. In addition to GinMAC, there are several other related MAC protocols. Z-MAC [11] is a hybrid protocol using both CSMA and TDMA. Z-MAC is a distributed protocol with two-phase operation. In the setup phase the nodes make a list of their two-hop neighbors and then locally decide on a time slot such that no two nodes select the same time slot. Z-MAC uses CSMA for low-contention and TDMA for high contention period. The DMAMAC protocol is based on offline scheduling similar to GinMAC. This means that the entire time-slot is pre-planned and thus differs in operation from Z-MAC. WirelesHART [14] is the wireless successor of HART proposed for process monitoring applications. WirelessHART is a framework, with a combination of protocols for different functions. TSMP [10] is the MAC protocol proposed for wirelessHART, but it mainly handles time synchronization. The routing and slot allocation is handled by a separate unit called the network manager. The DMAMAC protocol can be used within the wirelessHART framework as a MAC protocol.

2.1 The GinMAC Protocol

GinMAC [15] was developed as a part of the GINSENG [7] project with requirements including reliable and timely delivery of data. The GinMAC protocol design is based on a network having a tree topology. Along with satisfying real-time requirements, GinMAC also addresses energy efficiency via efficient duty cycling. The GinMAC superframe forms the basis of the transient mode superframe in the DMAMAC protocol. The GinMAC protocol has the following main characteristic features.

Off-line Dimensioning. The network deployment is pre-planned based on application requirements, and scheduling decisions are made offline. A TDMA schedule is created with a given superframe length. This frame is then divided into three types of slots: *basic*, *additional*, and *unused.* The *basic* slots are for regular sensor data transfer, the *additional* slots are used to increase reliability, and the *unused* slots (or sleep slots) are used to achieve a low duty cycle. *Basic* slots are defined for both sensor and actuator data. *Configuration* commands include control data such as time synchronization. The sensor data is sent from the sensors to the sink, and the actuator data is sent from sink to the actuators. Given the tree topology, nodes having children need to have basic and additional slots allocated for its children as well.

Exclusive-TDMA. GinMAC is designed such that data transmission of maximum length and acknowledgement is accommodated within the same slot. GinMAC uses exclusive TDMA with no slot re-use across nodes. The GinMAC protocol has been designed for a sink that can manage a maximum of 25 nodes [15].

Delay Conform Reliability Control. Additional slots are used to ensure packet delivery thus increasing reliability. Prior to deployment, measurements are to be performed in the deployment area to assess the channel characteristics and calculate the worst-case link reliability. The number of additional slots used is based on the calculated worst-case link reliability.

2.2 The DMAMAC Protocol

The hybrid variant of the DMAMAC protocol was initially proposed in [13], and has been further refined in the context of this paper to suit the goals of the protocol design and for improvements in performance. In addition, we propose a pure TDMA-based variant of the protocol. The detailed description of the DMAMAC-Hybrid protocol can be found in [13]. In this paper, we discuss in brief the two operational modes of DMAMAC, the changes made with respect to the original Hybrid variant, and the pure TDMA-variant. Also, the delay from the state-switch being identified and the sink being notified is discussed. We rely on a tree topology similar to that of GinMAC [15], and DMAMAC also follows the offline scheduling and reliability mechanisms of GinMAC. The network architecture, for which the DMAMAC protocol is designed, consists of a sink node, sensor nodes, and actuator nodes. The sensor/actuator nodes are ranked according to their position in the tree topology, with nodes closest to the sink having the lowest rank. The sink node, responsible for managing the entire network is assumed to be wire powered and computationally powerful. Below, we list the key design considerations and assumptions related to the design of the DMAMAC protocol:

1. The protocol is designed for applications where the steady state is dominant.
2. For multi-variable process models, we design the transient mode operation to continue until the slowest of the input reaches its steady state.
3. The setting of thresholds for the sensors to detect the state-switch is assumed to be based on the underlying process model.
4. The sink is assumed to be able to reach all nodes in one hop. Notification messages from the sink are sent in one-way communication without ACK in one slot to all nodes in the network, and include control data such as state-switch data. Since the sink is wire powered it is reasonable to assume that it can afford to have longer radio range.
5. A small amount of packet failure is tolerated by the control system. Model Predictive Control (MPC) [8] or network-aware control systems are used to compensate for the possible packet losses.
6. Static network topology: no addition or removal of nodes during operation. In case of a topology change, schedules are recomputed accordingly.
7. A single slot accommodates both DATA and ACK packets.

2.2.1 *Transient mode*

The process changes rapidly during the transient state, and the transient mode is designed to meet the data requirements of this state. The transient superframe of DMAMAC is shown in Fig. 2 and is similar to the GinMAC superframe, but differs in the actuator data slot positions. The transient superframe is of length Nt (t for transient) slots and has smaller sleep duration compared to steady mode operation to increase data reliability. Both DMAMAC-Hybrid and DMAMAC-TDMA have the same transient superframe structure.

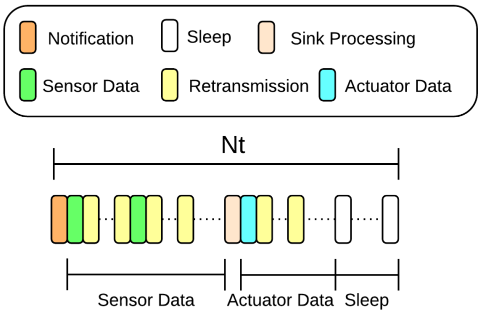

Figure 2: Transient mode superframe

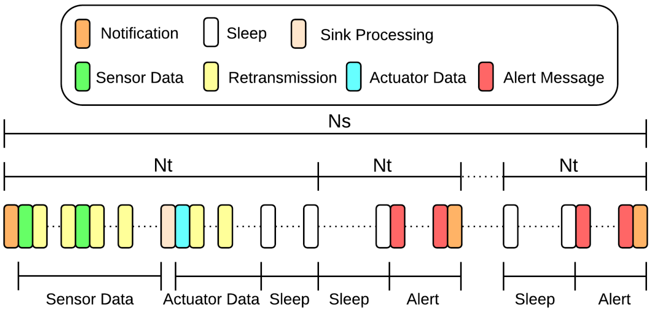

Figure 3: Steady mode superframe

2.2.2 *Steady mode*

During the steady state, the data-rate requirements of the controller is low, thus we also keep the communication of the sensed data low to save energy. The superframe structure shown in Fig. 3 for steady mode operation has Ns (s for steady) slots and is designed to be a multiple of Nt (number of slots in transient mode). Both DMAMAC-Hybrid and DMAMAC-TDMA have the same steady superframe structure, except for their alert slot scheduling which is discussed later with alert messages.

