



HAL
open science

Strategies for the Interconnection of CAN buses through an Ethernet switch

Abdelaziz Ahmed-Nacer, Katia Jaffrès-Runser, Jean-Luc Scharbarg, Christian Fraboul

► **To cite this version:**

Abdelaziz Ahmed-Nacer, Katia Jaffrès-Runser, Jean-Luc Scharbarg, Christian Fraboul. Strategies for the Interconnection of CAN buses through an Ethernet switch. IEEE International Symposium on Industrial Embedded Systems - SIES 2013, Jun 2013, Porto, Portugal. pp.77-80. <hal-01150343>

HAL Id: hal-01150343

<https://hal.science/hal-01150343v1>

Submitted on 11 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 12499

To link to this article : DOI :10.1109/SIES.2013.6601474
URL : <http://dx.doi.org/10.1109/SIES.2013.6601474>

To cite this version : Ahmed Nacer, Abdelaziz and Jaffrès-Runser, Katia and Scharbag, Jean-Luc and Fraboul, Christian *Strategies for the Interconnection of CAN buses through an Ethernet switch*. (2013)
In: IEEE International Symposium on Industrial Embedded Systems - SIES 2013, 19 June 2013 - 21 June 2013 (Porto, Portugal).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Strategies for the Interconnection of CAN buses through an Ethernet switch

Abdelaziz Ahmed Nacer, Katia Jaffres-Runser, Jean-Luc Scharbag and Christian Fraboul
 IRIT - ENSEEIHT, Toulouse University
 2, rue Charles Camichel 31000 Toulouse - France
 {aahmedna, katia.Jaffres-Runser, Jean-Luc.Scharbag, Christian.Fraboul}@enseeiht.fr

Abstract—This paper discusses the use of an Ethernet switch to interconnect remotely located fieldbuses, e.g. CAN. Mastering delays of flows is a key issue in such architectures. The focus of this paper is to propose and analyze bridging strategies for the gateways between CAN and the Ethernet switch. Our previous studies have highlighted the benefits of an optimal design of encapsulation policies at ingress gateways to the switch. In this paper, we show that regulating CAN traffic at the egress gateways is essential as well to provide timely transport of CAN flows.

I. INTRODUCTION

Fieldbuses (e.g. CAN, Profibus) have been developed in the context of real-time applications that have specific communication requirements, in particular in terms of timing constraints. However, the amount of information that is nowadays exchanged in such systems has been increasing steadily and is reaching the limits of traditional fieldbuses, especially in terms of bandwidth. Switching from fieldbuses to Ethernet is a classical trend in real-time systems due to the wide acceptance of the Ethernet standard and its evolution toward a more predictable switched architecture.

The network architecture considered in this paper interconnects a set of CAN buses through an Ethernet switch. The goal is to increase the available bandwidth while keeping the well established CAN technology. We argue that the key point to achieve a timely behavior of the flows in the network is to carefully define the bridging strategies of the gateways interconnecting the fieldbuses with the switch.

This paper proposes a set of bridging strategies both for the CAN to Ethernet and the Ethernet to CAN gateway directions. Previous studies have highlighted the benefits of an optimal design of encapsulation policies at ingress gateways to the switch. In this paper, we show that regulating CAN traffic at the egress gateways is essential as well to provide timely transport of CAN flows.

The paper is organized as follows. Section II summarizes the considered network architecture. Section III proposes different bridging strategies and section IV shows their impact on an example configuration. Section V concludes and gives some directions for future work.

II. HYBRID NETWORK ARCHITECTURE

The purpose of this study is to interconnect several CAN buses with a switched Ethernet network. First, CAN and switched Ethernet are presented in Section II-A. Then the targeted network architecture is detailed in Section II-B.

