Medium Access Control for Wireless Networks with Diverse Time and Safety Real-Time Requirements

Pablo Gutiérrez Peón*†, Elisabeth Uhlemann†, Wilfried Steiner*, and Mats Björkman†
*TTTech Computertechnik AG, Vienna, Austria
†School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
Email: {pablo.gutierrez-peon, wilfried.steiner}@tttech.com, {elisabeth.uhlemann, mats.bjorkman}@mdh.se

***Abstract*—The communication in-between embedded systems present in cars and planes, requires real-time networks. Up to now, fieldbus technologies like PROFIBUS and CAN have covered the demand for predictable communications in embedded systems. However, these fieldbuses do not suit some of the emerging application domains, that need more flexibility, support for dynamic traffic flows, different traffic classes, high throughput, and the inclusion of wireless capabilities. To this end, we propose several different medium access control (MAC) schemes with support for traffic with diverse time and safety requirements. We have calculated the worst case channel access delay for each proposal, and also simulated them in OMNeT++ to analyse and compare their performance in terms of average access delay and packet collisions as a function of different protocol settings and traffic patterns e.g., the channel load, data traffic emerging from one sender only versus evenly distributed between all senders. Our results indicate that the more that is known about the data traffic, the better performance can be achieved by selecting an appropriate MAC protocol. Conversely, when nothing is known, one MAC protocol emerges as the best trade-off.**

I. Introduction

In real-time networks, a guarantee to not exceed a certain maximum delay is given. Providing an upper bounded channel access delay is therefore one of the main tasks of the medium access control (MAC) layer in real-time networks. To accomplish this, the MAC layer should assure that a transmission through the shared medium will be done within a limited amount of time, regardless of the amount of data requested to be sent by other participants in the network. One of the most commonly found MAC protocols in wireless networks is carrier sense multiple access with collision avoidance (CSMA/CA). It is used as part of the widely known standards IEEE 802.11 for LAN and IEEE 802.15.4 for wireless personal area networks (WPANs), with the benefit that it is not too complex to implement, and in lightly loaded settings it can achieve good throughput. However, CSMA does not provide a predictable channel access delay since access to the medium is random and packets can collide. Consequently, several mechanisms have been designed to provide channel access in a bounded amount of time with CSMA. In [1], the authors use polling, i.e., triggering response transmissions upon reception of a special (polling) message, usually sent by a central coordinator. This method is, however, not very bandwidth efficient. In [2], the authors present VTP-CSMA, a token passing approach for IEEE 802.11. The token passing techniques provide access to the medium upon reception of a token message that circulates among the nodes, normally in a round-robin fashion. However, token protocols typically suffer from jitter, which can be problematic for periodic data traffic. In contrast, in time division multiple access (TDMA) techniques, where access is based on time-slots, the jitter can be negligible. TDMA can be found in protocols like IsoMac [3], and RT-WiFi [4]. In both cases, a centralized controller is in charge of assigning the time-slot access opportunities, resulting in a single-point-of-failure.

The demand for real-time communications in embedded systems has, for many years, been covered by fieldbus technologies like PROFIBUS or CAN. However, their quite limited throughput and lack of flexibility have intensified the efforts towards faster, cheaper and more flexible standardized solutions. Furthermore, many emerging applications require support for both time-triggered and event-driven data traffic and demand wireless capabilities to connect embedded systems due to ease and flexibility in deployment and reduced wiring costs. An example of a hybrid wired-wireless network with support for time-triggered data traffic is the HART/WirelessHART protocol [5]. However, even if a subset of all time-slots can be shared dynamically, WirelessHART still only provides deterministic delay to one type of data traffic, namely time-triggered periodic messages. In the other end, there are some recent work on wireless MAC protocols with support for event-driven real-time traffic, namely WirArb [6] and PriorityMAC [7]. Still, these protocols basically target either time-triggered or priority-based event-driven data traffic, not integrated levels of both. The work in [8] do provide support for both time-triggered and event-driven data traffic in IEEE 802.11 by traffic prioritization and an offline TDMA slot assignment, but guarantees are still given only with restrictions on the channel load as well as the periodicity of real-time traffic. Time-Triggered Ethernet (TTE) [9], is a communication technology that allows the integration of time-triggered and event-based rate-constrained data traffic. TTE is also compatible with Ethernet, meaning that legacy best-effort Ethernet traffic is also supported. Due to this, TTE emerges as a tractable option to be used in real-time networks in emerging application domains. Unfortunately, TTE does not have wireless capabilities.