2.2.3 *State-switch delay*

The state-switch delay is the time interval between identification of a threshold breach by the sensor and the state-switch happening in the network, i.e., a change from the use of steady mode superframe to transient mode superframe. The steady superframe considered in this paper has been changed with respect to our previous proposal [13]. In the previous version of the Hybrid variant of the protocol, the alert slots were placed towards the end of the data communication part or the Nt long sleep parts. The notification slots were placed in the beginning of each Nt long part of the steady state superframe. Thus when an alert is detected, a notification is sent to the entire network in the next Nt part, the one after the Nt part where the alert is detected by a sensor, and is notified to the sink first. But, the change of superframe would happen in the Nt part that follows the Nt part with notification. This effectively implies a minimum state-switch delay of $2*Nt$ in the earlier version. With the current version, the minimum state-switch delay is reduced to Nt. Based on the current superframe structure, the state-switch delay is either two transient superframes

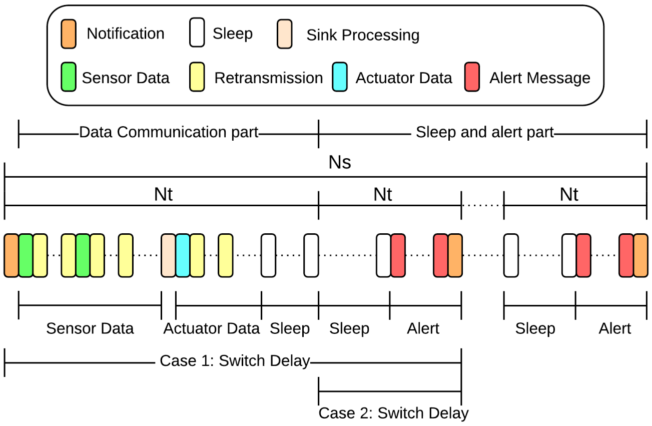

Figure 4: The two state-switch delay cases

(case 1) or one transient superframe (case 2) as illustrated in Fig. 4. The two cases are:

- Case 1: occurs if the alert is generated in the first part of the steady superframe (the data communication part). In this case, the state-switch happens after the second Nt part. Since the data is sent already in the first part, the real delay between two consecutive data communication part is still one transient superframe length.
- Case 2: is when the alert is generated in one of the sleep parts of the steady superframe, which will then have a state-switch delay of one transient superframe length.

Case 1 can be eliminated by adding alert and notification slots at the end of the data communication part. For case 2, we can further decompose the second part of steady superframe into a half or a quarter of the transient superframe length each ending with alert and notification slots. We have to note that these two cases represent the state-switch delay when there is no packet loss. The packet loss conditions where the state-switch attempt fails are discussed in Sect. 4.

2.2.4 Alert Messages

With respect to the previous design [13] we have updated the alert message part for the DMAMAC protocol. Now we have two methods to cater for alert messages. The first one is the existing alert message sending method proposed earlier for DMAMAC-Hybrid. The second is a method for the TDMA-variant of the protocol with alert message handling, where each node has a separate alert transmit slot in the alert period. The two methods are described here below.

DMAMAC-Hybrid alert method. As mentioned above, in [13] we proposed sending two alert messages in one slot to increase the probability of the alert messages reaching the next hop. The design consisted of one alert slot per rank in the network topology. Our initial simulation results showed that sending one alert message along with a Clear Channel Assessment (CCA) has a better performance than the former method. The main idea here is to have minimal number of alert slots whilst maintaining low probability ($< 1\%$) of state-switch failures. Given the network topology, we have one alert slot for each rank (level in the tree topology). We still maintain a random delay before sending the alert message but now with a duration within the interval $[0..(slotDuration - (\textit{Maximum time required to send alert message}))]$. The random delay along with CCA reduces the collisions (within the network), and thus reduces state-switch failures. Given the superframe structure collision is only possible in alert slots. Thus, collision is known to be a result of two nodes (at least) transmitting alert packets simultaneously.

DMAMAC-TDMA alert method. In DMAMAC-TDMA, each node has its own alert slot. This ensures no collision and reduces the possibility of switch failures. Switch failure is still possible due to packet loss on the wireless channel. The parent nodes send one alert based on either its own alert, or forwarding of an alert received from one of its children. Thus, the total number of alert slots is equal to the number of sensor nodes in the network. This method provides better reliability for alert messages to facilitate the state-switch from steady to transient which can be critical under consideration for the application.

2.3 Parameters and Design Alternatives

The DMAMAC protocol has several parameters that can be adjusted in accordance to the various trade-offs. These are listed below along with design alternatives:

1. The length of the sleep parts (currently Nt) in the steady superframe can be varied, which determines the maximum delay in state-switch, and also impacts energy savings.
2. The number of transient superframe length (Nt) parts to be used in the steady superframe, impacting energy savings.
3. The re-transmission count can be varied, impacting reliability. Re-transmission can be added into the alert slots as well.
4. Aggregation of data packets at the parent nodes. The order of the slots would then be: first the slots for the parent data packets and then slots for aggregated data from child nodes.
5. The number of alert slots could be increased to two slots for each rank for the Hybrid variant. This would increase the reliability of alert.

