Proc. IEEE Int. Systems Conference (SysCon 2015), Vancouver, Canada, April 2015, pp. 61-68.

# System Design Issues for Future In-Vehicle Ethernet-Based Time- and Safety-Critical Networks

Aboubacar Diarra
Robert Bosch GmbH
Body Electronics System Engineering (AE-BE/EKE)
Po Box 16 61, 71226 Leonberg, Germany
Email: aboubacar.diarra2@de.bosch.com

Armin Zimmermann
Systems and Software Engineering Group
Computer Science and Automation Department
Ilmenau University of Technology, Ilmenau, Germany
Email: armin.zimmermann@tu-ilmenau.de

***Abstract*—Timing behavior predictability is a necessary real-time requirement for future switched Ethernet-based networks in the automotive and similar domains. Schedulability of such safety- and time-critical networks has to be guaranteed. The paper proposes a system architecture for future Ethernet-based advanced driver assistance systems (ADAS) and analyzes its timing requirements on transmission and forwarding deadlines. It studies the schedulability for strict priority combined with earliest deadline first (EDF) scheduling. For that, conditions on bandwidth utilization factors guaranteeing schedulability in the whole network are formally determined.**

## I. Introduction

The development of future in-vehicle electric and electronic (E/E) architectures deals with complexity in terms of systems engineering and design. One reason is the growing number of distributed functions in vehicles and the resulting amount of network nodes and data to be processed. Consequently, meeting real-time challenges in those networks especially for advanced driver assistance system (ADAS) applications is a complex task for systems engineers.

As a specific example, more information is needed in radar object data for situation interpretation for (semi)automated driving. Close range and stereo video cameras will be needed for surround views and object detection/recognition. For that, high-resolution picture data has to be transmitted from sensors to processing electronic control units (ECUs). Other advanced sensors including light detection and ranging (LIDAR) will add to the required bandwidth.

The communication capabilities of traditional in-vehicle networking technologies such as CAN (Controller Area Network) and FlexRay are limited and will not be sufficient. The first issue is the limited bandwidth provided by CAN and FlexRay (maximum bandwidth 1Mbps for CAN and 10 Mbps for FlexRay) to handle the expected huge amount of sensor data. The second issue is their restriction to a bus topology and the corresponding transmission modes. Because a single medium is shared by all sensors in the cluster, the available bandwidth on the bus has to be shared by all connected nodes (sensors and ECUs). In addition, the half-duplex characteristic of these bus systems makes a simultaneous data transmission by all sensors impossible.

For these reasons, there is a need for affordable high-bandwidth networking solutions in future vehicles. There is a growing interest in Ethernet [1] as a communication technology which could solve the problem. The base technology has the advantage of being commercially available, but it is originally not intended for time-critical data. Thus, as critical messages need to be received within certain deadlines, a schedulability analysis is crucial to understand which bandwidth resources are needed in switches and end-stations.

Schedulability evaluation of switched Ethernet networks has been carried out in the literature. In [2], deadlines are divided on different segments in a switched Ethernet network. Selection of packets for transmission is done via earliest deadline first (EDF) scheduling. Frames are periodically transmitted with a method termed Flexible Time-Triggered Ethernet (FTT-E). This work has been extended in [3] proposing a symmetric deadline partitioning to analyze system schedulability. This approach is further improved in [4], in which an asymmetric deadline partitioning method is set up to evaluate system schedulability. In [2]–[5], the analyzed mechanism (FTT-E) is based on elementary cycles dedicated to messages transmission, and schedulability analysis is done for a network comprising only one switch where the maximum number of hops is two. The approach in these references is thus not exhaustive for a more complex network with several switches like the proposed topology in this paper.

Similar issues are considered in avionic systems, for instance [6] deals with schedulability analysis of an AFDX (avionic full duplex Ethernet) based system. Transmission selection is done via EDF, but schedulability evaluation is done for an end node. In [7], the worst-case response time is analyzed for real-time periodic frames and mean-time response time for time critical aperiodic frames. Transmission selection is done via Rate Monotonic Scheduling. However, the study is done for a single frame only, while in our case it should be done for a message fragmented into several frames.

In [8], a use-case in which frames are transmitted following EDF is presented. However, the end-to-end delay is analyzed without any schedulability test. [9] does a schedulability analysis for first-come-first-served based data handling in switches and end-stations. No sufficient condition on bandwidth utilization factor or on minimum needed bandwidth at a switch output port to achieve schedulability is given though. [10] examines schedulability on a traffic handled via CBSA (Credit Based Shaper Algorithm), but it is also done for only one switch and not at the network level.

The literature analysis shows that schedulability analysis is a useful tool for Ethernet-based time- and safety-critical applications, especially for automotive applications. This paper proposes an architecture for future in-vehicle networks and provides a method to predict its timing behavior analytically. Thus it can be applied early in the system design process of such a complex system without testing it expensively. In addition, a good part of the available literature deals with schedulability of FTT-E. However, because of its master-slave mechanism, it is complex to implement it as an in-vehicular network that has to be built on several switches. Schedulability analysis of FTT-E networks thus does not help to solve the problem for networks of the type proposed in this paper. Moreover, unlike the above-mentioned related work, another novelty of this paper is to bring a more exhaustive approach to evaluate schedulability on a network based on several switches, which are built on a tree architecture, instead of at switch level. Moreover, deadlines are considered here for a group of frames and not only for a single frame, because of fragmentation.

