<http://www.diva-portal.org>

Preprint

This is the submitted version of a paper presented at *6th International Workshop on Multiple Access Communications (MACOM 2013)*, Vilnius, Lithuania, Dec. 16-17.

Citation for the original published paper:

Jonsson, M., Kunert, K., Bilstrup, U. (2013)

A real-time medium access protocol supporting dynamic spectrum allocation in industrial networks.

In: Magnus Jonsson, Alexey Vinel, Boris Bellalta, Ninoslav Marina, Desislava Dimitrova, Dieter Fiems (ed.), *Multiple Access Communications: 6th International Workshop, MACOM 2013, Vilnius, Lithuania, December 16-17, 2013. Proceedings* (pp. 54-69). Heidelberg: Springer Lecture Notes in Computer Science

http://dx.doi.org/10.1007/978-3-319-03871-1_6

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-24280>

Jonsson, M., K. Kunert, and U. Bilstrup, "A real-time medium access protocol supporting dynamic spectrum allocation in industrial networks," *Multiple Access Communications, 6th International Workshop, MACOM 2013*, Vilnius, Lithuania, Dec. 16-17, 2013, *Lecture Notes in Computer Science*, vol. 8310, Springer International Publishing Switzerland, pp. 54-69, 2013, ISBN 978-3-319-03870-4.

The final publication is available at link.springer.com/chapter/10.1007/978-3-319-03871-1_6

A Real-Time Medium Access Protocol Supporting Dynamic Spectrum Allocation in Industrial Networks

Magnus Jonsson, Kristina Kunert, and Urban Bilstrup

CERES – Centre for Research on Embedded Systems, Halmstad University, Box 823, SE-30118 Halmstad, Sweden.

{Magnus.Jonsson, Kristina.Kunert,
Urban.Bilstrup}@hh.se

Abstract. Cognitive radio with spectrum sensing and spectrum reuse has great opportunities for industrial networking. Adapting to the current interference situation and utilising the available frequencies in an effective manner can greatly improve the data delivery capabilities. At the same time, real-time demands must be met. In this paper, we present a medium access control protocol supporting dynamic spectrum allocation as done in cognitive radio networks, providing deterministic medium access for heterogeneous traffic. The possibility of spectrum sensing in the nodes opens up for the possibility of increasing successful data transmissions, and a real-time analysis framework with three formalized constraints to be tested provides support for guaranteed timely treatment of hard real-time traffic. The real-time analysis framework includes a new type of delay check that more exactly bounds the delay compared to earlier work. Simulation experiments and performance comparisons are provided.

Keywords: Real-Time Guarantees, Medium Access Control, Cognitive radio

1 Introduction

Industrial communication systems often have to work in an environment where other networks or radiation create different levels of interference for the data traffic. To improve the situation, cognitive radio networks [1-2] offer great potential. They enable dynamic spectrum access by, e.g., sensing channel characteristics and channel usage to adapt transmission parameters to the current channel availability and channel quality. Networks with spectrum reuse have already been targeted in the standardization through the work on IEEE 802.22 [2]. The standard describes a network where secondary users can utilize frequency channels not currently in use by their primary users (licensed TV broadcasting). Even though the same type of channel reuse can be used in industrial networks [3], the great potential for cognitive radio in an industrial context, in our opinion, lies in the adaptation based on the current channel quality of

different available frequencies [4-5]. Spectrum sensing in an industrial context has also been addressed in, e.g., [6-7].

We present a real-time medium access control protocol for a cognitive radio network which provides deterministic medium access for heterogeneous traffic and supports dynamic spectrum allocation. The medium access is scheduled according to the earliest deadline first (EDF) algorithm. A real-time analysis is used as an admission control tool in order to only admit traffic that will not jeopardize any deadline.