3. SIMULATION MODEL AND SETUP

For the evaluation of the DMAMAC protocol, we perform simulation based analysis. Based on the scenario considered in [13], a topology is designed for the simulation and is shown in Fig. 5. The topology has a total of 26 nodes including the sink. The numbers in Fig. 5 refer to node numbers, and numbers prefixed with "R" refer to rank in the tree topology. The topology can be configured in different ways. We use this representative topology to explore the key aspects of the DMAMAC protocol which are data communication, alert messages, multiple hops, and the possibility of collision (the latter only for the Hybrid variant). Different configurations of each protocol variant (TDMA and Hybrid) are used for evaluating the protocol, and the performance of the protocol on the considered topology is discussed later. In this section, we present the node positioning for the 25 nodes configuration managed by a single sink, and its scheduling. The topology consists of 19 sensor nodes, 6 actuator nodes, and 1 sink node. The tree topology has 3 ranks of node placements, with the most distant leaf node being 3 hops away from the sink. The nodes closest to the sink have the highest load in the network. Based on the setup, node 3 will have the highest data load among the three nodes with the highest rank. Given this load distribution and equal initial battery level on all nodes, node 3 is expected to run out of energy before any other node in the network. Thus for our experiment, we calculate network lifetime based on time to first node death.

Simulation is done using MiXiM [6] which is an OMNeT++ [16] based modeling framework designed for simulating wireless networks. We simulate DMAMAC in different configurations mainly with different probabilities of transient superframes appearing. We compare DMAMAC with the performance of GinMAC in a similar configuration. The simulation parameters are listed in Table 1. We use the radio parameters from the CC2420 datasheet [5], a radio unit frequently used in sensor and actuator nodes. The current consumed by the radio in different states Receive (RX), Sleep (Sleep), Transmit (TX), setup currents, and switch currents is defined using the data obtained from the

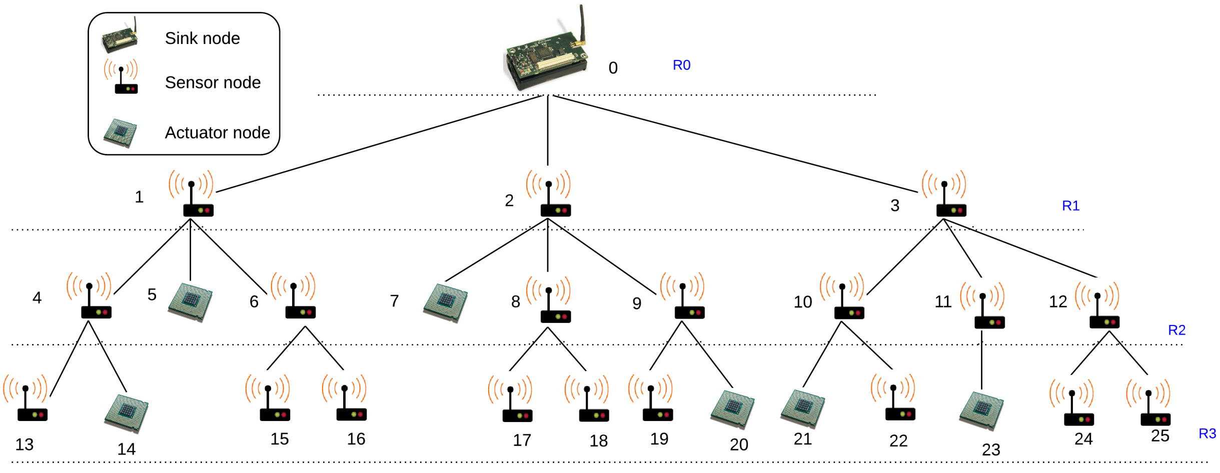

Figure 5: The logical topology for the simulation analysis

Parameter	Values
Number of Nodes	25 Nodes
Network dimensions(metres)	160 * 160
Simulation duration	1350 seconds
Transient Superframe length	1.5 seconds
Distance between nodes	15 to 25 meters
Number of rounds	900
MAC Protocols	DMAMAC-Hybrid DMAMAC-TDMA GinMAC
Radio Module	CC2420
Simulation repetition	100
State switch probability	10, 50
Superframe Ratio	2x, 3x, 4x
Data Packet size	44 bytes
Sink Packet size	11 bytes
ACK, Alert Packet size	11 bytes

Table 1: Simulation parameters

CC2420 datasheets. The time used to switch between radio states is also obtained from the CC2420 datasheets. We evaluate the protocol under ideal channel conditions.

An overview of the MiXiM simulation model used is shown in Fig. 6. In the MiXiM platform, we have created a project that contains C++ files, a NED package, XML files, packet description files, and a input parameters file. The C++ files are used to describe the MAC protocol for both the regular nodes and the sink node. The NED language in OMNeT++ is used to describe the topology, the connection between different modules and the hardware characteristics. Using NED files, we describe the regular nodes and the sink node along with their network interface cards (NIC). XML files are used as input files in the C++ files. The config XML file is for defining the path loss model used. The decider XML file describes the decider characteristics for the simulation. In this case we use the CC2420 decider for signal evaluation and demodulation [6]. Topology XML files gives the overall topology of the network describing the interconnection between different nodes. The different packets are: the regular MAC packet, Sink MAC packet, and the MAC packet used for Alert messages. These are all defined in separate files. The input parameters file defines values for the DMAMAC protocol parameters that can be varied to obtain different configurations. We refer to [6] and [16] for detailed information about designing simulation models in MiXiM.