A. Protocol descriptions

A CAN bus (Controller Area Network) [1] is a serial communication bus originally suited for in-vehicle communication but also used in other areas such as aerospace, or industrial automation. CAN bus is a message-based protocol. A CAN frame has neither a destination nor a source address. The protocol addressing system is based on messages identifiers. The CAN frames are broadcasted on the bus. When frames are received, station get the frame they are interested in by filtering out the identifiers. The identifier field represents the priority of the frame: the smaller the identifier, the higher the frame priority. CAN MAC is based on a Non-Destructive Bit Wise Arbitration process.

SOE	11-Bit identifier	RTR	IDE	r0	DLC	0...8 Bytes Data	CRC	ACK	EOF	IFS
-----	-------------------	-----	-----	----	-----	------------------	-----	-----	-----	-----

Fig. 1. Standard CAN frame format

Fig. 1 depicts a standard CAN frame format. The following fields are important to the rest of the paper :

- Identifier is an 11-bit field that identifies the data contained in the frame.
- DLC is a 4-bit field giving the length of the data.
- Data field contains the frames payload (up to 8 bytes).

The CAN frame length L is calculated following the work of Davis et al. [2] which integrates the worst case overhead induced by bit stuffing using $L = 47 + 8 \times DLC + \lfloor \frac{34 + 8 \times DLC}{4} \rfloor$.

A Full Duplex switched Ethernet network [3] is a collision-free network. Collisions on the physical link are eliminated since each equipment of the network is only connected to a switch by a Full Duplex link.

B. Interconnection architecture

Fig. 2 presents the illustrative network example studied in this paper. The network consists of three CAN buses interconnected via 3 gateways (G1, G2, G3) and one Ethernet switch. Each gateway interconnects a CAN bus to the switch. In this first study, we assume a basic Ethernet network composed of a single switch but we aim at investigating in the future a more complex network where several switches relay distant real-time data.

The switch includes a set of input and output ports. Only output ports have FIFO queues for storing frames before

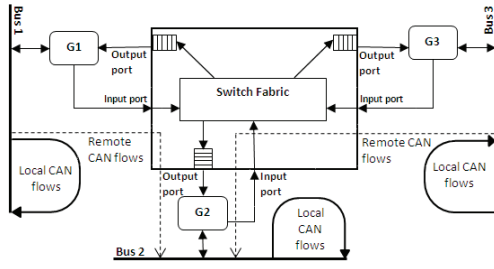


Fig. 2. Illustrative hybrid network architecture

transmitting them to the next network component. When a frame is received by the switch, the output transmission queue is determined and the frame is enqueued waiting for the transmission on the output link. The switching tables are set up statically and no spanning tree process is used since all flows are known in advance in this type of architecture. This is a reasonable assumption in our context.

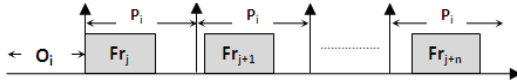


Fig. 3. CAN flow characterization

A CAN flow F_i , as depicted in Fig. 3, is a succession of CAN frames Fr_j characterized by the same period P_i , the same identifier (i.e. priority), and an offset O_i . Two types of flows will be considered in this paper:

- Local CAN flows: frames of these flows are sent and received on the same bus.
- Remote CAN flows: frames of these flows are transferred from one bus to another one.

III. GATEWAY DESIGN

Fig. 4 summarizes the transmission of a remote CAN frame via the Ethernet switch. The frame is transmitted on its source CAN bus and it arrives in its ingress gateway. It is transformed into an Ethernet frame and transmitted to its egress gateway. It is then transformed again into a CAN frame and transmitted to its CAN destination bus. Formally, an ingress gateway captures remote CAN frames and introduces them into the switched Ethernet network. Conversely, an egress gateway captures switched Ethernet frames and introduces them into the destination CAN bus it is connected to. The end-to-end delay of a frame highly depends on the transformation processes which are implemented at the gateways both in the ingress and the egress directions. In the following sections, we propose different strategies for both the ingress and the egress transformations.