To this end, we proposed and evaluated several different wireless MAC protocols supporting the same type of traffic classes as TTE, namely time- triggered (TT), rate-constrained (RC) and best-effort (BE), and operating on top of IEEE 802.11, rather than the lower rate protocol IEEE 802.15.4 that WirelessHART is based upon [10]. TT and RC traffic were given predictable channel access delays by assignment of pre-scheduled time-slots, whereas the remaining time was made available for event-driven traffic. The proposed MAC schemes where evaluated and compared in terms of how they handle event-driven traffic. A comparative study considering aspects such as delay, reliability and efficiency was made. However, the channel access delay was only evaluated in terms of the worst case. In this paper, we not only refine and improve the three MAC proposals, but we also evaluate the maximum and the average channel access delay for different data traffic patterns. The MAC proposals have been implemented in OMNeT++ [11], to thoroughly evaluate and differentiate their performance in terms of average channel access delay and number of collisions, as a function of different data traffic patterns and loads, e.g., the ratio of time-triggered traffic, the channel load, data traffic emerging from one sender only versus evenly distributed between all senders. Our results indicate that the more that is known about the BE traffic, or alternatively, the event-driven traffic, the better performance can be achieved by selection of a proper MAC protocol. Conversely, when nothing is known, one MAC protocol emerges as the best trade-off.

The reminder of the paper is structured as follows. In Section II, we present our wireless MAC protocol proposals coping with diverse time and safety requirements together with an analytical evaluation of the worst case access delay. Section III describes the simulator, and the results obtained from the simulation. Finally, we conclude in Section IV.

II. Wireless MAC Proposals Suitable for Diverse Time and Safety Real-Time Traffic Requirements

The design of our wireless MAC method is focused on guaranteeing a bounded access delay to the wireless medium, in the context of a hybrid wired-wireless network that supports three traffic classes: TT, RC and BE.

A. Hybrid network topology

We consider TTE a good candidate for being the core technology in the wired segment, since it provides support for diverse time and safety traffic requirements, including standard Ethernet. On the wireless side, we have selected IEEE 802.11 over other standard technologies, due to the high-speed data rates, and also its similarity to Ethernet.

The architecture of TTE is based on switched Ethernet, with networks comprised by end-systems and switches. Every end-system is connected to a switch through a full-duplex link, conforming a star topology. Switches are not only restricted to connect end-systems, but can also be connected to each other, so the network topology becomes a star of stars, commonly referred to as snowflake topology. On the wireless side, we also adopt a star topology using the IEEE 802.11 infrastructure topology. This becomes beneficial for the integration between the wired and wireless segments, considering that it does not add more complexity to the task of scheduling the traffic. In a star topology, the wireless end-systems communicate through the access point (AP), that is also responsible for interfacing the wired segment. An important difference between the wired and the wireless segments comes by the collision domains they define (Figure 1). A collision domain is a section of the network in where devices cannot transmit at the same time because their transmissions would overlap. In full-duplex links, each link defines two collision domains, one for sending and one for receiving. In half-duplex links, there is only one collision domain, for both sending and receiving. In wireless networks, the particularity is that all devices in range constitute a single collision domain, also referred as broadcast domain. Therefore, all wireless end-systems should be properly installed to be in the range of the corresponding AP. To avoid interferences between wireless collision domains, overlapping areas should either use different frequencies or coordinate transmissions using a wireless MAC with support for real-time traffic. The MAC protocols considered here are intended for a single hop network, where the access point acts as either receiver or as transmitter.