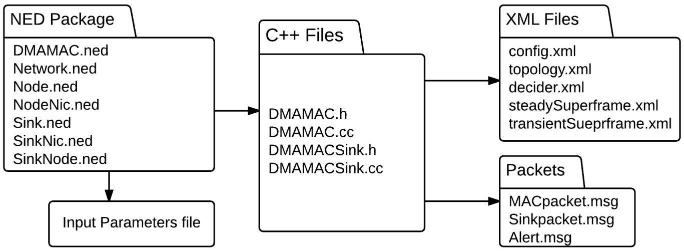

Figure 6: Overall structure of the MiXiM model

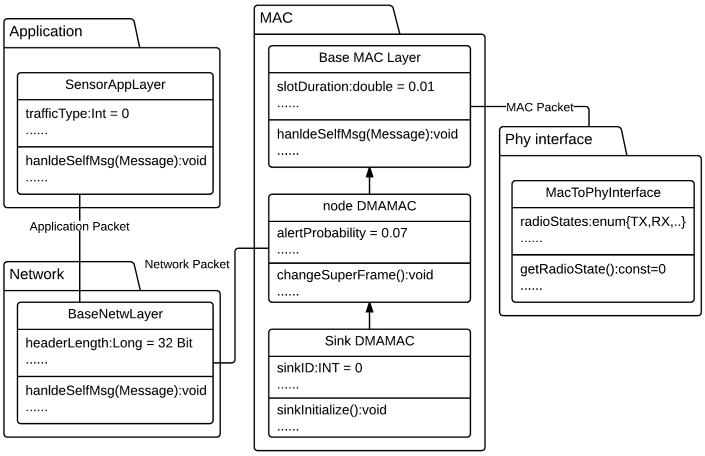

Figure 7: Overall architecture of the DMAMAC implementation

The overall architecture of the DMAMAC protocol implementation in C++ is shown in Fig. 7. We implemented the DMAMAC protocol for the nodes based on the inheritance from the BaseMacLayer from the MiXiM library. This is further inherited by the DMAMAC protocol for the sink (differs from nodes). We rely on the application and network layer from the MiXiM library defined in the input parameters file. The MAC protocol accesses the radio/physical layer via the interface provided in MiXiM,

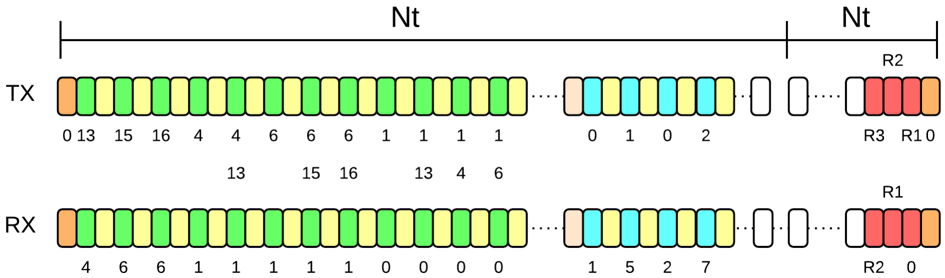

Figure 8: Slot schedule

and is used to switch between different radio states, mainly transmit (TX), receive (RX), and sleep (SLEEP).

The scheduling is done offline, and is given as input XML files separately for transient and steady superframe. A fragment of this is illustrated in Fig. 8. TX represents the transmitting node number and RX represents the receiving node number. The numbers prefixed with R represent the alert slot for a given rank in the tree topology, with leaf node rank represented by R3, and the lowest rank represented by R1. The sink has node identification 0. Notification messages are received by all nodes and hence there is no representation for the notification slot in RX. The second row of the numbers below the TX slots represents the child node, the data of which the parent transmits in the given slot. The sink node initiates the network wide state-switch, by sending a notification message to all nodes. The nodes then change the superframe upon receiving the state-switch notification using the appropriate XML input file describing the slot allocation schedule for the new state.

Configurations. We use three main configurations for the simulation study. They differ from each other on the probability of the transient state appearing in a given duration of time. The main configurations have transient state probability 10%, 50% and 100%. The state probability with 100% represents the GinMAC protocol and the other two represent the DMAMAC protocol. 10% transient superframes configuration represents processes that are stable through most of the process execution, 50% transient superframes appear in relatively less stable processes. These are further evaluated for multiple configurations of superframe length ratio (transient to steady superframe). Configurations with 2x represent steady superframe which is twice the length of transient superframe, 3x three times, and 4x four times. The ratio depends on the data rate required by the application in the steady state, the lower the data rate requirement, the higher the ratio can be used. All the considered configurations are listed in Table 2. The alert probability is the probability with which each sensor (not actuator) node generates an alert message. The alert probability was obtained using preliminary simulations in order to obtain frame distributions that correspond to state probability of 10% and 50%.

	[%] chance	Alert Probability		
Config.	Transient	DMAMAC [%] chance		Superframe
	Probability	Hybrid	TDMA	Multiplier
P-10-2x	10	1.00	1.20	2x
P-10-3x	10	0.80	1.00	3x
P-10-4x	10	0.70	0.70	4x
P-50-2x	50	11.80	11.00	2x
P-50-3x	50	9.10	10.50	3x
P-50-4x	50	8.20	10.00	4x
GinMAC	100	0	0	1x

Table 2: Parameter configurations considered for the evaluation

4. PERFORMANCE EVALUATION

We begin with the discussion of frame distribution, which is the ratio of transient superframes to steady superframes. This ratio in turn determines the ratio betwen transient states and steady states of the network within the simulation. Then, we discuss the energy performance of the two variants of the DMAMAC protocol in comparison with GinMAC protocol via total energy consumption and network lifetime metrics. Further, given the random nature of possible state-switch requests in DMAMAC-Hybrid, and the CSMA based method of transmission, alert messages could suffer collision resulting in state-switch failures, which is critical to the operation of the control system. Thus, state-switch failure is discussed in detail.