The paper is structured as follow: the next section gives an overview of the targeted application communication architecture and its real-time requirements. Section III deals with the schedulability evaluation approach and presents obtained results. Finally, some conclusions are given.

## II. System Overview and Requirements

Future in-vehicle E/E architectures are built on four main domains: infotainment, driver assistance, chassis and safety, and power train (c.f. Figure 1). This paper focuses on driver assistance including sensors data fusion. This domain is typically based on an ADAS ECU in charge of processing data from various sensors such as cameras, radars, and LIDAR. For reliability, an additional fail-back ADAS ECU is implemented to achieve fault-tolerance by structural redundancy. Both the ADAS ECU as well as the fail-back one are realized by individual processing entities (micro controller or FPGA) and switches (c.f. Figure 2). Sensor data is sent via Ethernet in our application architecture as stated above. Critical messages have to be transmitted from sensors to application processing entities which are in charge of processing them and to transmit resulting control frames to chassis and safety domain functions for braking system control, for instance. ADAS ECU and fail-back ADAS ECU exchange control data between them for an exhaustive processing because they are linked to different sensors.

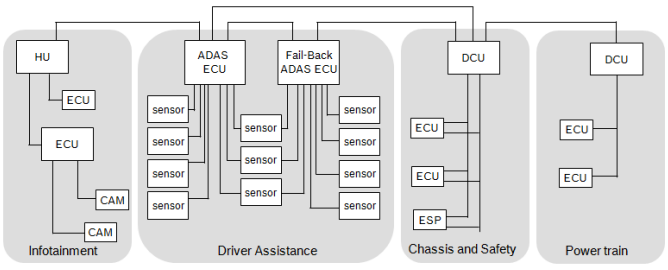



Fig. 1: In-vehicle E/E Architecture Example

### A. Real-Time Requirements

Figure 2 shows the targeted system architecture. Our study concentrates here on communication of one of the ECUs and should be transferrable to both ADAS ECUs because of their symmetric implementation. The following types of traffic are handled in the system: sensor data streams (e.g., radar object or camera picture data); diagnostic, service discovery, command and control data; and data to establish a common time base. In this analysis, radar object data and camera image data are targeted. Radar object data as well as camera image data are fragmented before being transmitted because their size exceeds one packet. All fragments except for the last one thus have the longest possible Ethernet frame size (full fragments). Each fragment is then sent in form of an Ethernet frame.

Sensor data is both time- and safety-critical, and corresponding communication requirements need to be specified as done in the subsequent section. In this study, all sensor image and object data are assumed to be of the same type, their sizes are the same, and they are assumed to be critical messages. The two other traffic types handled by the system are considered as non-critical.

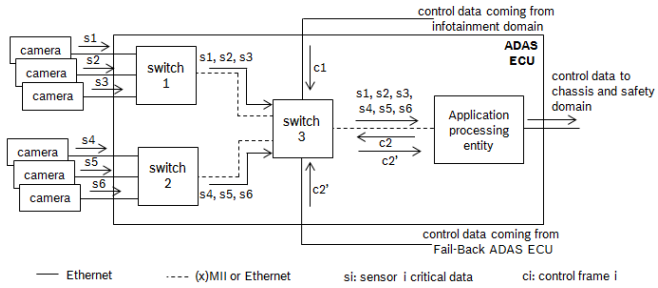



Fig. 2: System Communication Architecture

### B. Requirements on Transmission and Forwarding Delays

Image and object data are generated periodically. The accuracy of the period differs individually between sensors. In camera sensors, the image generation period is constant. Therefore, the availability period for transmission of image data frame fragments to a switch is constant. However, for radar sensors, the time between object data generations is not constant because of the varying number of detected objects. As a simplifying assumption in the presented study, we consider availability periods to be constant in the following. The delay deadline for each data transmission from a sensor is equal to its availability period to avoid an incoming frame when the previous one is still waiting. This period is actually the period at which new critical message fragments arrive in the sensor's output queue. On a given sensor $i$, it is denoted by $T_{Txi}$. Sensor critical messages are assumed to the same availability period here. A new sensor $i$ critical message is supposed to be available each $T_{Txi}$. All critical message fragments from a given sensor $i$ should thus be sent within $T_{Txi}$.

As illustrated in Figure 3, there are also requirements on forwarding deadlines in switches. Figure 3 depicts a given sensor $i$ critical message $n$ (in red) composed of three fragments. When the first one arrives at the directly linked switch 1, it has $T_{Txi}$ time left to forward all fragments of the critical message

$n$ to switch 3 (c.f. Figures 2 and 3). It is supposed here that sensor $i$ transmits all critical message $n$ fragments within $T_{Txi}$. When the first fragment enters the output queue of switch 3, $(T_{Txi} - dT)$ is the remaining deadline for switch 3 to forward critical message $n$ fragments to the destination processing entity. $dT$ is the time interval between critical message $n$'s first fragment reception at switch 1's input port and the start of its forwarding by that switch. This requirement is stated under the assumption that switch 1 has forwarded the fragments of critical message $n$ within the right deadline. The time $dT$ can be caused by the forwarding of non-critical frames and/or critical frames from other sensors. Non-critical messages are not considered in this study as shown in Figure 3.

