Extending CAN over the air: an interconnection study with IEEE802.11

Tony Flores Pulgar       Jean-Luc Scharbarg       Katia Jaffrès-Runser
Christian Fraboul
University of Toulouse, IRIT / ENSEEIHT
2 Rue Charles Camichel - F-31061, Toulouse, FRANCE
firstname.lastname@enseeiht.fr

Abstract

The flexibility of wireless connectivity is appealing in the context of industrial networks. This paper discusses the use of a wireless protocol to interconnect remotely located fieldbuses. The focus of this paper is to analyse the feasibility and design issues related to this type of hybrid network architecture. Therefore, we concentrate on deriving appropriate bridging strategies for a network topology composed of remotely located CAN buses interconnected through a wireless local area network following the IEEE802.11g protocol. Using this very simple and cost-effective architecture, we show in this study that by intelligently leveraging the features of CAN and IEEE802.11g in the interconnection policies employed, the missed deadlines can be limited for the CAN frames carried by the wireless network.

1. Introduction

Industrial fieldbus technologies are widely rolled out to offer real-time communication capabilities on the factory floor. A large set of protocols offer deterministic and timely bounded transmissions using tailored medium access schemes and architectures (e.g. PROFIBUS, PROFINET, TTEthernet, etc.). Controller Area Network [1] is one of the mainstream standards for embedded communications. Despite the fact that it has been originally developed for automotive communications, CAN has found its place in factory automation applications to handle sensor-actuator communications because of its ease of use and the low cost of its controllers.

Recent developments for industrial communications consider introducing wireless transmissions into the global network architecture [3][11]. First studies have assessed the capabilities of mainstream wireless technologies such as WiFi (IEEE802.11 [4]), Bluetooth or ZigBee (IEEE802.15.4) [11] for real-time communications. In parallel, new real-time wireless protocols have been designed [5] [6] [7] [10]. Recently, a TDMA-oriented solution called WirelessHART has been commercialised for factory automation applications [10]. The main pitfall of wireless communications is of course the increased unreliability the medium suffers from due to interference and pathloss compared to shielded wires.

There are several main motivations for developing a wireless fieldbus technology. First, wireless networks are much easier to deploy than wired networks. Second, mobile entities such as robots can communicate seamlessly. The works on wireless real time protocol design clearly aim at leveraging these two features. Another interesting benefit of wireless transmissions is to provide a cost-effective network to interconnect distant heterogeneous or homogeneous legacy fieldbuses. The focus of this paper is to discuss this last use case of wireless communications.

A wireless interconnection will benefit architectures where several fieldbuses, located far from each other, need a backhaul network to exchange data. Either legacy wireless technologies such as WiFi or dedicated wireless protocols such as WirelessHART may be chosen, depending on the nature of the traffic exchanged between the remote buses. For hard real-time data, a dedicated reliable wireless solution has to be picked, while for soft real-time data, a cheaper and probably less reliable wireless technology can be chosen. But for both cases, we argue that the key point to achieve a timely behaviour of the end-to-end flows in the network is to carefully define the bridging strategies of the wireless gateways interconnecting the fieldbuses with the wireless network.

This statement is illustrated in this paper with the interconnection of CAN buses through a standard IEEE802.11g wireless network using CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) medium access control in ad hoc mode. In this study case, soft real-time data is exchanged between the remote CAN networks using low-cost wireless Access Points. The paper proposes and discusses different bridging strategies that account for the specifics of CAN and CSMA/CA. The aim of this study is to highlight their impact on end-to-end communication delays in a network where periodic flows are to be received in a timely manner. Of course, CSMA/CA being far from deterministic, the end-to-end (E2E) communication delay is not bounded anymore. But we show through simulations that intelligent encapsulation strategies of CAN frames into WiFi frames signifi-
stantly improves the communication delay on the wireless network. Moreover we clearly show that this architecture is a good candidate for soft real-time traffic.