Cognitive radio systems are a hot research topic and a number of MAC (Medium Access Control) protocols targeting real-time traffic support have been presented. However, none of the reviewed protocols can provide the deterministic medium access necessary for real-time applications as found in, e.g., industrial control systems. In [8], the authors target Quality of Service (QoS) support only indirectly by striving to protect the QoS of the primary users. The QoS-aware MAC protocol presented in [9] cannot provide any delay bound guarantees due to its random access nature. Moreover, it only supports a limited number of traffic classes and does not include deadline-aware scheduling. In [10], a collision-free MAC protocol considering QoS demands is presented, but without a delay analysis to state delay bound guarantees. QoS support for cognitive radio systems is also discussed in [11], but without specifying a specific MAC protocol providing delay bound support. A framework to support both real-time and non-real-time traffic is introduced in [12], but no methods to calculate delay bounds are presented. More MAC protocols in general for cognitive radio systems can be found in, e.g., [13]. In [14] the authors proposed a method for hard scheduling of the unifying slot assignment protocol (USAP) based on virtual circuits, but the analysis framework is based on capacity bounds instead of deadline bounds.

Our main contributions in this paper are

1. a deterministic MAC protocol supporting spectrum-aware cognitive radio and heterogeneous real-time traffic handling for industrial applications (presented earlier in a work-in-progress paper [15]),
2. a real-time analysis framework, including three constraints to be fulfilled, able to guarantee real-time demands of hard real-time traffic,
3. a more exact delay analysis, as part of the real-time analysis framework, that can guarantee more real-time traffic,
4. a new constraint check, also part of the real-time analysis framework, that is introduced due to the limited size of control packets,
5. a performance evaluation of the medium access protocol through simulation, and
6. a comparison of the new analysis method with the one presented previously [15].

The remaining paper is structured as follows. Section 2 describes the system architecture assumed, followed by the protocol specification in Section 3. The timing and real-time analysis framework is presented in Section 4, while performance evaluations are presented in Section 5. Section 6 concludes the paper.

2 System Model

The target network is a single-hop network with all of the nodes being in each other's transmission range. One node acts as a control node taking responsibility for scheduling and coordination of which radio frequencies to use.

In cognitive radio networks the available network capacity depends on the existence of other networks using the same frequency bands. Therefore, all nodes are equipped with radio sensing equipment and the control node collects sensing information, i.e., information about the different available frequencies, as, e.g., experienced bit error rate or load on a specific frequency. The control node uses this information when scheduling packets and frequencies, and distributes to the remaining nodes not only the schedule, but also information on the frequency spectrum to use. Each node in the network has three queues dedicated to hard real-time (HRT), soft real-time (SRT), and non-real-time (NRT) traffic, respectively (Fig. 1). Packets are sorted in deadline-order using EDF scheduling in the HRT and the SRT queue, i.e., the packet with the shortest deadline is at the head of the respective queue. As non-real-time traffic is assumed to have no deadline, this queue is sorted according to the first come first serve algorithm (FCFS). Strict priority ordering is used in the multiplexing between the queues, i.e., all packets in the HRT queue are treated prior to those in the SRT queue, and those in the NRT queue are only treated if both the HRT and the SRT queue are empty.

Fig. 1. Node architecture

3 Protocol Specification

Cognitive radio networks need flexible MAC protocols that are able to adapt to the changing radio environment. Other interfering networks lead to different frequencies being available and therefore any MAC protocol needs to support dynamic spectrum allocation. The available network capacity is divided between hard real-time traffic on one hand and soft and non-real-time traffic on the other hand. This division is done in a way that hard real-time traffic is guaranteed an experienced minimum bit rate (R_{HRT}). R_{HRT} is a system parameter normally chosen according to $R_{HRT} \leq R_{Min}$, where R_{Min} is the minimum available physical bit rate expected at any given point in time. The remaining network capacity is dedicated for soft real-time and non-real-time

traffic in order to not starve these traffic classes and provide a certain degree of fairness in the network.