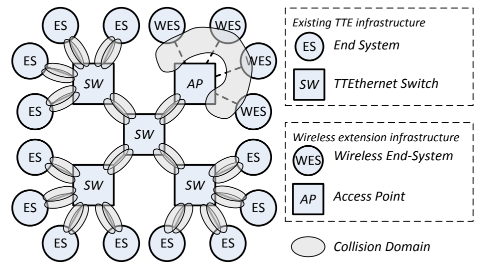

Fig. 1. Example of the proposed wired-wireless scenario, with switches (SW) and access points (AP) connecting the end-systems (ES, WES) forming recursive star topologies (snowflake topology). Note that every wired connection forms two collision domains, while the wireless systems constitute a single collision domain.

B. Real-time traffic management

The support for applications with diverse time and safety requirements in TTE is made by means of three different traffic classes. The TT traffic class is based on the time-triggered communication paradigm, with offline scheduled traffic based on the user requirements that specify the periods of the traffic flows. Applications making use of TT flows are provided with real-time capabilities, with a bounded time to access the medium. This bounded time is equal to the period of the flow, since the worst case comes when the application just missed the current message slot and has to wait for the next one. The second traffic class with real-time capabilities is RC. This class provides a guarantee given a certain minimum inter-arrival time for the messages. In the extreme case of having something to transmit at a rate equal to the inter-arrival time, the traffic flow behaves as a periodic TT flow. The difference between RC when compared to TT is that the network does not force periodic slots, but allows longer periods between messages if needed via online allocation of RC on the time left by TT. The RC traffic class is a perfect fit for e.g., audio and video streaming applications. Apart from the real-time support, an additional class named BE gives support for non-real-time legacy Ethernet traffic. BE traffic can occupy the free time left by TT or RC traffic, and our goal in this paper is to be able to provide an upper bound on the channel access delay also for this traffic class, such that it could be used to support event-driven real-time data traffic.

C. Scheduling

In order to provide real-time capabilities to the traffic flows sharing the same network infrastructure, all transmissions of real-time traffic flows must be done according to a schedule, that is known by all the network participants. In the simplest scenario, a network is composed by one switch and several end-systems that are connected to the switch. The problem of traffic scheduling becomes quite complex when dealing with topologies that account more than one switch, like in multi-hop switched networks. TTE solves this complex scheduling problem with a mathematical model of the network topology and the traffic, based on first-order logic constraints [12]. These constraints address the mutual exclusion of the dataflow links (contention-freedom), and others like having received the data on one hop before sending it again (path-dependencies). To obtain a schedule, the mathematical model is solved using satisfiability modulo theories (SMT) solvers [13].

The TTE scheduler is the tool we use to perform the integration of wired and wireless segments at traffic level for real-time (TT and RC) flows. For this, we need to extend the TTE scheduler in order to give support to the traffic traversing the wireless segment. This can be done through the definition of new first-order logic constrains that reflect the particularities of the wireless medium. Specifically, it is necessary to define a contention-free constraint that models the broadcast nature of the wireless medium, that is, it does not allow concurrent transmissions on the links included on each of the wireless collision domains. An optional constraint for the scheduler can model the differences in the transmission speed between the wired and wireless medium. In case this constraint is not defined, the slot size, that should be big enough to accommodate the transmissions plus any kind of protocol overhead, will be the same for the entire network, and adapted to the slowest transmission medium. The obtained schedule has a duration of the so-called hyperperiod.

D. Enabling event-driven real-time data

The time left after the allocation of the TT and RC traffic flows is available for BE traffic. Since BE is a non-real-time type of traffic, TTE does not provide any guarantee of delivery. When an application wants to send BE packets, these are enqueued in end-systems and switches and sent whenever the link is not used for TT or RC traffic. Due to its random generation pattern, BE can overflow the queues causing packets to be dropped. The problem can worsen in wireless networks with reduced bandwidth given by the half-duplex links and lowered robustness. To mitigate this, we propose three different wireless MAC protocols (Figure 2) that extend the ones proposed in our previous paper [10].