4.1 Frame Distribution

We investigate the reduction in energy consumption when the fraction of transient superframe is below 10% and also at 50% to give a broader evaluation. In Table 3, we detail the frame distributions considered. For the given simulation time, we have set the alert probability and the transient state probability such that it yields a transient state percentage corresponding to our desired configuration. Table 3 lists the obtained frame distribution across 100 runs. GinMAC has a total of 900 superframes for the simulation duration, and the transient superframes in DMAMAC are measured relative to 900 possible transient superframes for the same duration. The Average column gives the average number of superframes across the 100 runs. The alert probability of each node is independent to that of the other nodes. It was therefore required to estimate the correct alert probability and state probability combination in order to obtain the desired frame distribution. We conducted several simulation runs to obtain proper alert probability. Note that the

	Transient Superframes			
	DMAMAC-Hybrid		DMAMAC-TDMA	
Config.	Average	[%] of total	Average	[%] of total
P-10-2x	87.94	9.77	88.37	9.82
P-10-3x	88.36	9.82	88.28	9.81
P-10-4x	88.28	9.81	88.43	9.83
P-50-2x	441.41	49.05	441.61	48.96
P-50-3x	446.26	49.58	446.04	49.56
P-50-4x	449.03	49.89	448.33	49.81

Table 3: Frame distribution across 100 runs

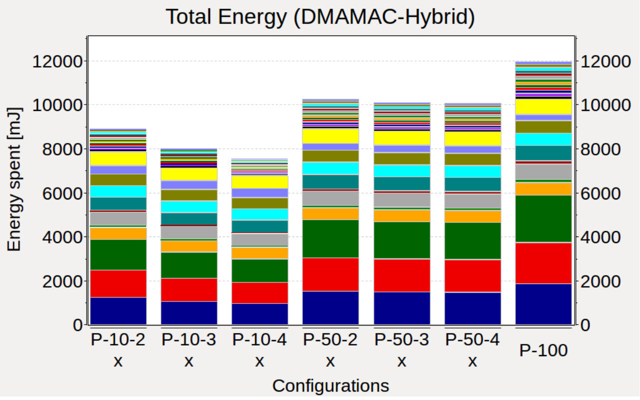

Figure 9: Energy consumption DMAMAC-Hybrid

transient state probability and the alert probability are two different probabilities. Transient state probability applies only for the state-switch from transient to steady which is done by the sink.

4.2 Energy Consumption

The DMAMAC protocol is designed for a tree topology which means that the energy consumption is not uniform across all nodes as discussed in Sect. 3. Thus, we consider *total energy* as the metric to measure and compare total energy consumption of the DMAMAC protocol and the GinMAC protocol. Total energy is the energy spent by all the nodes in the network (except the sink which is assumed to be wire powered) during the entire simulation. We compare all configurations, and variations of the DMAMAC protocol (Hybrid and TDMA) with GinMAC. Firstly, we present the total energy consumption across various configurations of the two variants DMAMAC-Hybrid and DMAMAC-TDMA. Graphs depicting the results are shown in Fig. 9 and Fig. 10. Note that, in these figures, the colors across each configuration represent different nodes. This is to highlight the difference in energy consumption among the nodes. Also, Table 4 lists the numbers for energy consumption for both the variants of the DMAMAC protocol. The comparison column (Relative) in the table represents energy spent using DMAMAC protocol (E_{DMAMAC}) in percentage of energy spent using GinMAC protocol (E_{GinMAC}). The total energy consumption in the network using the GinMAC protocol is $E_{GinMAC} = 11,950\text{mJ}$ or 11.95J. The average value of the energy consumed over 100 runs is used for the comparison.

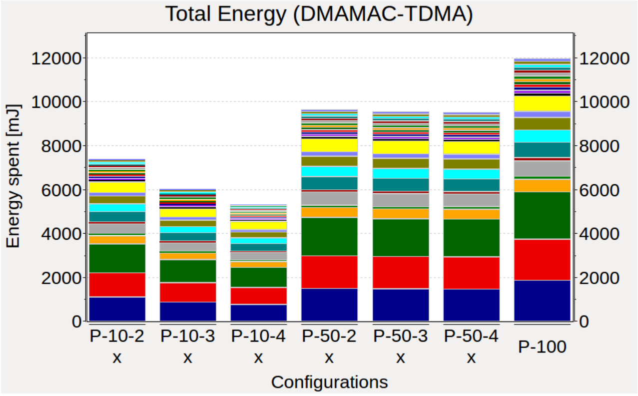

Figure 10: Energy consumption DMAMAC-TDMA

	DMAMAC-Hybrid		DMAMAC-TDMA	
Config.	Avg.[J]	Relative [%]	Avg.[J]	Relative [%]
P-10-2x	8.90	74.48	7.39	61.84
P-10-3x	7.99	66.86	6.00	50.20
P-10-4x	7.54	63.10	5.32	44.52
P-50-2x	10.25	85.77	9.65	80.75
P-50-3x	10.11	84.60	9.53	79.75
P-50-4x	10.06	84.18	9.5	79.50

Table 4: Comparing total energy consumption

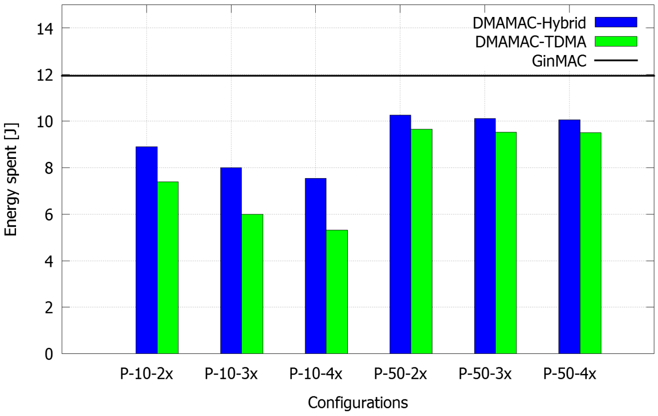

Figure 11: Comparing both variants of DMAMAC with GinMAC for total energy consumption