As mentioned above, the network has one node acting as a control node, responsible to collect all the nodes' spectrum sensing information, schedule the traffic, and broadcast the schedule and spectrum allocation to the rest of the network. The access to the network is cyclic and divided into superframes (Fig. 2). All nodes are synchronized on reception of the feedback from the control node. In the start of a superframe of length T_{cycle} each node will sense the available spectrum for the length of T_{sense} . During the following control phase ($T_{control}$), each node has an individual control slot during which it broadcasts the result of its spectrum sensing, plus information about the β first packets (referred to as data packets) in its local HRT EDF queue. In case the HRT EDF queue contains less than β data packets, the remaining space in the control packet is filled with information about the first packets in the SRT queue. Only if information about all packets in the HRT and SRT queue can be sent, the node will add information about packets in the NRT queue to the control packet until the maximum amount of information is reached (i.e., a maximum of β transmission requests). If, e.g., each request consists of 16 bits, ten bits can express the deadline, while six bits can express the destination address of the packet the request relates to. The six bit address can, in that case, address up to 64 nodes. If an underlying standard with longer addresses is used, each node can have a table to translate the six bit address to, e.g., a 48 bit MAC address. Since we target real-time systems for which predictability is important, those tables can be assumed to be known at system start-up. The control packets can potentially also be used to efficiently carry short messages for services like process synchronization [16]. The data transmission requests, gathered from the control packets during the collection phase, are stored in a global EDF queue in the control node. During $T_{feedback}$, the control node will schedule transmission requests globally in the same order as in the end nodes, i.e., according to EDF or FCFS within each traffic class and strict priority between the classes, and then broadcast the schedule to the nodes. Together with the schedule, information on the frequency to be used by all nodes during the current superframe is included. $T_{feedback}$ includes time for the nodes to tune to the right frequency (or frequencies if, e.g., OFDM is used) according to the spectrum information sent out by the control node. The remaining time of the superframe (T_{data}) is time-slotted according to the schedule sent out by the control node. T_{data} can be calculated straight forward as follows.

$$T_{data} = T_{cycle} - T_{sense} - T_{control} - T_{feedback} \quad (1)$$

Fig. 2. Superframe structure

It is important to avoid inconsistency in the scheduling result among the nodes. A node is therefore not allowed to initiate a new transmission during an on-going super-frame if its control packet during this superframe is lost. Concerning the reliability of data messages, we have developed a framework combining a retransmission scheme with real-time support [17-19]. The framework has also been adapted for IEEE 802.15.4 in a multichannel context [20]. An adaptation for the here described protocol is possible, but outside the scope of this paper.

4 Timing and Real-Time Analysis

In order to be able to guarantee that the deadlines of the hard real-time traffic are met, a real-time timing and scheduling analysis is necessary. This analysis will be able to provide the necessary guarantees under the assumption that traffic characteristics are specified and followed by the nodes.

The hard real-time traffic in the network is specified in the form of logical real-time channels (RT channels) which are traffic flows denoted as τ_i with $1 \leq i \leq Q$, where Q defines the number of RT channels. The RT channels are specified by the following parameters: sending node $m_{s,i}$, receiving node $m_{d,i}$, period P_i , message length L_i , and end-to-end delay bound D_i . Each RT channel τ_i can therefore be defined completely by the following expression: $\tau_i = \{m_{s,i}, m_{d,i}, P_i, L_i, D_i\}$.

Messages sent over the network are divided into packets. The maximum total transmission time $T_{x_tot,i}$ of one message belonging to τ_i , i.e., the total time the physical channel is occupied, is calculated in the following way:

$$T_{x_tot,i} = \left\lceil \frac{L_i}{L_{data}} \right\rceil \cdot T_{packet} \quad (2)$$

L_{data} denotes the maximum amount of pure data (payload) per packet, and T_{packet} is the maximum duration of a data packet (slot) plus one interframe space (T_{IFS}), i.e.,

$$T_{packet} = \frac{L_{data} + L_{header}}{R_{Min}} + T_{IFS} \quad (3)$$

where L_{header} is the length of the header per packet. The transmission time of a data message can then be defined as

$$T_{x_tot,i} = \left\lceil \frac{L_i}{L_{data}} \right\rceil \cdot \left(\frac{L_{data} + L_{header}}{R_{Min}} + T_{IFS} \right) \quad (4)$$

The queuing discipline assumed for the data packets is EDF, i.e., the packets are sorted in the queue according to their absolute deadlines (generation time plus relative deadline). Deadlines are not restricted to be equal to the periods, but arbitrary deadlines, i.e., shorter than, equal to, or longer than the periods, are supported.