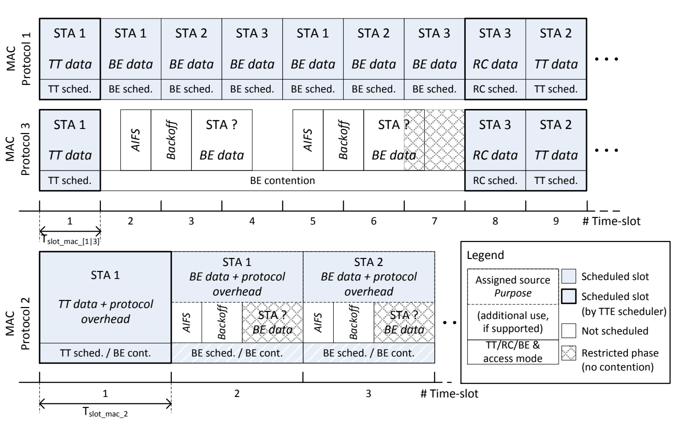

Fig. 2. Traffic scenario example for the proposed wireless MAC protocols.

MAC #1. Pre-scheduled time-slots: The time available for BE is divided into time-slots that are pre-assigned to the nodes using a round-robin schedule. If a node wants to send a BE packet, it has to wait until the round-robin assigned BE slot comes. As there is no other protocol overhead, each slot has to accommodate only one transmission together with some small guard time, and thus the size of the slot can approximately be equal to the transmission time. Given a set of slots reserved for TT and RC data traffic S, a certain number of nodes N, and the transmission time T_{transm}, yields a worst channel access delay which is bounded and equal to $T_{BE\_worst\_delay_1} = (S + N - 1)T_{transm}$.

MAC #2. Contention-based time-slots: In MAC Protocol 1, if a node does not have anything to send in its assigned time-slot, the slot is left unused. In MAC 2 if the slot is not used by the assigned node, the rest of the nodes can try to get access through a contention process whenever they have something to send. To implement this, all nodes perform channel sensing, but with one prioritized node using a shorter AIFS than the rest. In case the prioritized node does not have anything to send, the rest of the nodes will notice that the channel is still idle, since they sense the channel for a larger AIFS. To decrease the probability of collisions, the non-prioritized nodes wait for an additional random time (backoff), selected from a contention window (CW). The size of the slot in this proposal must therefore at least be big enough to fit a transmission and the larger AIFS, plus a guard interval. Alternatively, the slot can be made even bigger so that more values from the CW can fit, thereby reducing the likelihood of the slot remaining empty even further. The shortest possible slot (MAC 2A) only allows a transmission from the node having prioritized access, or alternatively from any other node that randomized a backoff value of zero. Note that several non-prioritized nodes can randomize the same backoff value, so the approach increases the chance of collisions. If the slot is bigger (MAC 2B), more values from the CW can fit, and therefore more chances are given to the non-prioritized nodes (but fewer slots can be allocated in total). IEEE 802.11e defines different AIFS and CW values to provide four priority levels aimed to support different types of applications. For the prioritized node we use the AIFS corresponding to voice applications (AC_VO), and for the rest of the nodes, the AIFS for best effort applications (AC_BE). When having a CW, we have chosen the CW size used in video applications (AC_VI), since the CW then had enough levels to reduce collisions without increasing the maximum delay too much. Note that the size of the CW does not increase after each collision, since we do not incorporate retransmissions. For Protocol 2, the worst channel access delay then becomes $T_{BE\_worst\_delay_2} = (S + N - 1)(T_{transm} + T_{AIFS} + T_{CW})$ since the worst case is that BE can only be transmitted in the prioritized slot.

MAC #3. Contention-based phases: In Protocol 2, if two or more BE slots occur one after another, the channel sensing must be restarted, ignoring previous information about the state of the channel that could help to access it more efficiently. In Protocol 3 we merge consecutive BE slots into a continuous phase, where nodes access the medium via contention without any pre-assigned priority. Due to the larger contention period, it is possible that more than one transmission fits, given that the size of each slot is the transmission time plus again some small guard time. When finishing a contention-based phase we further propose two alternatives, either to keep the current backoff counters and resume them at the beginning of the next phase (MAC 3A), or to randomize a new backoff value at the beginning of each contention phase (MAC 3B). The first option is thought to perform better, since the value of the backoff at each point in time depends not only on the original value randomized from the CW, but also on how long the node has been waiting to access the channel. If we randomize a new backoff value, this waiting time is likely increased. The worst case channel access delay is however still unbounded in both cases for Protocol 3, since it depends on the instantaneous channel load.