In our previous work [8], interconnection of CAN buses via legacy Ethernet has been studied. The motivation of studying interconnection through CSMA/CA for WiFi relies on the following facts:

First, a collision on Ethernet is far less time-consuming than on CSMA/CA since it is very quickly detected by emitters. Second, throughput on both technologies is different (e.g. 100 Mbps vs. up to 54 Mbps), at least doubling the transmission duration of a frame. Third, the overhead of collision avoidance and acknowledgement procedure triggers an additional timing overhead for CSMA/CA compared to CSMA/CD.

This paper is organised as follows. Section 2 describes the architecture of interest. Section 3 proposes different bridging strategies. The performance of these strategies is analysed in Section 4. Section 5 concludes the paper and gives directions for future work.

2. Case study architecture

2.1. Network architecture

![Figure 1. Architecture example](image)

An example of the hybrid architecture targeted in this paper is described in Figure 1. Remote embedded networks follow the widely available CAN standard [1]. The wireless local area network interconnecting the remote embedded buses follows the mainstream IEEE802.11g standard [4]. A gateway is implemented between each CAN bus and the IEEE802.11 network. The example in Figure 1 includes three CAN buses and three pure wireless nodes NW1...NW3. Four nodes including one gateway are connected to each CAN bus (e.g. NC4, NC5, NC6, GW2 for CAN bus 2).

All wireless transmitters (GW1, NW1, ...) are interconnected in ad hoc mode (no access point architecture).

2.2. Definition of the flows

Three kinds of flows are transmitted over this architecture:

- pure CAN flows are transmitted between stations connected on the same CAN bus: they do not transit on IEEE802.11,
- pure IEEE802.11 flows are transmitted between wireless stations: they do not transit on CAN,
- hybrid flows are transmitted between stations connected on different CAN buses: they transit on both technologies, via the gateways.

A pure CAN flow fC_i is defined by the following elements:

- an identifier IdC_i between 0 and 2047,
- a source node srcC_i and a set of destination nodes destC_i which all belong to the same CAN bus as the source node srcC_i,
- a period PC_i which is the duration between the generation of two consecutive frames of the flow,
- a critical delay DC_i which is the maximum allowed duration between the generation of a frame and its reception by its destination nodes,
- the size SC_i in bytes of the payload of each frame of the flow.

A hybrid flow fH_j is defined by the same elements as a pure CAN flow: an identifier IdH_j, a source node srcH_j, a set of destination nodes destH_j, a period PH_j, a critical delay DH_j and the size SC_j in bytes of the payload of each frame of the flow. The only difference is that destination and source nodes belong to different CAN bus.

A pure IEEE802.11 flow fW_k is defined by the following elements:

- a source node srcW_k and a set of destination nodes destW_k which are all connected to IEEE802.11,
- an average interframe duration PW_k, following an exponential distribution (Poisson traffic),
- the size SW_k in bytes of the payload of each frame of the flow.

Table 1 presents the set of flows which are transmitted on the network architecture in Figure 1. Hybrid flows are depicted with dotted lines while other ones with full black lines. There are three pure CAN flows (one per CAN bus), four hybrid flows and three pure IEEE802.11 flows. The four hybrid flows are generated by stations from CAN bus 1. Three of them (fH_1, fH_2 and fH_3) have their destination nodes on CAN bus 2, while the last one (fH_4) has its destination node on CAN bus 3.

This configuration will be used as an illustrative example in the rest of the paper.

2.3. CAN protocol

The Controller Area Network (CAN, [11]) is a serial communication protocol suited for networking sensors, actuators and other nodes in real-time systems. The CAN specification defines several versions of the protocols for the physical and the data link layer. In this paper, we focus on CAN 2.0 A.
the frames they are interested in by filtering out the identifiers.

\[
\begin{array}{ccccccc}
1 & 11 & 1 & 1 & 1 & 0.64 & 16 & 2 & 7 & 3 \\
\end{array}
\]

### Figure 2. CAN frame (sizes in bits)

The frame format is depicted in figure 2. Only the three following fields are relevant to the remainder of the paper:

- the identifier field, as mentioned earlier, identifies the data carried by the frame,
- the DLC field gives the length (in bytes) of the data field,
- the data field carries the payload of the frame.