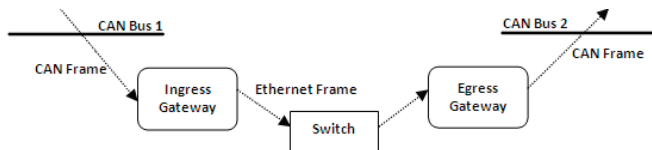


Fig. 4. Flows transmission over a hybrid network

A. Ingress gateway

As previously mentioned, CAN is a message based protocol (no address field). Conversely, Ethernet uses source and destination addresses for the identification of frames. Thus, an efficient and handy strategy consists in encapsulating CAN frames in the data field of Ethernet frames [4][5]. This process is illustrated in Fig. 5. The remaining problem is to determine how many CAN frames are encapsulated in each Ethernet frame. Indeed, a CAN frame includes at most 8 data bytes, while an Ethernet frame contents up to 1500 bytes. Several possibilities have been evaluated in [4][5] in the context of vintage Ethernet.

1) *The one-to-one strategy*: The gateway encapsulates each remote CAN frame it receives in a separate Ethernet frame and this frame is immediately forwarded to the switch. Obviously this strategy minimizes the waiting time of remote CAN frames in their ingress gateways. The drawback of this strategy is that it generates a large overhead on Ethernet (Ethernet frames with few data). Thus the Ethernet bandwidth is not used efficiently.

2) *The n-to-one strategy*: The gateway encapsulates exactly a given number N of remote CAN frames in each Ethernet frame. Clearly, it will reduce the number of Ethernet frames and increase their size. Thus the use of Ethernet bandwidth is improved. The main drawback of this strategy is that it creates a waiting delay at the ingress gateway for most of the remote CAN frames. Indeed, a remote CAN frame has to wait until there are $N - 1$ other pending remote CAN frames going to the same egress gateway.

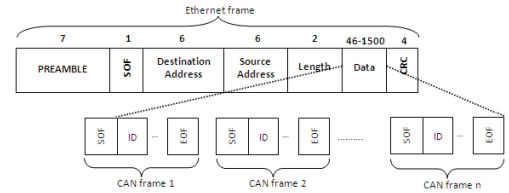


Fig. 5. Frame encapsulation

3) *Timed n-to-one strategy*: This encapsulation strategy is a trade-off between the two previous ones. The aim is to use efficiently the Ethernet bandwidth while bounding the delay incurred by remote CAN frames in their ingress gateways. This goal is achieved by associating a timer t_i to each remote CAN flow i . This timer t_i is started as soon as one frame of flow i arrives in the ingress gateway. An Ethernet frame is built with the pending CAN frames and forwarded to the switch as soon as the timer associated to one pending CAN frame elapses or there are N pending CAN frames. It has been shown in [4][5] that this strategy outperforms the other ones in the context of vintage Ethernet.

B. Egress gateway

Each time a gateway receives an Ethernet frame, it decapsulates the included remote CAN frames in it and transmits them on the CAN bus. Two approaches are proposed in this paper:

- in the first one (immediate forwarding), remote CAN frames are ready for transmission on the CAN bus as soon as they are decapsulated from the Ethernet frame.

- in the second one (traffic shaping), remote CAN frames can be delayed on their egress gateway in order to regulate the traffic on the CAN bus.

The following paragraphs detail these two approaches. They will be compared on a case study in section IV.