The graph in Fig. 11 shows the comparison for total energy consumption. The energy consumption of the network using the GinMAC protocol is represented as a black line in the graph. The energy consumption for the GinMAC protocol is obtained for a single configuration and a single run. Given the static nature of the GinMAC superframe (no random elements like alert message) running multiple simulations does not yield any difference. Also, the GinMAC superframe is fixed for a given network, and cannot be extended/modified as in the case of the DMAMAC transient superframe. Thus only one configuration is possible with GinMAC. We can observe that DMAMAC-TDMA is the most energy-efficient of all protocols compared across all configurations tested for the DMAMAC protocol. This is due to energy consumption on nodes that have to be awake for the entire alert slot duration in DMAMAC-Hybrid. Given the different configurations, it is possible to adapt the protocol design based on the application requirement. The length of the steady superframe can be varied to obtain higher energy efficiency or higher data rate (smaller steady superframe). Also, note that the results of 50% transient operation is primarily shown to get a wider perspective on results obtained. In general DMAMAC is aimed at serving applications where steady state is dominant ($\geq 90\%$).

4.3 Network Lifetime

The network lifetime is a measure of the survivability of the network on a single battery charge. This can be viewed in several different ways, one of which is time to first node death. Using this approach, we have evaluated GinMAC and DMAMAC. The graphs in Fig. 12 and Fig. 13 show the time to first node death for the two variants of the DMAMAC protocol across different configurations. The x-axis gives the relative time since the evaluation is based

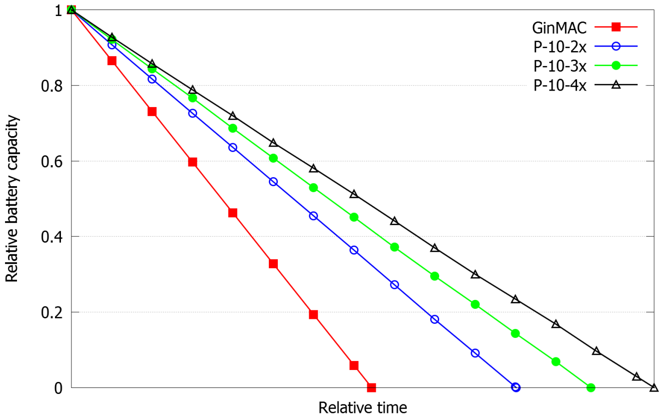

Figure 12: Network Life Time (DMAMAC-Hybrid)

on the initial battery capacity which may vary depending on the particular battery used on the nodes. Node 3 (see Fig. 5), is the node that dies first due to its position in the tree topology which causes it to have the highest number of ancestors in the network. The simulation was conducted with a fixed initial battery capacity, and run until node 3 had depleted all its battery. Given the focus on the 10% transient configuration we have skipped 50% configurations in these graphs.

A comparison between the two variants of the DMAMAC protocol and GinMAC is shown in Fig. 14. It shows the relative time to death for node 3 across different configurations. The time to death for node 3 with GinMAC is represented with a black line on the graph. In Table 5, we list all simulation configurations and the energy consumption for node 3 obtained from the simulation. The average energy consumption of the node is presented in the table. Among all the explored configurations, the 4x configuration of the TDMA-variant of DMAMAC is the most energy efficient. Also, overall the TDMA-variant fares well in comparison with GinMAC and the Hybrid-variant. It is also important to note that the time from an alert being generated and the state-switch happening is the same in all these configurations as discussed in Sect. 2.

4.4 State-switch Failures

The switch of operational modes is an integral part of the DMAMAC protocol, and particularly the switch from steady to transient mode is critical. The aim of an ideal MAC protocol is to ensure that this switch happens whenever alert messages are generated or basically when the process moves to the transient state.

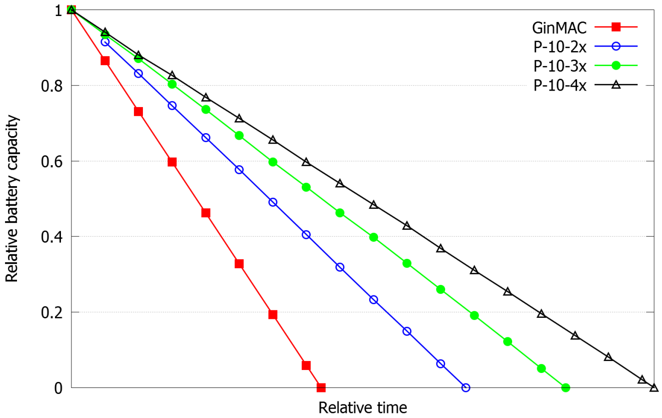

Figure 13: Network Life Time (DMAMAC-TDMA)

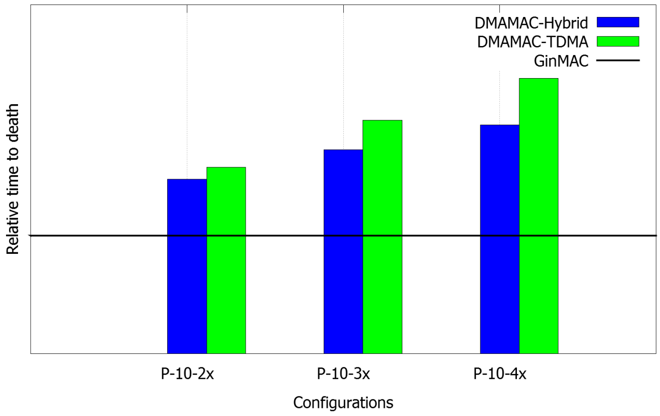

Figure 14: Time to death comparison between different protocols

Energy spent by Node 3 on GinMAC = 2.16 J				
	DMAMAC-Hybrid		DMAMAC-TDMA	
Config.	Avg. [J]	Relative [%]	Avg. [J]	Relative [%]
P-10-2x	1.40	64.94	1.30	60.30
P-10-3x	1.18	54.73	1.04	48.24
P-10-4x	1.06	49.17	0.91	42.21
P-50-2x	1.73	80.24	1.74	80.71
P-50-3x	1.70	78.85	1.72	79.78
P-50-4x	1.69	78.39	1.71	79.31

Table 5: Node 3 energy consumption

4.4.1 *DMAMAC-Hybrid*

In the DMAMAC-Hybrid variant, collision is inevitable with the use of CSMA in the alert slots. These collisions could result in state-switch failures when the alert packet is lost due to collision, and the sink is hence not notified of alerts. Alert messages exhibit random behavior, i.e., it is possible that several sensors detect the change of process states. Further, when two or more sensors send the data towards the sink, collision could occur if their parent node is the same, and both have comparably the same random delay. This results in the alert message being lost.

