Fail Silent Road Side Unit for Vehicular Communications

Joaquim Ferreira, Arnaldo Oliveira, João Almeida, Cristóvão Cruz

Fail Silent Road Side Unit for Vehicular Communications
(Invited Paper)

Joaquim Ferreira[1], Arnaldo Oliveira[2], João Almeida[2], and Cristóvão Cruz[2]

[1] ESTGA-Instituto de Telecomunicações, University of Aveiro, Portugal
[2] DETI-Instituto de Telecomunicações, University of Aveiro, Portugal

Abstract. After almost a decade of R&D, vehicular communications for cooperative systems and its enabling technologies, mostly relying on IEEE 802.11p and IEEE 1609.1-4 family of standards, are in their trial phase, as some major field operational tests are being carried out. Nevertheless, there are still some open problems concerning the timeliness, dependability and security of IEEE 802.11p based communications. This paper presents an architecture to implement fail-silent road side units (RSU) required to deploy infrastructure to vehicle (I2V) safety communications. Although generic, the proposed architecture is described within the scope of the Vehicular Flexible Time-Triggered protocol (V-FTT) and takes advantage of a flexible, FPGA-based softcore implementation of an IEEE 802.11-A6 / ETSI ITS G5 controller.

1 Introduction

Wireless vehicular networks for cooperative Intelligent Transport Systems (ITS) have raised widespread interest in the last few years, due to their potential applications and services. Cooperative applications with data sensing, acquisition, processing and communication provide an unprecedented potential to improve vehicle and road safety, passenger's comfort and efficiency of traffic management and road monitoring. Safety, efficiency and comfort ITS applications exhibit tight latency and throughput requirements, for example safety critical services require guaranteed maximum latencies lower than 100 ms while most infotainment applications require QoS support and data rates higher than 1 Mbit/s. The mobile units of a vehicular network are the equivalent to nodes in a traditional wireless network, and can act as the source, destination or router of information. Communication between mobile nodes can be point-to-point, point-to-multipoint or broadcast, depending on the requirements of each application. Besides the adhoc implementation of a network consisting of neighbouring vehicles joining up and establishing Vehicle-to-Vehicle (V2V) communication, there is also the possibility of a more traditional wireless network setup, with base stations along the roads in Vehicle-to-Infrastructure (V2I) communication that work as access points and manage the flow of information, as well as portals to external WANs.

Devices operating inside vehicles are called On Board Units (OBUs), while devices operating on the side of the road are Road Side Units (RSUs), and have different requirements and modes of operation.

The IEEE 1609 family of standards for Wireless Access in Vehicular Environments (WAVE) defines an architecture and a standardized set of services and interfaces that collectively enable secure V2X wireless communications. Additionally, the IEEE 1609 standards rely on IEEE 802.11-2012 Amendment 6 [5], also known as 802.11p, and the equivalent European standard ETSI ITS G5 [4]. The physical layer is almost identical to IEEE 802.11a, using also orthogonal frequency-division multiplexing (OFDM) with BPSK, QPSK, 16QAM and 64QAM modulations, but with double timing parameters to achieve less intersymbol interference due to the multi-path propagation and the Doppler shift effect. With double timing parameters, the channel bandwidth is 10MHz instead of 20MHz, and the data rate is half, i.e., 3...27 Mbit/s instead of 6...54 Mbit/s. The maximum range is 1000m, with line of sight (close to 300m in typical conditions) for vehicle speed below 200 Km/h.

The medium access control (MAC) layer adopts a carrier sense multiple access with collision avoidance (CSMA/CA), as IEEE 802.11a, but with a new additional, non-IP, communication protocol, either Fast Network and Transport Protocol (FNTP) or Wave Short Message Protocol (WSMP). Due to the tight timing constrains, non-IP protocols do not perform channel scanning, authentication and association. Since IEEE 802.11p medium access control is based on CSMA, collisions may occur indefinitely due to the non-determinism of the back-off mechanism. So, native IEEE 802.11p alone does not support real-time communications. The probability of collisions occurring may be reduced if the load of the network is kept low, which is difficult to guarantee in vehicular communications, or if some MAC protocol restricts and controls the medium access to provide a deterministic behavior.

Strict real-time behaviour and safety guarantees are typically difficult to attain in ad-hoc networks, but they are even harder to attain in high speed mobility scenarios, where the response time of distributed consensus algorithms, e.g. for cluster formation and leader election, may not be compatible with the dynamics of the system. There are basically two main design choices to implement a MAC protocol for wireless vehicular communications. It either could rely on the road side infrastructure or it could be based on ad-hoc networks, without road-side units support. Hybrid approaches are also possible. However, the presence of the infrastructure, e.g. road-side units and the backbone cabled network, adds a degree of determinism that will very useful to enforce real-time and safety at the wireless end of the network.

Recently, a proposal for deterministic medium access control (MAC) protocol was presented [10]. This protocols, called vehicular flexible time-triggered (V-FTT), adopts a master multi-slave time division multiple access (TDMA), in which the road-side units act as masters and schedule the transmissions of the on-board units. As depicted in 1, the protocol is divided into periodic elementary cycles (ECs) and each EC starts with a infrastructure window (I2V), containing

trigger messages, with the schedule of the OBUs allowed to transmit safety message, and warning messages.