Bit-stuffing is used to avoid the transmission of long sequences of 1s with identical value. The computation of the frame length has to take into account these additional bits. In this paper, we use the upper bound given in [2]. The length \( C \) of a frame carrying \( S \) bytes of data is:

\[
C = (55 + 10 \times S) \tag{1}
\]

The medium access control (MAC) is CSMA/CR: the start of frame transmissions on the bus are synchronous. When two or more stations start a transmission simultaneously, the one with the smallest frame identifier wins and the others stop their transmission. This mechanism guarantees strict priority order on identifiers. It implies limitations on the bandwidth and the maximal length of the bus (e.g. 1 Mbps for 40 meters).

2.4. IEEE 802.11 MAC protocol

IEEE 802.11-2012 [4] defines several standards to offer a wireless connectivity at transmission rates ranging from 11Mbps (e.g. legacy versions such as IEEE 802.11b) up to 600Mbps (IEEE 802.11n). Bandwidth increase is due to improvements at the physical layer (OFDM, larger bandwidth, MIMO transmissions, etc.). Next generation physical layers are expected to increase data rates beyond 600Mbps (cf. IEEE802.11ac).

The fundamental medium access in IEEE 802.11 is a Distributed Coordination Function (DCF) which is a distributed random access scheme based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Additional protocols are defined to meet specific requirements but all use the service provided by DCF. For instance, the Point Coordination Function is a centralised protocol where an Access Point (AP) provides a contention-free medium access. The Hybrid Coordination Function introduces Quality of Service (QoS) either in a distributed manner (EDCA protocol) or in a centralised manner (HCCA protocol). In this work, we firstly concentrate on the basic DCF medium access to characterise the wireless medium access. In future work, it will be interesting of course to analyse how real time applications can leverage the QoS capabilities of HCF using EDCA or HCCA.

In DCF mode, a station performs carrier sensing to detect ongoing communications. If the channel is free for a period of time called Distributed InterFrame Space (DIFS), it transmits its frame immediately. If the channel is sensed busy, it defers its transmission until the end of the current transmission. Then, the station selects a random backoff \( b \) following an exponential backoff scheme. If the medium is idle for a DIFS period of time, the backoff is decremented every aSlotTime duration. The backoff interval time is decremented as long as the channel is idle and is frozen as the node detects a transmission. At the end of this transmission, when the channel remains idle during DIFS, decetration resumes. As \( b \) reaches zero, the transmission is attempted immediately by the station.

After a successful transmission, the receiver sends an ACK after a duration called Short Inter Frame Space (SIFS). As SIFS is shorter than DIFS (DIFS = SIFS + 2 × aSlotTime), there is no station that sees the channel idle until the end of the ACK transmission. If no ACK is received by the transmitter after an Extended Interframe Space (EIFS = SIFS + DIFS + ACKTxTime\(^1\)), the transmission is attempted again.

A new backoff \( b \) is then uniformly chosen in the range \([0, w − 1]\) where \( w \) is the contention window. This window depends on the number of failed attempts experienced by the current transmission. At the first attempt \( w \) is equal to the minimum contention window \( CW_{\text{min}} \). Each unsuccessful transmission involves the multiplication of \( w \) by 2 until a maximum value of \( CW_{\text{max}} \) is reached.

This Basic Access mechanism can be extended by the RTS/CTS message exchange to avoid the hidden terminal problem. In our architecture, all wireless transmitters are fixed and positioned in such a way that the hidden terminal problem doesn’t occur. Thus, there is no motivation for implementing RTS/CTS mechanism in our case study.

\(^1\) with ACKTxTime the transmission time of an ACK at the lowest mandatory transmission rate
In this paper, all wireless nodes (gateways, pure wireless emitters) function in ad hoc mode using DCF medium access protocol following the specification of the OFDM-PHY layer of 802.11g (20 MHz channel spacing). Table 2 gives the main timing parameters of the protocol. Since WiFi access points are static, we can consider that they are located at a distance where they can operate at the highest rate of 54 Mbps. We assume proper channel assignment has been performed so as to mitigate inter-node interference. In this ideal case study, transmissions are error-free.