As the real-time scheduling analysis only considers delay introduced by queuing, plus the total transmission time $T_{x_tot,i}$ of a message, all other delays need to be excluded before further analysis. The maximum queuing delay, d_i , can be isolated by subtracting all other delays from the relative deadline D_i , resulting in the following maximum queuing delay (and new relative deadline):

$$d_i = D_i - T_{cycle} - T_{feedback} - T_{control} \quad (5)$$

The worst case scenario reflected in the equation happens if the packet arrives in the node queue just a fraction too late for the node to be able to include its information in the next control packet. This will lead to the data packet having to wait one complete cycle time, plus additional time during the control and the feedback phase until the next data phase starts before it can be sent. For the worst case delay analysis, furthermore, the worst case position of the control packet is assumed, i.e., it is sent first of all control packets during the control phase. Relaxing this assumption, taking into account the actual position of the control packet, would make it possible to improve the analysis by making it less pessimistic. Anyway, no blocking time for a lower-priority packet has to be subtracted from the end-to-end delay bound of the data packet under consideration, as the control node schedules all of T_{data} at the same time.

The real-time scheduling analysis is used to check if a traffic allocation over the network is feasible, i.e., it checks if all messages belonging to the scheduled RT channels will meet their deadlines. The first of three conditions to be checked is link utilization. It is a necessary, but not sufficient (as the deadlines of any RT channel are not restricted to be equal to the periods of that channel), condition that the utilization U_{HRT} by the allocated HRT traffic over the network does not exceed its allocated maximum utilization U_{HRT_Max} . The formal demand is:

$$U_{HRT} \leq U_{HRT_Max} \quad (6)$$

The utilization U_{HRT} of periodic hard real-time traffic can be calculated as follows:

$$U_{HRT} = \sum_{i=1}^Q \left(\frac{T_{x_tot,i}}{P_i} \right) \quad (7)$$

while the maximum possible utilization U_{HRT_Max} by HRT traffic is given by

$$U_{HRT_Max} = \frac{R_{HRT}}{R_{Min}} \cdot \frac{T_{data} - T_{block}}{T_{cycle}} \quad (8)$$

where T_{block} is the maximum transmission time of one data packet (including the necessary preceding interframe space). This blocking time has to be taken into consideration because of the case when the remaining time of T_{data} is just a fraction too short to send a further (maximum sized) data packet. T_{block} is given by

$$T_{block} = \frac{L_{data} + L_{header}}{R_{Min}} + T_{IFS} \quad (9)$$

A second constraint has to be fulfilled in the real-time scheduling analysis in order to ensure that the workload on the network at no time instance will be higher than allowed, so as to meet the deadlines for the allocated traffic. For the description of this second part of the feasibility check, a number of concepts, originating from the area of real-time scheduling, have to be introduced. Firstly, the hyperperiod, HP, of periodic traffic is the least common multiple of all periods of the RT channels, i.e., the interval starting when the periods of all traffic flows start at the same time and ending when they do so again. Secondly, the analysis uses the concept of busy periods, BP, which are any intervals within an HP when the link is busy. Additionally, we need the workload function, $h(t)$, which measures the traffic demand at any point in time in the network. Originally, this function was designed for the control of the processor demand in a real-time system, but due to the assumption of EDF scheduling, this analysis can be adapted for analysing traffic scheduling in a network [21]. Generally speaking, $h(t)$ is the sum of the transmission times of all data packets for all message instances of all RT channels that have an absolute deadline that is less than or equal to a point in time t , and where t is the number of time units elapsed since the beginning of HP. This synchronous traffic pattern where all RT channels' periods start at the same time is the worst case in terms of workload, leading to the worst case queuing delay for the data packets pertaining to those RT channels [22-24]. $h(t)$ is calculated as:

$$h(t) = \sum_{i \in [1, Q], d_i \leq t} \left(1 + \left\lfloor \frac{t - d_i}{P_i} \right\rfloor \right) \cdot T_{x_tot, i} \quad (10)$$

The summation only includes terms for which the value of t is equal or higher than the value of the corresponding relative deadline. The second constraint, introduced in [22-23] and generalized in [24], was added in order to be able to ensure the continued feasibility of the traffic allocation even for the case when a new traffic flow is added. The original constraint (to be modified below) demands that:

$$h(t) \leq t \quad \forall t \geq 0 \quad (11)$$

In our case, transmissions cannot occur all the time due to the sensing phase, the control phase, etc. After the worst-case delay described above, the data phase starts and the total guaranteed time $g(t)$ available for transmission up until time t can be calculated as

$$g(t) = \left\lfloor \frac{t}{T_{cycle}} \right\rfloor \cdot (T_{data} - T_{block}) + \min(T_{data} - T_{block}; \text{mod}(t, T_{cycle})) \quad (12)$$