1) *Immediate forwarding strategy*: This approach is illustrated in Fig. 6, assuming one single CAN flow F_i from CAN bus 1 to CAN bus 2. The first frame Fr_1 from F_i is generated at time t_0 . It is delayed by competing frames on CAN bus 1 and its transmission on this bus is achieved at t_1 . Assuming a one to one strategy at the ingress gateway, Fr_1 is encapsulated in an Ethernet frame and immediately forwarded to the gateway of CAN bus 2. Fr_1 is then decapsulated and immediately ready for transmission on CAN bus 2 (at time t_2). It is delayed by competing frames and finishes transmission on CAN bus 2 at time t_3 . The second frame Fr_2 from F_i is generated at time $t_0 + P_i$. It is immediately transmitted on CAN bus 1 since there are no competing frames at this instant. Then it is encapsulated in an Ethernet frame and forwarded to the gateway of CAN bus 2. It is decapsulated and immediately ready for transmission (at time t_4). It is immediately transmitted, since there are no competing frames at this instant. Similarly the third frame Fr_3 of F_i is generated at $t_0 + 2P_i$, transmitted on CAN bus 1, forwarded to CAN bus 2 where it is finally transmitted.

We can see on the example of Fig. 6 that Fr_1 and Fr_2 create a small burst of traffic on the destination bus. Indeed, the time elapsed between the arrival of Fr_1 and Fr_2 at CAN bus 2 is smaller than the period of flow F_i . In other words, the immediate forwarding of remote CAN frames by their egress gateway can reduce significantly the duration between the instants where two consecutive frames of a given CAN flow get ready on their destination bus. Moreover, such a burst of traffic has an impact as well on the delay of the local flows of CAN bus 2 which can be increased. This is even the case for the worst case delay of these flows since its derivation is a function of the periods (or generalized periods) of all flows transmitted on the destination CAN bus [2].

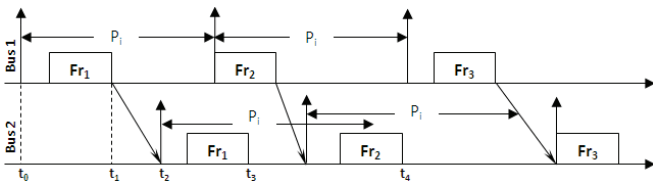


Fig. 6. The immediate forwarding strategy

2) *Traffic shaping strategy*: The aim of this strategy is to guarantee a minimum duration between any two consecutive frames of a remote CAN flow on their destination CAN bus. In order to achieve this goal, the egress gateway computes, for each decapsulated remote CAN frame Fr_j , the time elapsed since the arrival of the previous frame of the same flow in the gateway. If this time is at least the period of the flow, the frame Fr_j is immediately ready for transmission. Otherwise Fr_j is delayed so that the period of the flow is guaranteed at the egress gateway level. Fig. 7 shows the impact of this strategy on the example of Fig. 6. The second and third frames of F_i are postponed by the gateway of CAN bus 2 in order to guarantee at least P_i between any two consecutive frames of

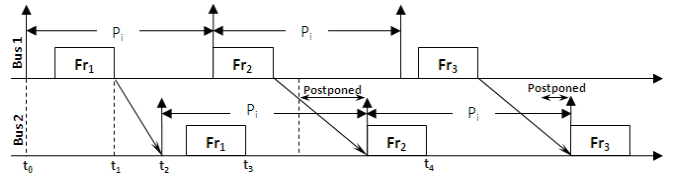


Fig. 7. The traffic shaping strategy

F_i . Thus no burst on flow F_i is created on CAN bus 2 and its local flows may not be delayed by the burst of F_i anymore. As a consequence, the worst case delay of the local flows is not increased anymore since the period of remote flows is maintained. The next section illustrates the impact of the egress gateway strategies on the delays of flows.

IV. IMPACT OF THE EGRESS GATEWAY POLICIES

The network architecture depicted in Fig. 2 is considered. 42 flows are transmitted on this architecture as given in Table I. 40 of them are local flows: 15 on CAN bus 1, 15 on CAN bus 3, and 10 on CAN bus 2. One remote flow originates on bus 1 while the other one is generated on bus 3. Both of them have their destination on bus 2. These remote flows have the lowest priority on their source bus and the highest one on their destination bus. In the following paragraphs, we show the impact of traffic shaping on a very specific scenario. Then we measure this impact on a random set of scenarios by simulation. At the ingress gateway, the one-to-one strategy is sufficient since there is at most one remote flow generated by each bus.