The following derivations are illustrated using the example in Figure 1.

A possible scenario is depicted in Figure 3. A subset of the system is shown, i.e. CAN buses 1 and 2 and the wireless network. Rising arrows indicate the instant when frames are ready for transmission on the corresponding medium.

Let’s focus on the hybrid flows $fH_1$, $fH_2$, $fH_3$, $fH_4$. All of them have their source node on CAN bus 1.

- One frame from hybrid flow $fH_3$ becomes ready for transmission on CAN bus 1 while one frame of flow $fC_1$ is being transmitted. At time $t + 0.135 \, ms$, CAN bus 1 becomes idle. The frame from $fH_3$ is transmitted since it is the only pending frame. It is fully received by $Gw_1$ at time $t + 0.270 \, ms$. It is then encapsulated by $Gw_1$ in an IEEE802.11 frame which becomes ready during the transmission of a frame from $fW_2$. After this transmission, frames from $fH_3$ and $fW_3$ compete for the medium. They select the same random backoff. Consequently, they collide. At the end of this collision the frame from $fH_3$ gets access to the medium thanks to a random backoff which is smaller than the one of $fW_3$. Then, the frame from $fH_3$ is received by $Gw_2$, de-encapsulated and transmitted on CAN bus 2 which is idle. This transmission is achieved at time $t + 0.8 \, ms$.

- Similarly, frames from $fH_4$, $fH_2$ and $fH_1$ are generated in this order. With respect to their priorities, they are transmitted in the same order on CAN bus 1. Each of them is then encapsulated in a separate IEEE802.11 frame. These frames have to share the wireless medium with frames from pure IEEE802.11 flows $fW_1$ and $fW_3$. In this scenario, the selected random backoffs do not lead to any collision. Frames from $fH_2$ and $fH_1$ are transmitted last. Finally, they are de-encapsulated by $Gw_2$ and transmitted on CAN bus 2. They are received by their destination nodes at times $t + 1.2 \, ms$ and $t + 1.335 \, ms$.

### 3.2. Grouping for a better use of the wireless medium

This basic strategy is very simple but it doesn’t use efficiently the wireless medium. Indeed, encapsulating one single CAN frame (let’s say 10 bytes) in an IEEE802.11 frame (up to 2312 bytes of payload) generates a significant overhead: typically, assuming no collision and a null backoff, the overall time needed for one frame with a 10 bytes payload (DIFS + transmission of the frame + SIFS + transmission of ACK) is obtained from formula 2:

$$34 + \left(20 + 4 \left[\frac{22 + 8(34 + x)}{216}\right]\right) + 16 + 24 = 102 \, \mu s$$
Given a payload of 50 bytes, the same sequence needs 110 µs.

A straightforward solution which limits this overhead consists in encapsulating more than one CAN frame in one IEEE802.11 frame. We denote \( N_{l,m} \), the exact number of CAN frames which are encapsulated in an IEEE802.11 frame at a given gateway \( G_{W1} \) when its destination is gateway \( G_{Wm} \).

Let’s come back to the example in Figure 1. Let’s assume that \( G_{W1} \) encapsulates three CAN frames in each IEEE802.11 frame with destination \( G_{W2} \) (\( N_{1,2} = 3 \), while all the other \( N_{l,m} \) are equal to 1). We recall that \( f_{H1} \) is sent to \( G_{W3} \). Thus, only frames from \( f_{H3}, f_{H2} \) and \( f_{H1} \) can be grouped together. The impact of this strategy on the example is shown in Figure 4. \( G_{W1} \) receives frames from \( f_{H3}, f_{H2} \) and \( f_{H1} \) at times \( t + 0.270, t + 0.735 \) and \( t + 0.870 \), respectively. Then, it encapsulates the three CAN frames in one IEEE802.11 frames (called \( f_{HG1} \) in Figure 4). This frame is transmitted on the wireless medium after a DIFS, since the medium is idle and there are no other pending frames. \( G_{W2} \) receives the frame and de-encapsulates the three CAN frames. Finally, these frames are transmitted on CAN bus 2 in their order of priority.