The first part of the equation delivers the possible transmission time contributed by an integer number of whole superframes, while the second part contributes with the corresponding time for the last but not fully passed superframe. The start of a superframe, i.e., $t = 0$, in this context actually occurs when the data phase starts since delays compensating for the sensing, control, and feedback time have already been subtracted from the end-to-end delay bound in order to isolate the queuing deadline, d_i . In other words, as we in Eq. 5 derives the pure queuing deadline by subtracting other delays, we can assume that $t = 0$ when the data phase starts, i.e., when we can start sending queued packets. The original constraint is thereby modified to the following demand:

$$h(t) \leq g(t) \quad \forall t \geq 0 \quad (13)$$

Unfortunately, the constraint stated in Eq. 13 suffers from a high degree of computational complexity, but [25] presents a way of reducing the time and memory complexity by limiting the number of time instances for which $h(t)$ has to be checked to a sufficient subset. It is possible to reduce the number of instances of evaluation to those of absolute message deadlines during an interval upper-bounded by BP_1 , the first busy period in the first hyperperiod of the schedule where all periods start at time zero, i.e.,

$$t \in \bigcup_{i=1}^Q \{m \cdot P_i + d_i : m = 0, 1, 2, \dots\} \quad (14)$$

where

$$t \in [1; BP_1] \quad (15)$$

A third constraint is introduced by the use of control packets. The length of a control packet will limit the amount of data packets about which control information can be sent to the control node. The condition on the control packet is therefore that there has to be enough space in the control packets to be able to inform about possible queued hard real-time packets. So the following must be true for all i , $1 \leq i \leq M$, where M is the number of end nodes in the network:

$$\beta \geq \sum_{j=1}^{Q_i} \left\lceil \frac{T_{cycle}}{P_j} \right\rceil \cdot \left\lceil \frac{L_j}{L_{data}} \right\rceil \quad (16)$$

where β is (as stated earlier) the number of data packets a control packet can carry information about, and Q_i is the number of RT channels with node $m_{s,i}$ as their source node. In Eq. 16, the summation contains only RT channels from one specific node $m_{s,i}$ at a time.

When assuming a maximum of S packets during the data phase and S is given by

$$S = \left\lfloor \frac{T_{data}}{T_{x,min}} \right\rfloor, \quad (17)$$

denoting the number of minimum-sized data packets fitting in the data phase, and $T_{x,min}$ being the length of a minimum-sized data packet, it is sufficient that the following inequality holds:

$$\beta \geq S. \quad (18)$$

The reason is that even in the worst case, i.e., when one node has all S packets that are to be scheduled first by the control node scheduler, this node can still send information about all of them in one single control packet.

Only in the case when all three constraints (utilization, workload, and control packets) are fulfilled, a feasible traffic scheduling can be guaranteed, i.e., only then will it be possible to guarantee that no deadlines will be missed.

In the runtime implementation of this schedulability test the three constraints are checked every time a new RT channel is added in order to guarantee the requested delay bounds for both the new and the existing RT channels. As new RT channels are not expected to be requested frequently, the computational demands of this admission control will not be very high, and any of the nodes might be responsible for it and be chosen as the control node in the network. If all RT channels are known at the design stage or system start-up of the network, the analysis can be made offline instead.

5 Performance Analysis

In order to analyse the behaviour and show the performance of the studied network, we have conducted simulation experiments using MATLAB. We have simulated the network both with hard real-time traffic, including the real-time analysis, and with soft real-time traffic, i.e., injecting traffic without the use of the real-time analysis. We have studied throughput, average delay and deadline miss ratio. Moreover, we have compared the real-time analysis, in terms of reachable utilization for hard real-time traffic, with the analysis presented earlier in [15].

5.1 Simulation Assumptions

We assume a single-hop network with $M = 20$ nodes, where one node is acting as the protocol control node. Several of the network parameters have values inspired by the IEEE 802.11b standard, as described below. The minimum physical bit rate allocated for hard real-time traffic, R_{HRT} , is assumed to be equal to the minimum bit rate R_{Min} available in low-bandwidth situations (according to the spectrum sensing). We assume the minimum bit rate to be 11 Mbit/s, i.e., $R_{HRT} = R_{Min} = 11$ Mbit/s. The propagation delay is assumed to be negligible.