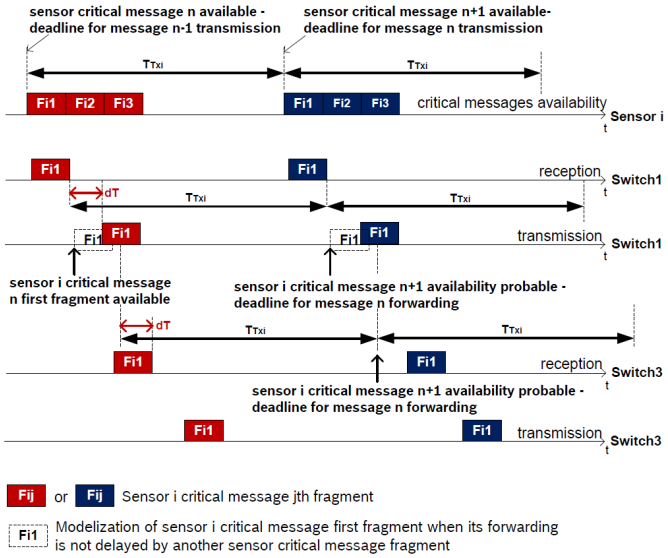



Fig. 3: Transmission and forwarding delays requirements

## III. Schedulability Analysis

The above-mentioned real-time requirements raise a general design problem which can be expressed as follow:

*What is the minimum required bandwidth or the maximum required bandwidth utilization factor at sensors and at switches output ports to guarantee that all critical messages are transmitted and forwarded within their deadlines?*

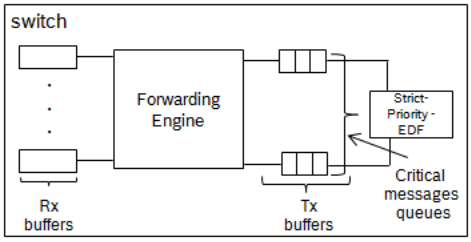



Fig. 4: Generic overview of a switch

Figure 4 shows the generic architecture of a switch. We consider three main ways to schedule Ethernet-based flows in the switches: strict priority based scheduling, reserved traffic based scheduling, and time-triggered scheduling. Table I lists some mechanisms and technologies applying the considered scheduling concept. Each mechanism with EDF in parenthesis means that it can be combined with earliest deadline first scheduling.

TABLE I: Traffic handling mechanisms

| Strict priority (EDF) | Reserved traffic | Time-triggered |
|---|---|---|
| | CBSA | Time-Aware Shaper |
| | Rate-Constrained (EDF) | Time-Triggered Ethernet |
| | AFDX (EDF) | |

The following set of parameters is defined for the subsequent schedulability analysis:

- $size_i$: critical message size on sensor $i$ (including fragment headers and trailers)
- $p_k$: number of sensors linked to switch $k$
- $resBW_i$: bandwidth reserved for critical messages at sensor $i$
- $C_{si}$: sensor $i$ critical message transmission delay without interruption
- $C_{swij}$: sensor $j$ critical message forwarding delay on switch $i$
- $portTxRate$: data transmission rate on each sensor output port
- $swportTxRate_i$: data transmission rate at the output port of switch $i$
- $U(sw_i)$: bandwidth utilization factor at switch $i$

Among the listed scheduling mechanisms we start the schedulability analysis in this paper with strict priority combined with EDF. Thus, sensors send critical messages via strict priority and switches forward them based on EDF.

Some assumptions are taken into consideration to perform a first simple analysis in this paper. However, the study will be extended in the future by removing them step-by-step. It is important to note that they present some deviations from the real world. We examine them one by one in the following, to see how (un)realistic they may be and how difficult the analysis may become upon their removal.

### *1) All sensors are of the same type*

In reality, this can be safely assumed for the considered architecture of a future driver assistance domain; this is, however, obviously not true for the whole automotive domain with diverse sensors. The fact that all sensors are connected to a switch $k$ ($k = 1, 2$), both being of the same type is very realistic. Switch 3 can also handle messages coming from sensors of the same type. Nevertheless that will lead to a change of the overall architecture of the driver assistance domain because a switch forwarding messages coming from different types of sensors to the application processing entity is needed.

*2) Critical messages from different sensors have the same size*

This assumption is comparable to the first one, being a direct consequence because transmitting messages of same type is one of the characteristics of a set of identical sensors. In addition to that, it is reasonable to assume that there will be a predefined type of critical message, which is identical for all sensors in our restricted architecture.

*3) The availability period is the same for all sensors*

This assumption is a direct consequence of the first one but not a consequence of the second one because different sensors could transmit critical messages of same size but at different time intervals.

Removing the three first assumptions complicates the analysis because of the relationship between different critical messages sizes and different availability periods. The fact that these relationships are random makes the study more involved. Proofs may still be possible in the case of unrelated (and thus memoryless) delays; if time-dependencies have to be considered, it will have to be evaluated if a numerical proof, simulation, or a proof with worst-case delay assumptions is applicable.