Besides the mechanisms described for each of the proposals, Protocol 2 and 3 also need an additional mechanism to avoid that a message being sent at the end of a BE slot/phase overlaps with the following TT or RC slot, interfering with their real-time behaviour. For this, we define a restricted phase at the end of the slot/phase, in which no messages can be sent and the channel cannot be sensed and thus no backoff counters are decremented. The duration of the restricted phase is exactly the time it takes for a transmission to complete, so that a message can be sent just before the restricted phase starts, and it will be finished at the end of the restricted phase without interfering with the next TT or RC slot.

III. Simulation and Results

In order to simulate the wireless MAC protocols, we have created a model for each of them in OMNeT++, a discrete event C++ library and simulation framework[1]. Concretely, the models are implemented using the INET network simulation framework [14], that supports the simulation of the IEEE 802.11b physical and MAC layer. The simulation registers a value for the channel access delay for every packet sent. The channel access delay reflects the time it takes for a packet from the instant it enters the MAC layer, to the instant when it is going to be sent. Furthermore, the average channel access delay including queuing is also evaluated. The queuing occurs when two or more packets from the same traffic class are waiting to be sent. The aim of the simulation is to test how the different configurations for each of the MAC protocols behave under different traffic patterns. Apart from the channel access delay, the simulation accounts also the number of collisions. This way, it is possible to analyse the connection between the two performance metrics, and detect situations in where, e.g., a MAC protocol performing well in regards to channel access delay is paying the price of having a great number of collisions.

The setup we have used to compare the different configurations comprises a small wireless network in infrastructure mode, having five end-nodes and one AP, big enough to obtain results that allow to see the differences between the configurations. The traffic goes only from the nodes to the AP (uplink), a setup that is common in industrial sensor networks. The size of the exchanged packets is relatively small (62 bytes, of which 4 bytes are payload), and has been selected as packets in industrial networks usually are small. The selected bitrate is 11 Mbps, the highest possible in 802.11b. For MAC Protocol 2, we test two protocol variations: either we include time for contention in each slot (Protocol 2B: $T_{slot} = T_{transm} + T_{AIFS} + T_{CW}$) or we only include time for the longest AIFS and thus only zero as randomized backoff value (Protocol 2A: $T_{slot} = T_{transm} + T_{AIFS}$). For MAC Protocol 3, we also have two options: storing the backoff value from a previous phase (Protocol 3A) or restarting and randomizing a new one every time (Protocol 3B). Concerning the data traffic, TT/RC packets are generated periodically, while BE packets are randomly generated according to the specified packet load. The load is defined as the percentage of slots that are occupied within each hyperperiod. To make the results comparable, the application generating packets adjusts the rate to the protocol with the largest slot size, so that for a load of 100%, only the protocol with the largest slot size is fully loaded, whereas all other protocols with smaller slot sizes are not, i.e., do not have a packet to send in each slot. Regarding the traffic patterns, we have selected a low-medium load of 40% and a high load of 100%. These loads can be achieved by the combination of different types of traffic. We have selected combinations in where there is a majority of TT/RC, a majority of BE, or a balance between them. Furthermore, this load can be generated by a single sending node, or be shared such that all the nodes are sending. Further, all the MAC protocols have been tested under different BE slot distributions. The BE slot distribution refers to how the BE slots are allocated along the hyperperiod: either all packed together after the TT/RC traffic or evenly distributed in between TT/RC traffic. All traffic-related simulation parameters are summarized in Table I. The combination of different MAC protocol variations and different traffic patterns resulted in a total of 140 different tests. The simulator has been run for enough time to have around 500 channel access delay records for each of these combinations.

TABLE I. Traffic pattern related simulation parameters.