The length of the sensing phase is assumed to be $T_{sense} = 2$ ms. The duration of a control packet is set to $196 \mu\text{s}$, corresponding to 320 bits excluding preamble, header, and interframe spacing. The total duration of the control phase is therefore:

$$T_{control} = 20 \cdot 196 = 3920 \mu\text{s} . \quad (18)$$

The maximum number of packets covered by one control packet is assumed to be $\beta = 20$. The duration of the feedback phase, including time for the control node to do the traffic scheduling, is set to $T_{feedback} = 1$ ms, while the whole superframe duration is set to $T_{cycle} = 30$ ms.

The traffic over the network belongs to one of three traffic classes and the parameters for all RT channels are chosen from these with even random distribution. The three traffic classes have periods of 50 ms, 100 ms, and 200 ms, with the deadlines in all traffic classes being equal to the periods. The periods of the different traffic flows are not synchronized in the simulations. Instead, each traffic flow starts with an offset from the starting time, randomly generated with even distribution between zero and its period. Independent of the traffic class, a packet with duration of $T_{x_tot,i} = 200 \mu\text{s}$ is generated at the start of each period of an RT channel. The $200 \mu\text{s}$ correspond to a 45 Byte (360 bit) packet excluding preamble, header, and interframe spacing.

5.2 Hard Real-Time Traffic

Fig. 3 shows the throughput when simulating only hard real-time traffic. After the generation of about 200 RT channels, some of the RT channels are starting to be rejected by the real-time analysis in the admission control. Each simulation point is run ten times for the duration of ten hyperperiods.

Fig. 3. Throughput for the case with hard real-time traffic.

Since the delay bounds are guaranteed, the shape of the average delay curve (Fig. 4) is given. The average delay is significantly lower than the delay bounds specified for the traffic classes (50 ms, 100 ms, and 200 ms). This is also not surprising since the worst-case delay analysed in the real-time analysis constitutes a case not always happening. The deadline miss ratio has also been measured, but is always zero as expected when using the real-time analysis. The curve is therefore omitted.

Fig. 4. Average delay for the case with hard real-time traffic.

5.3 Soft Real-Time Traffic

When simulating soft real-time traffic, no RT channels are rejected. The network can thereby reach saturation, which becomes evident in Fig. 5 showing the average delay. For each amount of requested RT channels, ten simulations are run to get smoother curves, while each simulation is run for the duration of ten simulated hyperperiods.

Fig. 5. Average delay for the case with soft real-time traffic.

The throughput is plotted in Fig. 6. A throughput of about 75% is reached before saturation. Compared to numerous other networks this is very high, which shows that the control overhead in the network is well invested, not only supporting spectrum reuse but also efficient scheduling.

Fig. 6. Throughput for the case with soft real-time traffic.

Fig. 7 shows the deadline miss ratio, where again the bottleneck is evident. Deadlines are missed first at (or near) the saturation point.

Fig. 7. Deadline miss ratio for the case with soft real-time traffic.

5.4 Real-Time Analysis Comparison

As mentioned above, we have compared the presented real-time analysis with the analysis proposed in [15], which was based on the concept of experienced bit rate. The experienced bit rate was scaled as seen in [26], compensating for the fact that parts of the superframe are not available for the transmission of data packets. This was done by using the average available network capacity in the workload analysis. In the current paper, the workload function is checked more exactly against the specific parts of the superframe that have passed. This results in a less pessimistic analysis, able to potentially guarantee more hard real-time traffic.

Fig. 8 shows a comparison of the here proposed real-time analysis with the previous one in terms of utilization of accepted hard real-time traffic. The behaviour when the curves separate (at about 175 requested RT channels) is explained by the fact that the previous method cannot accept traffic with tough delay requirements. With an even higher amount of requested RT channels, though, enough RT channels with less strict delay demands have been requested to reach utilization closer to the new method. In order to avoid this behaviour during this evaluation, we restrict the RT channel generation to only one traffic class, in this case with period and delay bound set to 50 ms (otherwise the same simulation parameters are kept). As seen in Fig. 9, the performance difference between the two methods now is much larger. The performance is compared in terms of total achievable utilization for hard real-time traffic, for a few different cases in Table 1. The number of queued packets a control packet can carry information about is assumed to be unlimited in order to instead focus on the difference of the two methods in checking the workload. This, in combination with only having a single traffic class, eliminates the randomness. The parameters are otherwise the same as explained above. As seen, an improvement as high as 31.6% has been observed for shorter delay bounds.