Identifiers	P_i (ms)	Length (bits)	Src bus	Dest bus	Type
1, 3, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29	4	135	1	1	Local
5	2	75	1	1	Local
38, 39, 40, 41, 42, 43, 44, 45, 46, 47	2	135	2	2	Local
2, 4, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30	4	135	3	3	Local
6	2	75	3	3	Local
31	2	135	1	2	Remote
32	2	135	3	2	Remote

TABLE I. NETWORK APPLICATION

A. Traffic shaping on a selected scenario

Let's assume that the offsets of all the flows are null. It means that the first frame of each flow is ready for transmission at time 0. Fig. 8 shows the scheduling of frame transmissions on the three CAN buses for two different cases. The top part of Fig. 8 considers immediate forwarding strategy at $G2$ while the bottom part considers the traffic shaping strategy. For both cases, the 15 local frames of bus 1 and the remote CAN frame are transmitted in this order between time 0 and time $2.1ms$. A similar scheduling occurs on bus 3. Thus remote frames 31 and 32 are ready for transmission on bus 2 at time $2.13ms$. On this bus, the 10 local frames ready at time 0 are transmitted between time 0 and $1.135ms$. At time $2ms$, 10 other frames from the local flows of the bus 2 are ready for transmission. The highest priority one (flow 38) is transmitted at time $2ms$, since frames 31 and 32 are not ready yet. At the end of the transmission of frame 38, frames 31 and 32 have arrived and they are transmitted since they are pending frames

with the highest priority. These transmissions are completed at time $2.4ms$. Meanwhile a second frame from flows 31 and 32 has arrived at $G2$. Indeed, these frames have been generated at bus 1 and 3 at time $2ms$ and they have been transmitted after frames 5 and 6 which have been generated at the same time and have a higher priority. At this point, the scheduling on bus 2 depends on the egress gateway strategy. When the immediate forwarding strategy is applied (Fig. 8-top), frames 31 and 32 are ready as soon as they are received by $G2$ at time $2.315ms$. Thus, they are transmitted between time $2.4ms$ and time $2.67ms$, delaying local frames 39 to 47. When the traffic shaping strategy is applied (Fig. 8-bottom), frames 31 and 32 have to wait at $G2$ until their period (i.e. $2ms$) has elapsed since the arrival of the previous frames of the same flows. Thus they are ready for transmission at time $4.13ms$. Thanks to this delay, frames from local flows are transmitted earlier than with the immediate forwarding strategy. We can conclude that, on this specific scenario the delay of the the second frames of 39 to 47 is decreased of $0.270ms$ as traffic shaping is applied.

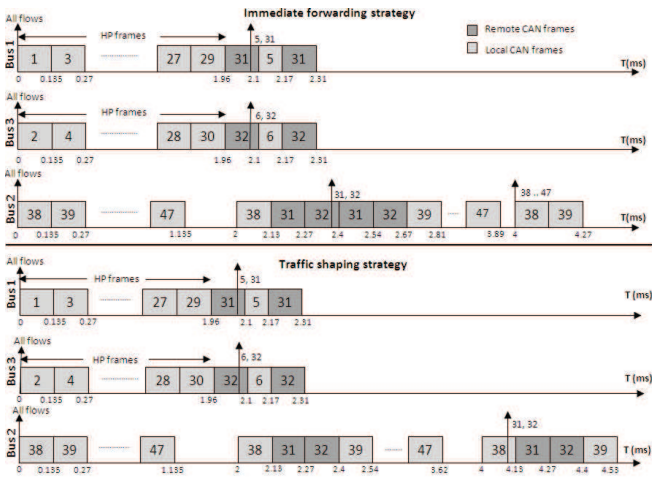


Fig. 8. Network behavior

B. Impact of traffic shaping on random scenarios

The previous paragraph shows that traffic shaping at egress gateways can have a positive impact on the delay of local flows. The next question is: is this impact significant if we consider random scenarios (which is the general case since buses are not synchronized)? A simulation campaign has been conducted in order to answer this question for the configuration considered herein.