*4) The influence of non-critical frames is not considered*

Non-critical frames are also handled in our considered in-vehicle network. When this assumption is omitted, the influence of interfering non-critical traffic on critical messages transmission delays should be taken into account. For that, worst case interference delays have to be known to perform system schedulability evaluation. However, as these frames have a lower priority, the maximum delay will be directly computable.

*5) All port transmission rates are the same*

This assumption is very realistic and does not restrict the analysis, as the type of communication network will be the same. By removing it, different port transmission rates could be considered, e.g. 1 Gbps and 100 Mbps. This would only make sense if the sensor types and/or tasks would differ a lot despite our symmetric architecture (c.f. assumption 1). Its omission is not expected to considerably affect the difficulty of the analysis.

Following the aforementioned assumptions, it appears that for all switches $k$ $(k = 1, 2)$ (c.f. Figure 2), all critical messages have the same length and the same availability period. Therefore, the bandwidth that needs to be reserved on each sensor for critical messages is also identical. It follows that

$$C_{s1} = C_{s2} = \ldots = C_{sp_k} = C_0$$

$$T_{Tx1} = T_{Tx2} = \ldots = T_{Txp_k} = T_{Tx}$$

$$resBW_1 = \ldots = resBWp_k = resBW$$

Moreover, bandwidth utilization can then be expressed as follows:

$$U(sw_k) = \sum_{i=1}^{p_k} \frac{C_{si}}{T_{Txi}} = \frac{p_k \cdot C_0}{T_{Tx}}$$

Under these assumptions, timely scheduling of critical messages will be guaranteed for the switches that are directly connected to sensors in our topology (1 and 2 in our example), if the following two requirements are met:

1) For any given sensor $i$ (c.f. Figure 2), the port transmission rate should be at least equal to the bandwidth needed to be reserved for critical messages:

$$resBWi = \frac{size_i}{T_{Txi}}, portTxRate \geq resBW_i$$

2) For each switch $k$ $(k = 1, 2)$, the utilization of the switch's output port must not exceed 100% $U(sw_k) \leq 1$.

*Proof:* We know that $U(sw_k) \leq 1 \leftrightarrow T_{Tx} \geq p_k \cdot C_0 \leftrightarrow min(T_{Tx}) = p_k \cdot C_0$ . So, we show in the following that if $T_{Tx} \geq p_k \cdot C_0$, all critical messages are forwarded within the deadlines.

In Figure 5, three sensors are directly linked to a switch and send critical messages (each of them consisting of two fragments). In this example, the maximum difference between critical messages transmission start times from different sensors is so small that it is considered that transmissions are simultaneous. This means that the first fragments of critical messages originating from different sensors arrive simultaneously at the switch's input ports. Its forwarding engine transports them to the output port, but not simultaneously because input ports are sequentially monitored via a polling mechanism. Consequently, the arrival order of frames at the input ports is not necessarily the same at the output port. When sensors send their critical frames simultaneously as sketched in Figure 5, for a given sensor $i$ ($i = 1, 2$ or 3), the forwarding of critical message $n$ by the switch can be delayed in the worst case by fragments of all critical messages $n$ from other sensors. Forwarding of sensor $i$ critical message $n$ cannot be delayed by any fragment of critical message $n + 1$ from other sensors. The reason is that the last fragment of sensor $i$'s critical message $n$ arrives at the output port before any fragment of critical message $n + 1$ from other sensors. For this fragment, the deadline is earlier than that for any fragment of critical message(s) $n + 1$ from other sensors (c.f. Figure 5). Thus, following the EDF scheduling policy, it will be forwarded before them.

The maximum forwarding delay that can be experienced by critical messages at the switch output port from a given sensor $i$ is in this case the delay due to other sensors' critical messages ($(p_k - 1) \cdot C_0$) plus sensor $i$'s critical message forwarding delay without interruption ($C_0$). Therefore we have $max(C_{swij}) = p_k \cdot C_0$. In this case, critical messages are thus forwarded on time when $T_{Tx} \geq p_k \cdot C_0$. Figure 5 depicts a case where $T_{Tx} = p_k \cdot C_0$, showing that the forwarding of the last fragment from sensor 3's critical message $n$ is completed right on time.

When sensors do not transmit critical messages simultaneously (c.f. Figure 6), forwarding of a given sensor $i$'s critical message $n$ cannot be delayed by any fragment of critical message $n + 1$ from other sensors. The reason here is the same as the one stated before for the simultaneous case. As it can be seen in Figure 6, for a given sensor $i$ ( $i = 1, 2$ or 3), the maximum forwarding delay of critical message $n$ by the switch will be less than all fragments of critical messages $n$