Fig. 8. Utilization comparison of the new and old real-time analysis.

Fig. 9. Utilization comparison of the new and old real-time analysis with a single traffic class.

Table 1. Comparison of achievable utilization for hard real-time traffic.

Period and deadline	Old analysis	Current analysis	Improvement
40 ms	16.9%	22.2%	31.6%
50 ms	22.8%	30.0%	31.6%
60 ms	31.7%	38.0%	20.0%
70 ms	38.0%	39.7%	4.5%
80 ms	42.8%	47.3%	10.5%
90 ms	46.7%	50.7%	8.6%
100 ms	49.6%	50.8%	2.4%
110 ms	52.0%	55.3%	6.3%
118 ms	53.6%	58.1%	8.5%
120 ms	54.0%	57.2%	5.9%
125 ms	54.9%	54.9%	0.0%
130 ms	55.7%	56.6%	1.7%
140 ms	57.1%	59.7%	4.5%
150 ms	58.4%	60.9%	4.3%
160 ms	59.5%	60.4%	1.5%
200 ms	62.9%	64.7%	2.9%

6 Conclusion and Future Work

In this paper we presented a MAC protocol supporting both dynamic spectrum allocation and deadline guarantees for hard real-time traffic. The spectrum sensing is distributed over the nodes in the network to better catch the available spectrum and taking these measurements into consideration when scheduling packet transmissions over the network. A real-time analysis framework was presented, including several novel methods compared to earlier work. Despite the control overhead, the network shows good performance figures, as demonstrated in the simulation study.

References

1. Y.-C. Liang, K.-C. Chen, G.Y. Li, and P. Mähönen, "Cognitive radio networks and communications: an overview," *IEEE Trans. on Veh. Techn.*, vol. 60, no. 7, Sept. 2011, pp. 3386-3407.
2. C. R. Stevenson, G. Chouinard, Z. Lei, W. Hu, S.J. Shellhammer, and W. Caldwell, "IEEE 802.22: the first cognitive radio wireless regional area network standard," *IEEE Comm. Mag.*, vol. 47, no. 1, Jan. 2009, pp. 130-138.
3. D. Cavalcanti, S. Das, J. Wang, and K. Challapali, "Cognitive radio based wireless sensor networks," *Proc. of 17th International Conference on Computer Communications and Networks (ICCCN '08)*, St. Thomas, U.S. Virgin Islands, Aug. 3-7, 2008.
4. D. Block and U. Meier, "Wireless deterministic medium access: a novel concept using cognitive radio," *The Third International Conference on Advances in Cognitive Radio (COCORA 2013)*, Venice, Italy, Apr. 21-26, 2013, pp. 35-38.
5. S. Vitturi, L. Seno, F. Tramarin, and M. Bertocco, "On the rate adaptation techniques of IEEE 802.11 networks for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 198-208, Feb. 2013.
6. K. Ahmad, U. Meier, H. Kwasnicka, A. Pape, and B. Griese, "A cognitive radio approach to realize coexistence optimized wireless automation systems," *Proc. 14th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA 2009)*, Mallorca, Spain, Sept. 22-26, 2009.
7. L. Tytgat, M. Barrie, V. Goncalves, O. Yaron, I. Moerman, P. Demeester, S. Pollin, P. Ballon, and S. Delaere, "Techno-economical viability of cognitive solutions for a factory scenario," *Proc. IEEE Symp. on New Frontiers in Dyn. Spectr. Acc. Netw. 2011 (DySPAN '11)*, May 2011, pp.254-264.
8. Y. Yao, Z. Feng, W. Li, and Y. Qian, "Dynamic spectrum access with QoS guarantee for wireless networks: a Markov approach," *GLOBECOM '10*, Miami, FL, USA, Dec. 2010.
9. L.X. Cai, Y. Li, X. Shen, J.W. Mark, and D. Zhao, "Distributed QoS-aware MAC for multimedia over cognitive radio networks," *GLOBECOM '10*, Miami, FL, USA, Dec. 2010.
10. C. Passiatore and P. Camarda, "A MAC protocol for cognitive radio wireless ad hoc networks," *4th Int. Works. on Multiple Acc. Comm. (MACOM '11)*, C. Sacchi et al. (eds.), LNCS 6886, Springer-Verlag, pp. 1-12, 2011.
11. D. Willkomm and A. Wolisz, "Efficient QoS support for secondary users in cognitive radio systems," *IEEE Wirel. Comm.*, vol. 17, no. 4, Aug. 2010, pp. 16-23.
12. A. Alshamrani, X. Shen, and L.-L. Xie, "QoS provisioning for heterogeneous services in cooperative cognitive networks," *IEEE J. on Sel. Areas in Comm.*, vol. 29, no. 4, Apr. 2011, pp. 819-830.