Load per traffic type	Low-medium load	20% (TT/RC) - 20% (BE)
		10% (TT/RC) - 30% (BE)
		30% (TT/RC) - 10% (BE)
	High load	0% (TT/RC) - 100% (BE)
		20% (TT/RC) - 80% (BE)
		50% (TT/RC) - 50% (BE)
		80% (TT/RC) - 20% (BE)
BE load distribution	One sending node	
	All sending nodes	
BE slot distribution	Packed together	
	Evenly distributed	

When evaluating MAC Protocol 3 for different BE slot distributions (all packed together or evenly distributed in between TT/RC slots), it became clear that channel access was not possible when the BE slot distribution was evenly distributed and there was 50% or more TT/RC traffic. This is due to the fact that when there is 50% or more TT/RC traffic, it is not possible to have two or more consecutive BE slots which in turn means that there is not enough time to finish a transmission in a single slot. Hence, if it is not possible to influence the scheduler such that BE slots can be scheduled consecutively, Protocol 3 should not be used. Conversely, the BE slot distribution only have negligible effect on Protocol 1 and Protocol 2 (it does affect the minimum delay, but only marginally influence the average delay), and consequently, our evaluation henceforth only considers the case when then the BE slots occur consecutively.

In Figure 3, the average channel access delay for MAC

[1]A complete report about the simulator is available at www.es.mdh.se/publications/4302-.

Protocol 1 is presented. We can see that Protocol 1 is not good when having only one sender, but performs very well when having all the nodes sending. This is a clear benefit of the round-robin mechanism, that evenly distributes the opportunities to access the medium. If there is only one sender, all the channel access opportunities assigned to the non-sender nodes are lost, and the channel access delay for the sender node is significantly increased. We can also see that the delay increases with the traffic load, and BE traffic is especially harmed by the amount of TT/RC. However, the main benefit of Protocol 1 is that it does not suffer any collisions and thereby provides predictable upper-bounded channel access delay for all three traffic classes.

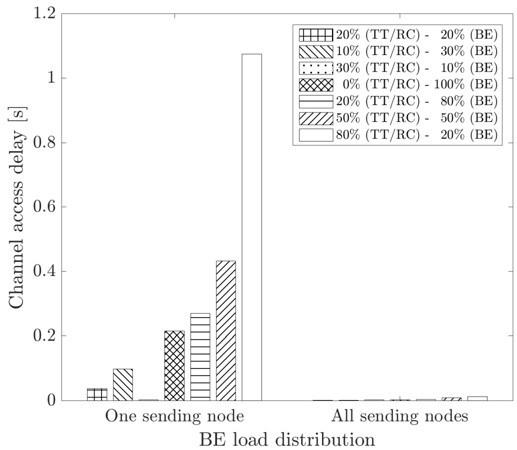

Fig. 3. Average channel access delay for MAC Protocol 1 for different loads and distributions of load.

Figure 4 shows the average channel access delay for Protocol 2. If no contention is allowed, Protocol 2 performs similarly to Protocol 1. However, when allowing contention, the difference in channel access delay between traffic being sent from one or from all the nodes is negligible. Unfortunately, in Figure 5, we see that collisions occur when data emanates from several nodes, and the number of collisions increase when contention is allowed.

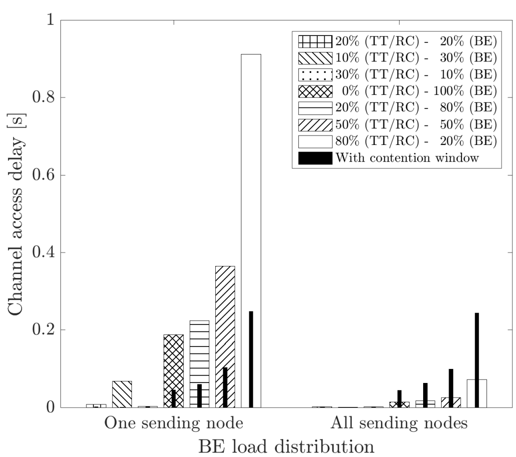

Fig. 4. Average channel access delay for MAC Protocol 2A for different loads and distributions of load. Thin black bars are for MAC 2B.