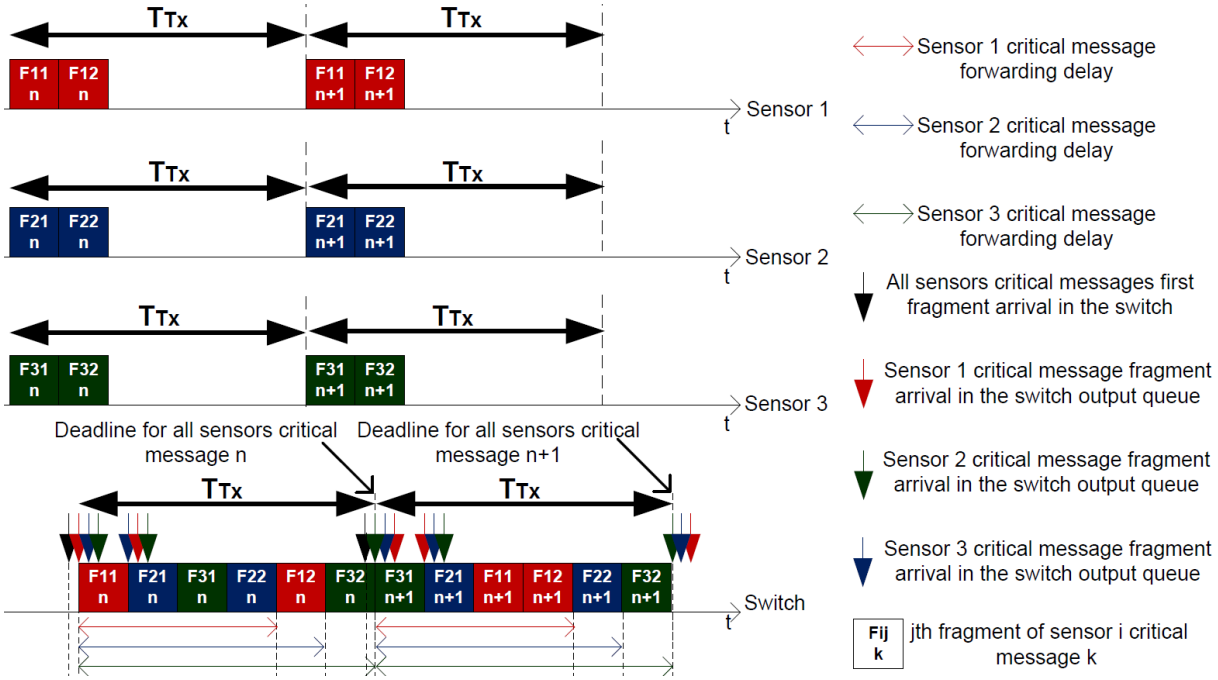



Fig. 5: Schedulability Analysis (1)

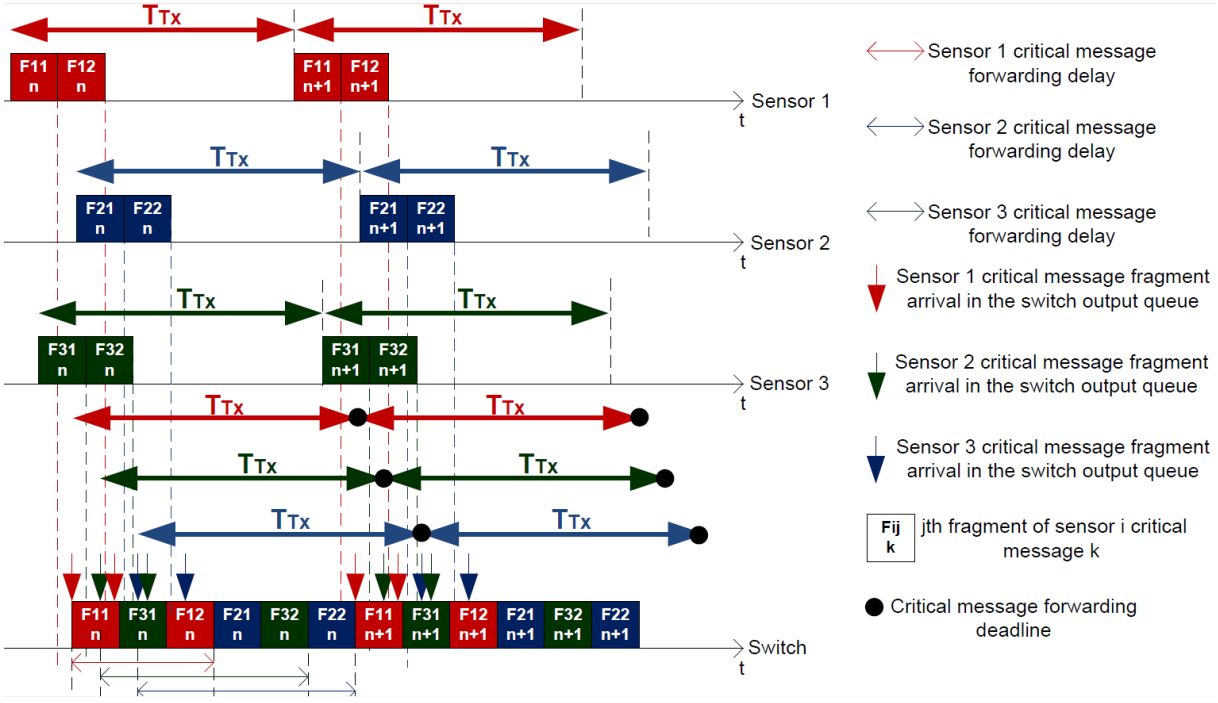



Fig. 6: Schedulability Analysis (2)

from other sensors. In this case: $max(C_{swij}) < p_k \cdot C_0$. Thus, the maximum critical message forwarding delay by a switch can be obtained when sensors transmit data simultaneously. This maximum delay is $max(C_{swij}) = p_k \cdot C_0$. Therefore, the condition to achieve guaranteed schedulability on a switch $k$ is $T_{Tx} \geq p_k \cdot C_0 \leftrightarrow U(sw_k) \leq 1$. ■

It remains to show that critical messages will not only pass switches 1 and 2 in a timely fashion, but will also pass the subsequent switch 3 correctly and will finally arrive within their deadline at the destination processing entity.