13. C. Cormio and K.R. Chowdhury, "A survey on MAC protocols for cognitive radio networks," *Ad Hoc Netw.*, vol. 7, no. 7, Sept. 2009, pp. 1315-1329.
14. S. Perumal, V. Tabatabaee, J.S. Baras, C.J. Graff, D.G. Yee, "Modeling and sensitivity analysis of reservation based usap hard scheduling unicast traffic in manets," *Military Communications Conference (MILCOM 2009)*, Oct. 2009.
15. K. Kunert, M. Jonsson, and U. Bilstrup, "Deterministic real-time medium access for cognitive industrial radio networks," *Proc. of the 9th IEEE International Workshop on Factory Communication Systems (WFCS 2012)*, Lemgo/Detmold, Germany, May 21-24, 2012.
16. M. Jonsson, C. Bergenheim, and J. Olsson, "Fiber-ribbon ring network with services for parallel processing and distributed real-time systems," *Proc. ISCA 12th International Conference on Parallel and Distributed Computing Systems (PDCS-99)*, Fort Lauderdale, FL, USA, Aug. 18-20, 1999, pp. 94-101.
17. M. Jonsson and K. Kunert, "Towards reliable wireless industrial communication with real-time guarantees," *IEEE Trans. on Ind. Inf.*, vol. 5, no. 4, Nov. 2009, pp. 429-442.
18. K. Kunert, E. Uhlemann, and M. Jonsson, "Predictable Real-Time Communications with Improved Reliability for IEEE 802.15.4 Based Industrial Networks", *Proc. 8th IEEE Int. Works. on Factory Comm. Syst. (WFCS '10)*, Nancy, France, May 2010, pp.13-22.
19. M. Jonsson and K. Kunert, "Reliable hard real-time communication in industrial and embedded systems," *Proc. 3rd IEEE Int. Symp. on Ind. Emb. Syst. (SIES '08)*, Montpellier, France, June 2008, pp. 184-191.
20. K. Kunert, M. Jonsson, and E. Uhlemann, "Exploiting time and frequency diversity in IEEE 802.15.4 industrial networks for enhanced reliability and throughput," *Proc. 15th IEEE Int. Conf. on Emerging Techn. and Factory Automation (ETFA '10)*, Bilbao, Spain, Sept. 2010.
21. H. Hoang and M. Jonsson, "Switched real-time Ethernet in industrial applications - deadline partitioning", *9th Asia-Pacific Conf. on Comm. (APCC '03)*, Penang, Malaysia, Sept. 2003, vol.1, pp.76-81.
22. S.K. Baruah, A.K. Mok, and L.E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor", *Proc. 11th Real-Time Syst. Symp. (RTSS '90)*, Dec. 1990, pp. 182-190.
23. S.K. Baruah, L.E. Rosier, and R.R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor", *Real-Time Syst.*, vol. 2, nr. 4, Nov. 1990, pp. 301-324.
24. M. Spuri, "Analysis of deadline scheduled real-time systems", *Tech. Rep. 2772*, INRIA, France, 1996.
25. J.A. Stankovic, M. Spuri, K. Ramamritham, and G.C. Buttazzo, *Deadline scheduling for real-time systems - EDF and related algorithms*, Kluwer Academic Publishers, Boston, MA, USA, 1998.
26. A. Böhm and M. Jonsson, "Supporting real-time data traffic in safety-critical vehicle-to-infrastructure communication," *The 2nd IEEE LCN Workshop On User MObility and Vehicular Networks (ON-MOVE)*, Montreal, Canada, Oct. 14, 2008, pp. 614-621.