Figure 6 shows the results for the average channel access delay with MAC Protocol 3. We can see that the protocol

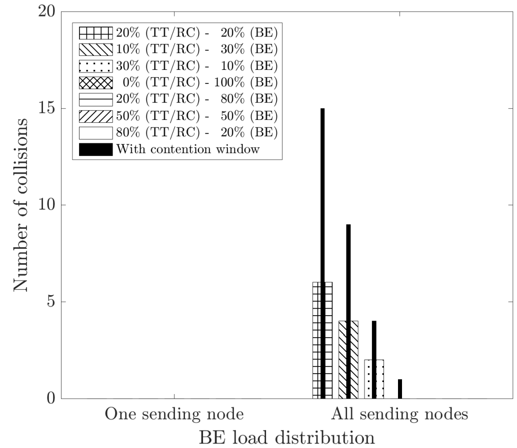

Fig. 5. Number of collisions for MAC Protocol 2A for different loads and distributions of load. Thin black bars are for MAC 2B.

option of storing or restarting the backoff counter at every phase does not have any significant effect, and is only slightly lower in the case of storing the backoff value between phases. We can also see that the channel access delay is generally much lower than with Protocol 1 and Protocol 2. However in Figure 7, we see that the price to pay is that the number of collisions is quite high when all nodes are sending.

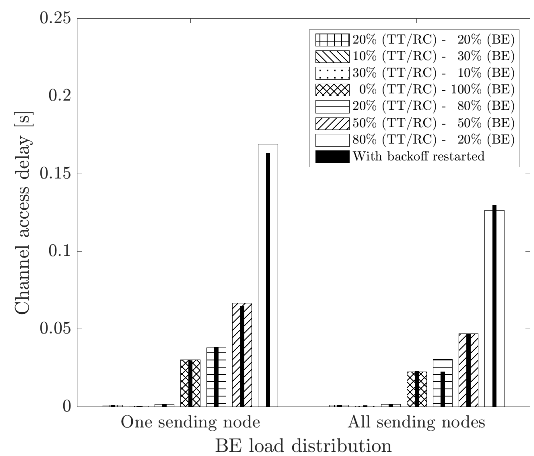

Fig. 6. Average channel access delay for MAC Protocol 3A for different loads and distributions of load. Thin black bars are for MAC 3B.

Figure 8 summarizes the mean values for the channel access delay for all the different MAC protocols. Given these results, we can say that Protocol 3 yields the lowest average channel access delay, unless 50% or more of the traffic is reserved for TT/RC and the slots are evenly distributed. Protocol 1 is best when all the nodes are sending traffic, both in terms of average and guaranteed maximum channel access delay. In regards to the number of collisions, Protocol 2 and 3 obviously do not suffer from collisions when having only one sender, and since they have lower channel access delay than Protocol 1, they are the preferred options when the data traffic emerges from only one node. When the load distribution is unknown, Protocol 2 with contention provides the best tradeoff as the worst case delay is bounded and collisions only occur when the prioritized node has nothing to send.

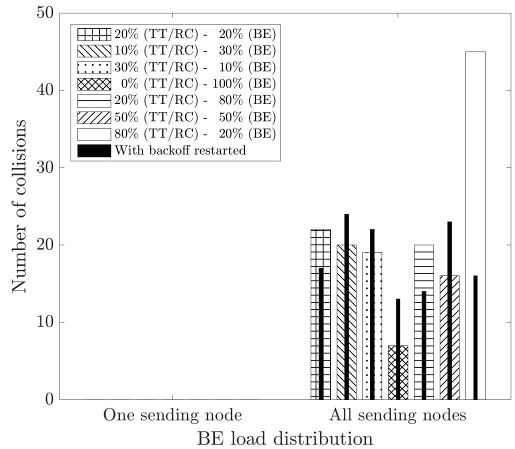

Fig. 7. Number of collisions for MAC Protocol 3A for different loads and distributions of load. Thin black bars are for MAC 3B.

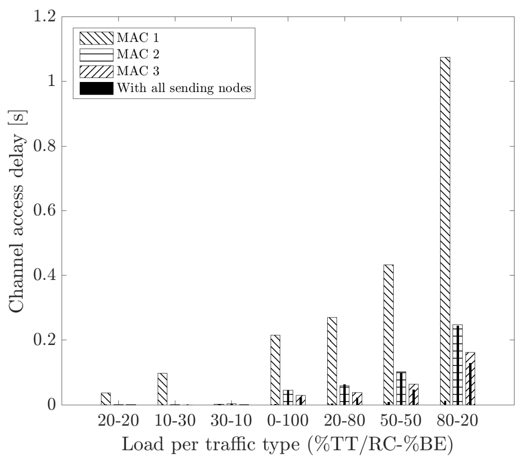

Fig. 8. Average channel access delay for the different MAC protocols under different loads and distributions of load. For MAC 2 and MAC 3, only the best configurations are shown (MAC 2B and MAC 3B). Thin black bars represent the case when traffic comes from all nodes.