On this very simple example, the grouped strategy reduces the number of wireless frames (from 7 to 5) and there are no more collisions (there are never two pending frames at the same time).

3.3. Timers to decrease the delay of hybrid flows

The drawback of the grouped strategy is that it can delay some frames of hybrid flows that are combined together at their source gateway. In the example in Figure 4, the \( f_{H3} \) frame has to wait until the arrival of the \( f_{H1} \) frame. This delay can be very large. Considering the same example, let’s assume that \( G_{W1} \) encapsulates two CAN frames instead of three in each IEEE802.11 frame with destination \( G_{W2} \). A possible scenario is depicted in Figure 5. Only the events on CAN bus 1 and gateway \( G_{W1} \) are shown. The two first frames from hybrid flows \( (f_{H3} \) and \( f_{H2} \) ) are received and encapsulated by \( G_{W1} \) at time \( t + 0.6 \) ms. The third frame (from hybrid flow \( f_{H1} \) ) has to wait in \( G_{W1} \) for the arrival of another frame from an hybrid flow. On this example, the time elapsed between the generation of the frame from \( f_{H1} \) and its encapsulation in an IEEE802.11 frame is 7.8 ms. Since the critical delay of \( f_{H1} \) is 8 ms, this frame has no chance to respect its deadline.

One solution to overcome this problem is to upper bound the waiting time of a CAN frame in a gateway. It can be implemented by associating to each hybrid flow \( f_{Hj} \) a maximum waiting time \( W_{max,j} \) in its source gateway \( G_{Wj} \). When a frame from \( f_{Hj} \) arrives at its source gateway \( G_{Wj} \), two situations may occur:

1. There are already \( N_{l,m} - 1 \) pending CAN frames in \( G_{Wj} \) with the same destination CAN bus as the frame from \( f_{Hj} \); these \( N_{l,m} \) CAN frames are immediately encapsulated in an IEEE802.11 frame which is then ready for transmission.

2. There are less than \( N_{l,m} - 1 \) pending CAN frames in \( G_{Wj} \) with the same destination CAN bus as the frame from \( f_{Hj} \); a timer with duration \( W_{max,j} \) is started. If the frame from \( f_{Hj} \) is still waiting when the timer expires, the frame is immediately encapsulated in an IEEE802.11 frame with all the other pending CAN frames having the same CAN destination bus.

Figure 6 shows the impact of these timers on the scenario in Figure 5. In this example, the maximum waiting time of a frame in gateway \( G_{W1} \) is 2 ms for flows going to \( G_{W2} \), and \( N_{1,2} = 2 \). \( f_{HG1} \) is generated because there are two pending frames with CAN bus 2 as destination. \( f_{HG2} \) is generated because the timer which is associated with the frame of flow \( f_{H1} \) has expired. Thus \( f_{HG1} \) encapsulates two CAN frames, while \( f_{HG2} \) encapsulates only one. This strategy obviously favours a timely transmission of the \( f_{H3} \) frame.

3.4. Summary of the proposed strategies

All previously introduced strategies (basic, grouped, timed) can be characterised by the following parameters:

- The maximum number \( N_{l,m} \) of CAN frames which can be encapsulated by each gateway \( G_{Wl} \) in one IEEE802.11 frame,

- The maximum waiting time \( W_{max,j} \) of a frame of each hybrid flow \( f_{Hj} \) in its source gateway.

Thus, we have:
4. Bridging performance analysis

A quantitative analysis of the proposed bridging strategies has been conducted. This analysis is based on simulations. Therefore, a home-made simulation tool has been developed using QNAP2 [9].

In the following paragraphs, we consider two network configurations:

- the illustrative configuration of Figure 1
- a more complex configuration introduced in Section 4.2.

4.1. Illustrative configuration (CS1)

The first configuration is depicted in Figure 1 and the features of the flows are summarised in Table 1. All proposed encapsulation strategies (basic, grouped, timed) are evaluated as presented in Table 3.