In Table 6, we present the results for state-switch failures for different configurations. Also, a graph representation of the switch-attempt failure in comparison to the collisions and the total number of switches is shown in Fig. 15. The aim of our DMAMAC protocol is to keep the state-switch failure within [0-1]%, mainly for the configurations of P-10 (2x,3x and 4x). From the results presented, considering the configurations P-10-2x, P-10-3x and P-10-4x, the state-

Config.	Failed Switches (avg)	Successful Switches (avg)	Relative [%]
P-10-2x	0.23	76.13	0.30
P-10-3x	0.15	76.17	0.12
P-10-4x	0.07	76.62	0.10
P-50-2x	6.73	199.14	3.38
P-50-3x	22.20	199.52	11.18
P-50-4x	21.77	196.17	11.10

Table 6: State-switch attempt failures

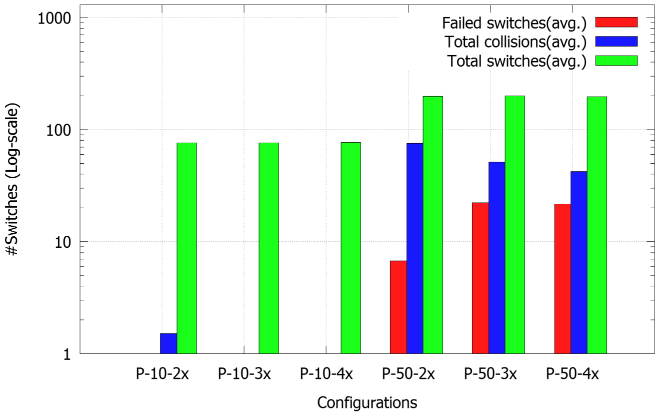

Figure 15: Switch failure for 25 nodes

switch failure is at most 0.23 % (i.e., within [0-1]%) or 2.3 failures in 1000 state-switches (on average). An alternative method to prevent state-switch failures could be proper identification of collisions, as collision of actual alert packets and not random noise, and considering them as alert.

Even if a state-switch attempt fails, the state-switch would eventually happen since nodes can store the information about a threshold being violated, and then notify the sink in the next alert. But this goes through the same process, and has the same probability of state-switch failure. Given the low probability of state-switch failure obtained, the expected state-switch delay is low. Assuming that the probability of failure is independent, the number of expected alert messages before switch can happen equals $\frac{1}{0.9977} = 1.0023$ (based on the 0.23% of 10-2x configuration). Hence we would expect that the state-switch happens in the immediate next alert at worst. The configurations with 50% transient state have much higher failed switches, and are presented here for a broader view of the performance.

4.4.2 DMAMAC-TDMA

The TDMA-variant of the DMAMAC protocol eliminates the collision possibility, thus in ideal conditions switch failure does not exist. But for packet failure conditions in a real wireless channel, alert messages must be used along with re-transmission similar to data transmission part in the DMAMAC protocol. This introduces an extra energy expenditure into the existing energy consumption, but can be used for increasing reliability of switches.

4.5 Packet Transmission Delay

The end-to-end (sensor-to-actuator) transmission delay depends on the superframe structure, since it is of static nature. For the considered topology, shown in Fig. 5 consisting of 25 nodes with 1 re-transmission slot for each sensor/actuator data slot, the maximum end-to-end transmission delay is 1200ms (1.2s). This is the time interval between the first sensor sending the data until the last actuator receives the data. We have a lower delay compared to the GinMAC superframe where the actuator slots are placed towards the end of the superframe [15]. But this is a design choice and can be appropriately modified in GinMAC as well. Alternatively, the delay can be reduced or increased by changing the number of re-transmission slots used in transient and steady mode.

4.6 Maintaining Reliability on lossy links

In the DMAMAC protocol, the re-transmission slots in general are based on the re-transmission slots in GinMAC. But given that we have a dual-mode protocol, we can vary the re-transmission slots in transient and steady mode such that they are different. This could in general be done to further increase energy efficiency, i.e. relatively lower number of re-transmission slots in steady than in transient mode. Alternatively, a larger number of re-transmission slots in the steady state could be considered to make sure the data reaches the actuator with a higher probability, but at a higher energy cost. In the current study, we used the same number of re-transmission slots for both GinMAC and DMAMAC protocol, and the same number for both operational modes of the DMAMAC protocol. The re-transmission slots are used in the simulation to give an idea of possible delay caused due to the usage of re-transmission slots. There is no difference between the performances of the two protocols in terms of reliability. Concerning the sink's single notification slot used, the loss of a notification packet can cause a difference in operational modes between different nodes and sink. In general this can be prevented by using dedicated repeaters to assist sink signals in reaching the nodes correctly.