We thus analyze switch 3 (c.f. Figure 2) in the following. Here, bandwidth utilization is:

$$U(sw_3) = \sum_{i=1}^{p_1+p_2} \frac{C_{si}}{T_{Txi}} = \frac{(p_1+p_2).C_0}{T_{Tx}}$$

Now it will be demonstrated that schedulability is achieved on switch 3 if $U(sw_3) \leq 1$.

*Proof:* It holds that $U(sw_3) \leq 1 \leftrightarrow T_{Tx} \geq (p_1+p_2) \cdot C_0 \leftrightarrow \min(T_{Tx}) = (p_1+p_2) \cdot C_0$. We show in the following that if $T_{Tx} \geq (p_1+p_2) \cdot C_0$, all critical messages are forwarded within the deadlines.

Now we consider a network in which switch 1 is linked to three sensors and switch 2 to two sensors (c.f. Figure 7). All sensors send critical data simultaneously in this figure, as described previously in Figure 5. Critical data is transmitted by the sensors simultaneously in this case. This impacts switches 1 and 2 (c.f. Figure 7). We assume that switches 1 and switch 2 have the same characteristics, as expected in our architecture. The first fragments of the critical messages sent by two sensors thus arrive in the switch simultaneously. Both transmissions will start at the same time. The $n$th critical messages (from any source sensor) will always be forwarded by a given switch $k(k = 1, 2)$ before the subsequent critical messages $n+1$ (from any sensor). Consequently, in switch 3, the $n$th critical messages arrive at the output port before any fragment of any critical message $n+1$. Moreover, any critical message $n$ deadline is earlier than any critical message $n+1$ deadline. Therefore, at switch 3's output port, the forwarding of any $n$th critical message cannot be delayed by a fragment belonging to an $n+1$th critical message. For a given sensor $i(i = 1, 2, \ldots, p_1+p_2)$, an $n$th critical message is at most delayed by all fragments of the $n$th critical messages from the remaining $p_1+p_2-1$ sensors. For this reason, the maximum forwarding delay that can be experienced by the $n$th critical message at switch 3 is $\max(C_{swij}) = (p_1+p_2) \cdot C_0$. The example setup in Figure 7 shows this worst case in which $T_{Tx} = (p_1+p_2) \cdot C_0$, and the forwarding of the last fragment of the $n$th critical message transmitted by sensor 1 is completed just at the deadline. The example shown in Figure 7 is thus a worst case in which schedulability is achieved in switch 3 when $T_{Tx} = (p_1+p_2) \cdot C_0$ (or, equally, $U(sw_3) = 1$).

In the case when sensors do not transmit critical messages simultaneously (c.f. Figure 8), the $n$th critical messages arrive at switch 3's output port before critical messages $n+1$. Their deadlines are earlier than the ones of all critical messages $n+1$. Thus, for a given sensor $i$, the forwarding of any $n$th critical message can be delayed only by all other $n$th critical messages from the remaining sensors. The maximum forwarding delay can be expressed as follows:

$$\max(C_{sw3j}) < (p_1+p_2) \cdot C_0$$

It has been shown above that the worst case delay occurs when sensors transmit data simultaneously, and when switches 1 and 2 start forwarding them simultaneously as well. This maximum delay is $\max(C_{swij}) = (p_1+p_2) \cdot C_0$. Therefore, the condition to achieve schedulability on switch 3 is $T_{Tx} \geq (p_1+p_2) \cdot C_0 \leftrightarrow U(sw_3) \leq 1$. ■