We first look at the percentage of pure CAN and hybrid frames that miss their deadline. Whatever strategy is chosen, pure CAN frames never miss their deadline. Similarly, with the basic and timed strategies, hybrid frames never miss their deadline. Conversely, with the grouped strategies, some hybrid frames miss their deadline (0.2 % with G1, 0.5 % with G2). These results are not surprising. Indeed the overall configuration is lightly loaded. Thus a missed deadline is impossible for pure CAN flows and it can arise for hybrid flows only when the delay between the two CAN buses is high. This delay includes the waiting time at the source gateway, the IEEE802.11 delay (very unlikely to be high since the wireless medium is lightly loaded) and the waiting time at the destination gateway (short de-encapsulation time). Then a missed deadline will occur only when the waiting time in the source gateway is high. This can be the case with the grouped strategies, as shown in section 3.2. This result confirms the flaws introduced by the grouped strategies. These strategies are not further considered.

A second result concerns the frames which are transmitted on the wireless medium. Given the Poisson traffic distribution of pure wireless flows, their average number of frames is steady. Conversely, the number of wireless frames from hybrid flows depends on the encapsulation strategy which is used. Table 4 compares the number of wireless frames of hybrid flows to the number of frames of one of the pure wireless flows (e.g. $fW_1$). For the basic strategy, these two number of frames happen to be the same because of the period allocation of hybrid flow. Without surprise, this ratio decreases with the increase of the maximum delay at the gateway.

<table>
<thead>
<tr>
<th>B</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.88</td>
<td>0.77</td>
<td>0.7</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Table 4. CS1: load of hybrid wireless frames

Table 5 shows the percentage of collisions for each strategy. This percentage is very low (never more than 0.4 %). This is a consequence of the light load of the wireless medium. Moreover, this percentage decreases when the value of the maximum delay in the gateway increases (the overall number of wireless frames decreases).

<table>
<thead>
<tr>
<th>Col. (%)</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.375</td>
<td>0.358</td>
<td>0.343</td>
<td>0.335</td>
<td>0.328</td>
</tr>
</tbody>
</table>

Table 5. CS1: wireless collisions

Table 6 shows the average delay of pure wireless flows. This delay decreases when the value of the maximum delay in the gateway increases. This is also a consequence of the reduction of the overall number of wireless frames.

<table>
<thead>
<tr>
<th>Delay (µs)</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
</tr>
</thead>
<tbody>
<tr>
<td>135</td>
<td>133</td>
<td>131.7</td>
<td>130.9</td>
<td>129.6</td>
</tr>
</tbody>
</table>

Table 6. CS1: average delay of pure wireless flows

The results confirm the qualitative analysis of section 3. However, the small illustrative configuration considered in this section is too lightly loaded for a significant quantitative analysis of the different strategies. Such an analysis is conducted in the next section.

4.2. More complex configuration (CS2)

Figure 7 introduces the CS2 configuration.

It includes four CAN buses. Each CAN bus 1, 2 and 3 is the source of 8 hybrid flows. Bus 4 is the destination of all these hybrid flows. Table 7 summarises the parameters of these hybrid flows. For the sake of simplicity, only the
We have conducted three sets of experiments:

- in the first one, we consider the basic encapsulation strategy,
- in the second one, we consider a timed encapsulation strategy where all hybrid flows are allocated the same maximum waiting delay,
- in the last one, we consider another timed encapsulation strategy where each hybrid flow is allocated a dedicated maximum waiting delay following $W_{max,j} = (4.1 - (j \times 0.1)) \text{ ms}$.

Table 9 summarises the simulated strategies.