4.7 Scalability

Scalability of MAC protocols is a general challenge in WSN and WSAN. For the DMAMAC protocol, we suggest a maximum of 25 nodes similar to GinMAC for a single sink to keep the delay low for process monitoring and control applications. In case a larger number of nodes are required, a backbone connecting multiple sub-networks of 25-nodes each managed by a separate sink is suggested. The backbone may use powerful high-rate data transfer (wired or wireless). This can address the scalability issue to some extent. In principle, larger networks can be managed in case the delay requirements permit it. The more the energy spent in transient state, the better is the relative energy efficiency of the DMAMAC protocol (both variants) than that of GinMAC. This indicates that the increase in packet size, number of nodes in the network, lower-bit rate wireless nodes, could result in larger energy savings with the use of the DMAMAC protocol. From this perspective, the DMAMAC protocol can be considered as scalable.

4.8 Hybrid and TDMA

On an ending note of the performance evaluation, we can observe that DMAMAC-TDMA has better energy efficiency among the two variants proposed. We preserve the representation of the DMAMAC-Hybrid to encourage future investigations to use its hybrid nature effectively. In particular for larger networks where the topology is horizontal and the alerts are less frequent, the hybrid variant may have better energy efficiency than TDMA. But it is still prone to collisions, and switch delay issues resulting from collision.

5. CONCLUSIONS AND FUTURE WORK

The DMAMAC protocol is aimed at satisfying real-time requirements of process control systems while preserving energy to prolong network lifetime. The DMAMAC protocol design has been improved in comparison to the previous version based on initial simulation results and analysis. We simulated both variants of the DMAMAC protocol using the OMNeT++ platform in conjunction with MiXiM libraries for wireless networks to evaluate the performance of the protocol. The DMAMAC protocol shows considerable reduction in energy consumptions compared to the GinMAC protocol for process control systems with dominating steady state. In particular, the DMAMAC-TDMA exhibits the lowest energy consumption among all. We mainly compared the DMAMAC-Hybrid,DMAMAC-TDMA, and the GinMAC protocols based on metrics of total energy and network lifetime. For network lifetime, we considered time to first node death. Given the random nature of alert messages in DMAMAC-Hybrid, we have also evaluated this protocol variant for state-switch failures to give a better idea of directions for further study. The state-switch failures are $\leq$ 0.23% for configurations of 10% transient superframes which is tolerable. The study of varying re-transmission slots in steady and transient superframes could add on to the simulation results.

The DMAMAC protocol is an application specific protocol proposed for process monitoring and control applications. The idea can be applied across domains for similar monitoring and control applications that have event-based traffic conditions. Two possible application domains are monitoring and control in healthcare and home automation. Given the nature of applications in these domains and the number of nodes used, the requirements match the design constraints of the DMAMAC protocol. Given the requirements and challenges existing in Wireless Body Area Networks (WBAN) [9], DMAMAC protocol could be used to address some of the challenges along with prime focus on energy efficiency. In WBAN, mainly two types of data are handled: periodic monitoring data and emergency event based data [2]. The dual mode operation allows for facilitating high data rate communication for priority traffic during emergencies and otherwise using low data rate to facilitate energy efficiency prolonging the lifetime and reducing the cost. The applications in the healthcare domain could particularly benefit from the reliability and energy efficiency provided by the TDMA-variant. One such application is the sensor-actuator implementation to maintain glucose level in a diabetes patient, which consists of glucose sensors and insulin actuators. Adaptation for non-critical parts of home automation would be easier considering that the application is far less critical, and thus switch delay requirements would not be as stringent. Such applications could benefit from the use of either variants of the DMAMAC protocol. This could include temperature management systems. But critical or emergency systems including fire-safety and theft-safety in home automation could benefit from the TDMA-variant. Thus, in general the dual-mode operation and other features of DMAMAC protocol can be applicable across domains in internet of things [12], but would require adaptation based on requirements.

Near future work in the development of DMAMAC is to create an implementation of DMAMAC and perform deployment testing. This would further validate the applicability of the protocol in real process control systems and also, provide insights on the differences between the results obtained from simulation and implementation.

6. REFERENCES

[1] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367, 2004.

[2] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688 – 2710, 2010.

[3] A. Bachir, M. Dohler, T. Watteyne, and K. Leung. MAC essentials for wireless sensor networks. *IEEE Comm. Surveys & Tutorials*, 12(2):222–248, 2010.

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *IEEE Proceedings*, 95(1):138–162, 2007.

[5] T. Instruments. Chipcon CC2420 datasheet. 2007.

[6] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In *Proc. of SIMUTOOLS*, pages 1–8, 2008.

[7] T. O Donovan, J. Brown, U. Roedig, C. J. Sreenan, J. do O, A. Dunkels, A. Klein, J. Silva, V. Vassiliou, and L. Wolf. GINSENG: Performance control in wireless sensor networks. In *Proc. of SECON*, 2010.

[8] A. Onat, T. Naskali, E. Parlakay, and O. Mutluer. Control over imperfect networks: Model-based predictive networked control systems. *IEEE Transactions on Industrial Electronics*, 2011.

[9] M. Patel and J. Wang. Applications, challenges, and prospective in emerging body area networking technologies. *IEEE Wireless Communications*, 17(1):80–88, February 2010.

[10] K. Pister and L. Doherty. Tsmp: Time synchronized mesh protocol. *IASTED Distributed Sensor Networks*, pages 391–398, 2008.

[11] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-mac: A hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, June 2008.

[12] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung. A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6):91–98, December 2013.

[13] A. A. K. Somappa, K. Øvsthus, and L. M. Kristensen. Towards a dual-mode adaptive MAC protocol (DMA-MAC) for feedback-based networked control systems. *Procedia Computer Science*, 34(0):505–510, 2014. ComSense 2014.

[14] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, and M. Nixon. Wirelesshart: Applying wireless technology in real-time industrial process control. In *IEEE Real-Time and Embedded Technology and Applications Symposium, 2008*, pages 377–386, April 2008.

[15] P. Suriyachai, J. Brown, and U. Roedig. Time-critical data delivery in wireless sensor networks. In *Proc. of DCOSS*, pages 216–229, 2010.

[16] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Proc. of SIMUTOOLS*, pages 1–10, 2008.