IV. Conclusion

In this paper, we have presented three proposals of wireless MAC protocols that based on their worst case channel access delay (evaluated analytically) are able to support traffic with diverse time and safety requirements, and be used to extend an existing wired real-time network. We have also implemented our proposals in the well-known network simulator INET (for OMNeT++), and retrieved average values for the channel access delays, and the number of collisions for the protocols tested under different protocol configurations and traffic patterns. For the TT and RC traffic classes, the delays are known and predictable, but for BE traffic this is generally not the case. Our goal in this paper is to be able to provide an upper bound on the channel access delay also for this traffic class, such that it could be used to support event-driven real-time data traffic. Based on our simulations, we can conclude that the selection of the best MAC protocol and its settings depends on the traffic pattern. The more that is known about the data traffic, the better performance can be achieved by selecting an appropriate MAC protocol. Specifically, if data emerges from one node, MAC Protocols 2 or 3 are preferred, whereas when data is evenly distributed among the nodes Protocol 1 is best. Conversely, when nothing is known, MAC Protocol 2 emerges as the best trade-off.

Acknowledgements

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement 607727 (RetNet), and 610640 (DREAMS). E. Uhlemann and M. Björkman are partly funded by the Knowledge Foundation through the READY project.

References

[1] J. Son, H. Choi, and S.-C. Park, "An effective polling MAC scheme for IEEE 802.11e," in *Proc. IEEE ISCIT*, Sapporo, Japan, 2004, pp. 296–301.

[2] R. Moraes, F. Vasques, P. Portugal, and J. A. Fonseca, "VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks," *IEEE Trans. Ind. Informatics*, vol. 3, no. 3, pp. 215–224, 2007.

[3] H. Trsek and J. Jasperneite, "An isochronous medium access for real-time wireless communications in industrial automation systems - A use case for wireless clock synchronization," in *Proc. IEEE ISPCS*, Munich, Germany, 2011, pp. 81–86.

[4] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *Proc. IEEE RTSS*, Vancouver, Canada, 2013, pp. 140–149.

[5] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, and M. Nixon, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. IEEE RTAS*, St. Louis, USA, 2008, pp. 377–386.

[6] T. Zheng, M. Gidlund, and J. Akerberg, "WirArb: A New MAC Protocol for Time Critical Industrial Wireless Sensor Network Applications," *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2127–2139, 2016.

[7] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks," *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 824–835, 2014.

[8] R. Costa, P. Portugal, F. Vasques, R. Moraes, and R. F. Custódio, "A coordination layer to handle real-time communication in Wi-Fi networks with uncontrolled traffic sources," in *Proc. IEEE LCN*, Bonn, Germany, 2011, pp. 263–266.

[9] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "TTEthernet: Time-Triggered Ethernet," in *Time-Triggered Communication*, R. Obermaisser, Ed. CRC Press, 2011.

[10] P. Gutiérrez Peón, E. Uhlemann, W. Steiner, and M. Björkman, "A Wireless MAC Method with Support for Heterogeneous Data Traffic," in *Proc. IEEE IECON*, Yokohama, Japan, 2015, pp. 3869–3874.

[11] A. Varga, "The OMNeT++ discrete event simulation system," in *Proc. ESM*, vol. 9, no. S 185, Prague, Czech Republic, 2001, p. 65.

[12] W. Steiner, "An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks," in *Proc. IEEE RTSS*, San Diego, USA, 2010, pp. 375–384.

[13] L. De Moura and N. Bjørner, "Satisfiability modulo theories: introduction and applications," *Communications of the ACM*, vol. 54, no. 9, pp. 69–77, 2011.

[14] "The INET Framework. An open-source communication networks simulation package for the OMNeT++ simulation environment," http://inet.omnetpp.org/, accessed: 2016-07-06.