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sum f_{H_i}$</td>
<td>4.8</td>
<td>2.8</td>
<td>1.9</td>
<td>1.4</td>
<td>1.2</td>
<td>1.6</td>
</tr>
<tr>
<td>$f_{H_i}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$% \text{ Col}$</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>148</td>
<td>122</td>
<td>114</td>
<td>111</td>
<td>109</td>
<td>112</td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$% \text{ Col}$</td>
<td>0.9</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>172</td>
<td>138</td>
<td>129</td>
<td>124</td>
<td>122</td>
<td>125</td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$% \text{ Col}$</td>
<td>1.4</td>
<td>0.6</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>205</td>
<td>158</td>
<td>147</td>
<td>141</td>
<td>138</td>
<td>142</td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$% \text{ Col}$</td>
<td>2.2</td>
<td>1.0</td>
<td>0.8</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>250</td>
<td>185</td>
<td>169</td>
<td>162</td>
<td>158</td>
<td>163</td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$% \text{ Col}$</td>
<td>1.6</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.2</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>221</td>
<td>199</td>
<td>189</td>
<td>184</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$% \text{ Col}$</td>
<td>2.5</td>
<td>2.0</td>
<td>1.8</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>279</td>
<td>242</td>
<td>227</td>
<td>220</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td>AvgD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$% \text{ Col}$</td>
<td>2.7</td>
<td>2.5</td>
<td>2.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_{W}$</td>
<td>287</td>
<td>274</td>
<td>294</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10 also shows the percentage of collisions and the average delays of pure wireless flows for each strategy and each simulated scenario (i.e. number of pure wireless flows). Empty values mean that the wireless network is overloaded and transmission delays diverge. It can be noticed that these percentages of collisions ($\% \text{ Col}$) and delays (AvgD) are deeply linked to the number of contending IEEE802.11 frames.

Thus higher values of timers give smaller delays on the wireless network. However, they also increase the waiting time of hybrid frames in gateways. It means that a trade-off has to be found between the waiting delays in the gateways and the delays on the wireless network.
In order to better capture this trade-off, Table 11 presents the rates of hybrid frames which miss their deadlines. More precisely, each value in Table 11 is the average number of hybrid frames that miss their deadlines when $10^6$ of such frames are generated and transmitted. Such missed deadline rates are compatible with soft real-time data transmissions.

<table>
<thead>
<tr>
<th>B</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>97</td>
<td>21</td>
<td>19</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>200</td>
<td>257</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

number of missed deadline every $10^6$ Frames

Table 11. CS2: Missed deadlines for hybrid flows

We can conclude that the timed strategies outperform the basic one. Indeed, the basic strategy cannot cope with more than 4 pure IEEE802.11 flows: the system doesn’t converge. Conversely the timed ones are still working with 6 or 7 pure IEEE802.11 flows. These results are compatible with soft real-time data transmissions.

The value of the timer also has an impact on the number of missed deadlines. In the configuration studied in this paragraph, the best value is 3 ms for all the hybrid flows (less than half of their period). It should be noticed that strategy T5 gives slightly larger numbers of missed deadlines. The idea behind this strategy was to limit the waiting time in gateways for the hybrid flows with the lowest priorities, since they can experiment larger delays on CAN buses. The results for this strategy are not convincing.

In the general case, the choice of this timers has to take into account the distribution of the delays on the wireless network as well as the distribution of the delays on the CAN buses. Then these distributions have to be combined so that the distribution of the overall delay of the flows, excluding the waiting time at the gateways, can be obtained. A maximum waiting time at the gateway could be deduced from this distribution. To the best of our knowledge, the computation of such a distribution is still an open problem.

5. Conclusion and future works

This paper studies the extension of CAN over the air for the exchange of soft real-time data. CAN buses are interconnected through a standard IEEE802.11g wireless network using CSMA/CA medium access control in ad hoc mode. The interconnection between CAN and the wireless network is done by gateways. The bridging strategy implemented in these gateways is a key issue in such an architecture. We show on a case study that the best strategy consists in encapsulating a group of CAN frames in each IEEE802.11 frame while bounding the waiting time of each CAN frame at the ingress gateway. With such a strategy, the number of missed deadlines can be kept very small, provided that the wireless channel is reliable (we assume no transmission errors except collisions).

In future work, we will investigate the impact of a lossy wireless link where wireless nodes transmit at different rates. Another improvement is to derive analytically the timers of the hybrid flows. Lastly, in order to cope with hard real-time data, we will investigate more deterministic wireless protocols (HCCA, WirelessHART, ...).

References