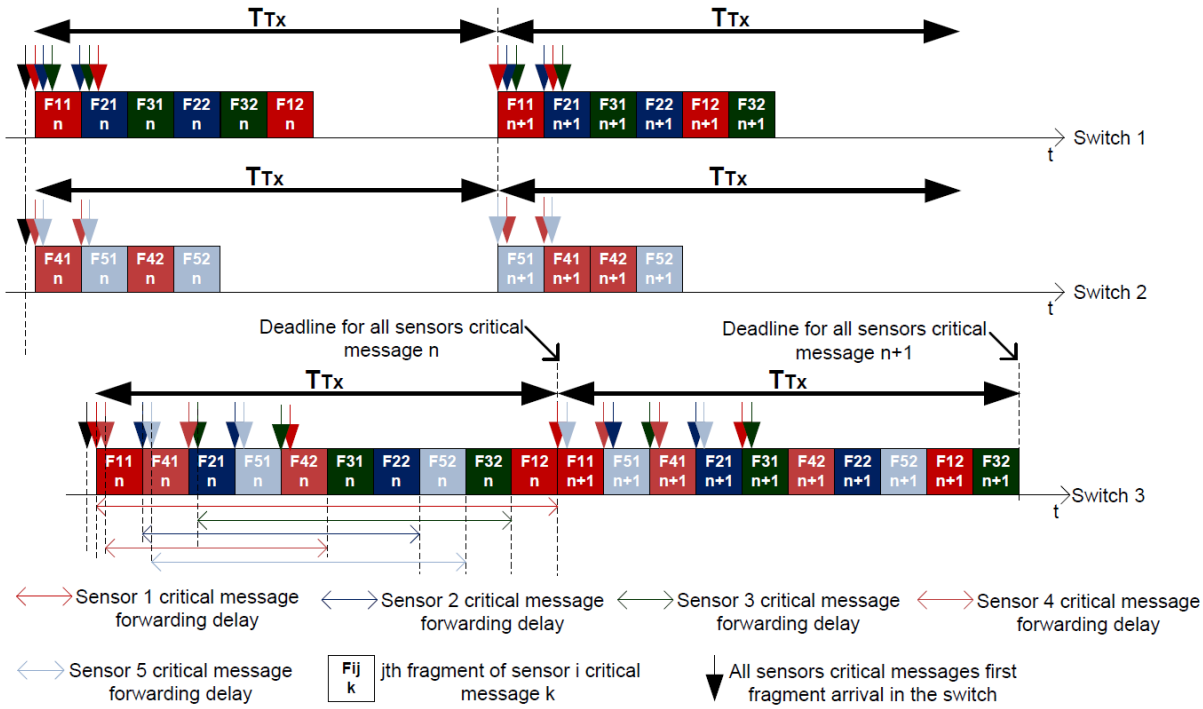
Putting the pieces together, we are now ready to analyze the overall behavior in the whole network (c.f. Figure 2). We already know that $U(sw_3) = U(sw_1) + U(sw_2)$. Therefore, $U(sw_3) \leq 1 \rightarrow U(sw_1) \leq 1$ and $U(sw_2) \leq 1$. Thus, $U(sw_3) \leq 1$ is a *sufficient* condition for guaranteeing schedulability in the whole network. It remains to be shown that when schedulability is achieved, we always have $U(sw_3) \leq 1$ (the condition is also *necessary*). Proving this is equivalent to proving that when $U(sw_3) > 1$, schedulability cannot be guaranteed in the network.

*Proof:* To show this, a contradicting example is sufficient. Lets consider the setup shown in Figure 7 as an example, in which $T_{Tx} = (p_1+p_2) \cdot C_0$, and thus $U(sw_3) = 1$. Forwarding of the last fragment of the $n$th critical message from sensor 1 is completed just when the first fragment of the next critical message $n+1$ arrives from the sensor in switch 3 output queue. This situation happens when all sensors start transmitting critical messages simultaneously, and when switch 1 and switch 2 start forwarding them simultaneously as well. In this scenario, when $U(sw_3) > 1$ (being equal to $T_{Tx} < (p_1+p_2) \cdot C_0$), the first fragment of sensor 1's critical message $n+1$ arrives in switch 1's output queue when the forwarding of the last fragment of the previous critical message is still being performed. Schedulability is therefore not achieved. ■

In conclusion, a necessary and sufficient condition to guarantee schedulability in the whole network is $U(sw_3) \leq 1$. What about networks consisting of $m$ switches linked to different sensors, and linked to a switch which is connected to the processing entity? We can apply the proofs given above as base case of a proof by induction, while the inductive step would be a generalization of the second proof. Continuing the arguments and proofs in this way would thus prove that the respective condition becomes $\sum_{i=1}^{m} U(sw_i) \leq 1$.