We have modeled the network architecture of Fig. 2 using Omnet++ simulator [6]. As a first step, we have considered gateways that implement the immediate forwarding strategy described in Section III-B. As a second step, we have considered gateways that implement the traffic shaping strategy. Flow offsets O_i are randomly generated in the interval $[0, P_i]$.

Table II shows the maximum and average end-to-end transmission delays achieved by simulation for both strategies for flows traveling on CAN bus 2. Simulations show that using the traffic shaping on the egress gateway $G2$ increases maximum and average end-to-end delay of remote flows. This is due to the introduction of the waiting time at the egress gateway level. As expected, traffic shaping of remote flows

Identifiers	Max without shap.(ms)	Max shap.(ms)	Average without shap.(ms)	Average shap.(ms)
31	2.145	2.159	0.440	0.518
32	2.175	2.226	0.446	0.525
38	0.543	0.542	0.200	0.199
39	0.679	0.677	0.208	0.208
40	0.875	0.815	0.219	0.219
41	0.951	0.943	0.232	0.232
42	1.124	1.085	0.248	0.248
43	1.292	1.222	0.268	0.268
44	1.451	1.359	0.293	0.293
45	1.610	1.495	0.323	0.323
46	1.747	1.631	0.363	0.363
47	1.815	1.632	0.415	0.414

TABLE II. MAXIMUM AND AVERAGE END-TO-END DELAYS

reduces the maximum observed delay of local CAN flows (e.g. from $1.815ms$ to $1.632ms$ for flow 47). This corresponds to the conclusion of paragraph IV-A. Conversely, the egress gateway strategy has no impact on the average delay of the local flows. Indeed, the shaping can decrease or increase the delay of a frame from a local flow. For instance, in Fig. 8, the delay of the second frame of flow 39 decreases from $0.81ms$ to $0.54ms$, while the delay of the third frame of flow 39 increases from $0.27ms$ to $0.535ms$.

V. CONCLUSION

This paper studies the interconnection of CAN buses through a switched Ethernet network. The interconnection between CAN and Ethernet is done by gateways. The bridging strategy implemented at these gateways is a key issue in such an architecture. We propose different strategies both for the CAN to Ethernet and the Ethernet to CAN directions. We show on an example configuration the impact of the strategy of the Ethernet to CAN direction on the delay of frames. A more comprehensive study is needed in order to better understand the impact of the strategies on the delays. The goal will be to establish guidelines for the choice of the strategies, based on the kind of flows transmitted on the network. We also intend to evaluate the impact of pure Ethernet flows which could share the switched Ethernet network with CAN flows. This could imply the use of QoS mechanism at the switch level. Finally, many real-time Ethernet solutions exist (e.g. TTEthernet, Profinet,...). The introduction of such technologies in our architecture should be evaluated.

REFERENCES

- [1] "ISO International Standard 11898- Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication," 1993.
- [2] R. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller area network (can) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272.
- [3] J. Jasperneite, P. Neumann, M. Theis, and K. Watson, "Deterministic real-time communication with switched ethernet," in *Factory Communication Systems, 2002. 4th IEEE International Workshop on*, 2002, pp. 11–18.
- [4] J.-L. Scharbarg, M. Boyer, and C. Fraboul, "Interconnecting can busses via an ethernet backbone," in *FeT'05*, november 2005, pp. 206–213.
- [5] J. Scharbarg, M. Boyer, and C. Fraboul, "Can-ethernet architectures for real-time applications," in *ETFA'05*, vol. 2, 2005.
- [6] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Simutools '08*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–10.