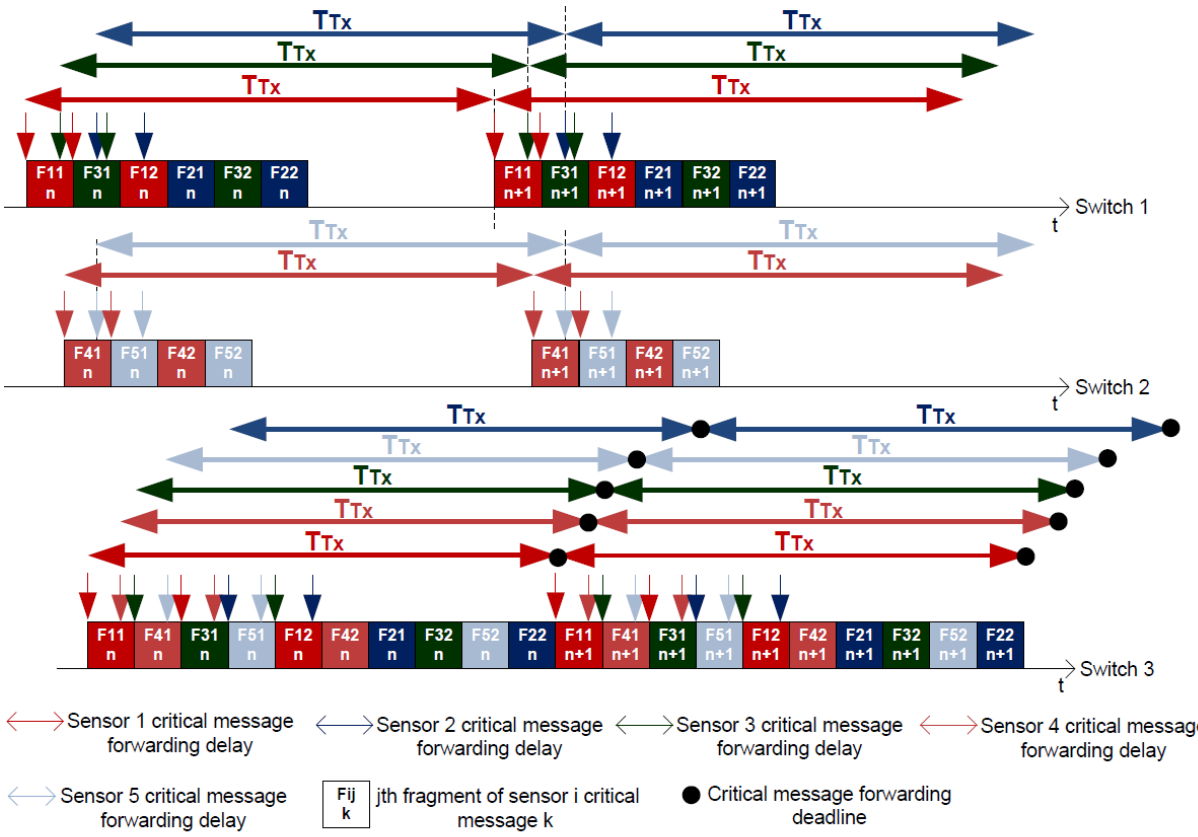
## IV. Conclusion

The paper highlighted real-time communication challenges in future Ethernet-based in-vehicle networks. To meet these challenges, a formal schedulability analysis methodology has been presented to ensure timely and thus safe operation. Future work will include similar analysis for other traffic-handling modes. In addition to that, we will extend the work by loosening the assumptions, allowing also critical messages with different sizes and different availability periods. Moreover, it will be interesting to check the influence of non-critical frames as well as control frames. A detailed network simulation will also be implemented to validate our theoretical results and to evaluate the accuracy of the results of this work.




**Remark:** all other arrows in color indicate corresponding critical message fragment arrival in the switch output queue

Fig. 7: Schedulability Analysis (3)




**Remark:** all other arrows in color indicate corresponding critical message fragment arrival in the switch output queue

Fig. 8: Schedulability Analysis (4)

## REFERENCES


[1] K. Matheus and T. Königseder, *Automotive Ethernet*. Cambridge University Press, 2014.

[2] H. Hoang and M. Jonsson, "Switched real-time Ethernet in industrial applications — deadline partitioning," in *The 9th Asia-Pacific Conference on Communications (APCC 2003)*, vol. 1, Sept 2003, pp. 76–81.

[3] H. Hoang, M. Jonsson, U. Hagström, and A. Kallerdahl, "Switched real-time Ethernet and earliest deadline first scheduling — protocols and traffic handling," in *Proc. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'2002) at the Int. Parallel and Distributed Processing Symposium (IPDPS'02)*. Fort Lauderdale, USA: IEEE Computer Society, 2002, pp. 308–.

[4] H. Hoang, M. Jonsson, A. Larsson, R. Olsson, and C. Bergenhem, "Deadline first scheduling in switched real-time Ethernet — deadline partitioning issues and software implementation experiments," in *Proc. 14th Euromicro Conf. on Real-Time Systems (EUROMICRO 2002)*. Vienna, Austria: IEEE Computer Society, June 2002, pp. 68–71.

[5] R. Marau, L. Almeida, P. Pedreiras, K. Lakshmanan, and R. Rajkumar, "Utilization-based schedulability analysis for switched Ethernet aiming dynamic QoS management," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2010)*, Bilbao, Spain, Sept 2010, pp. 1–10.

[6] M. Tawk, G. Zhu, J. Li, X. Liu, Y. Savaria, and F. Hu, "Optimal scheduling and delay analysis for AFDX end-systems," in *SAE Aero-Tech Conference*, Toulouse, France, Oct 2011.

[7] J. Chen, Z. Wang, and Y. Sun, "Real-time capability analysis for switch industrial Ethernet traffic priority-based," in *Control Applications, 2002. Proceedings of the 2002 International Conference on*, vol. 1, 2002, pp. 525–529.

[8] Q. Zhang and W. Zhang, "Priority scheduling in switched industrial Ethernet," in *Proc. of the American Control Conference*, June 2005, pp. 3366–3370 vol. 5.

[9] X. Fan, M. Jonsson, and J. Jonsson, "Guaranteed real-time communication in packet-switched networks with FCFS queuing," *Computer Networks*, vol. 53, p. 400?417, 2009.

[10] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng, "Schedulability analysis of Ethernet AVB switches," in *Proc. Int. Conf. Real-Time Computing Systems and Applications (RTCSA)*, 2014, pp. 1–10.