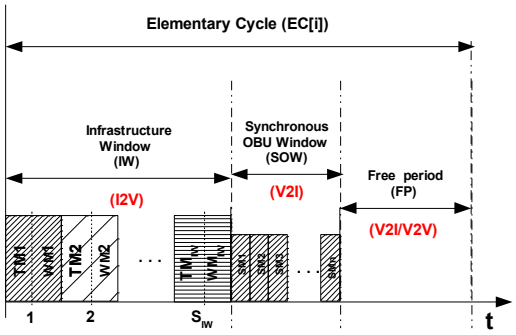

Fig. 1. Elementary Cycle of Vehicular Flexible Time-Triggered protocol [10].

The infrastructure window is followed by the synchronous OBU window, where OBUs have the opportunity to transmit information to RSUs (V2I). Each OBU will have a fixed size slot to transmit vehicle information (speed, acceleration, heading, etc.) and/or a safety event. The Synchronous OBU Window duration is variable. The elementary cycle ends with an optional free period window, a period where non V-FTT enabled OBUs are able to transmit safety messages and RSUs and OBUs are able to transmit non-safety messages.

The information broadcast by the RSU must be reliable, so that internal RSU faults are not propagated to other nodes. RSUs must validate OBU events and edit the information they broadcasts to the vehicles. Consider, for example, the case where a faulty OBU tries to send an Emergency Electronic Brake Light message. If no editing is made, several vehicles would receive a false alarm which could lead to dangerous situations or even accidents. This edition operation must obviously be performed in bounded time so that the results can be transmitted to the OBUs in real-time.

This paper proposes an architecture to enforce fail silent behaviour in vehicular networks based on the V-FTT protocol. More specifically, it defines a fail silence enforcement entity, located at RSU, that is capable of enforcing an agreement, both in the time and in the value domains, between two replicas of an ETSI ITS G5. In case of disagreement, no message is actually sent to the network. Nodes of a distributed system exhibiting fail silent behaviour ease the task of designing fault tolerance mechanisms since local faults are not propagated to other nodes.

The rest of the paper is organized as follows. Section 2 describes the problem and provides some background information on the advantages of enforcing fail silence failure mode, the V-FTT protocol and on the target system architecture

and design assumptions. Section 3 describes some relevant related work on the enforcement of fail silence behaviour, while section 4 discusses some alternatives to design a fail silent road side unit for the V-FTT protocol. Finally, Section 5 summarizes the main conclusions of the paper and unveils some future work.

2 Problem statement and background

The proposed system architecture takes advantage of a flexible, FPGA-based softcore implementation of a IEEE 802.11 A6 / ETSI ITS G5 controller, the IT2S platform, which has been developed from scratch at Telecommunications Institute (Aveiro site), in the scope of two research projects: HEADWAY-Highway Environment ADvanced WArning sYstem, funded by Brisa, a motorway operator, and ICSI - Intelligent Cooperative Sensing for Improved traffic efficiency, an FP7 project. Most of the implementation efforts have been concentrated on the non-IP of the WAVE standards (WMSP) and supporting 802.11p MAC and PHY layers. The approach followed to implement the defined subset of the WAVE standard explores a mix of hardware (analogue components and multiple processors, both general purpose and specialized) and software to achieve the required performance and flexibility levels. 2 shows the global structure of our current implementation and a picture of the IT2S board.

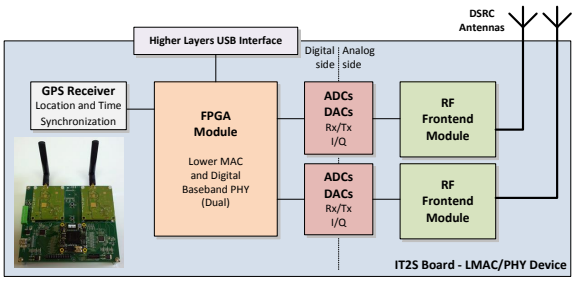

Fig. 2. Architecture of the IT2S board.

The IT2S transceiver board can be integrated in a WAVE communication box together with a single board computer. This box can either be used as OBU or as RSU. Each communication box requires several interfaces (with both the vehicle devices and the user/driver), namely: power supply, antenna connectors, OBD-II interface, USB. Driver interface may rely on a specific display with touch capabilities or through an Embedded or Portable Navigation Device (PND), using the USB interface.

An impairment to the dependability of vehicular networks for safety applications, as the ones based on the V-FTT protocol, is the fail uncontrolled nature

of the nodes of the system, both RSUs and OBUs. Several schemes have been reported over the years to control failure modes of distributed systems nodes. Enforcing fail silence failure mode is one of such schemes.

2.1 Fail silence failure mode

A faulty node of a distributed system that sends unsolicited messages at arbitrary points in time (babbling idiot failure mode [6]) without respecting the media access rules can disable nodes with legitimate messages to access the network. However, this failure mode can only occur if a node fails in an uncontrolled way. Network topologies that support the operation of fail uncontrolled nodes are costly [7]. Thus a node should only exhibit simple failure modes and ideally it should have just a single failure mode, the fail silent failure mode [11], i.e., it produces correct results or no results at all. In this matter, a node can be fail-silent in the time domain, i.e., transmissions occur at the right instants, only, or in the value domain, i.e., messages contain correct values, only. With fail silence behaviour, an error inside a node cannot affect other nodes and thus each node becomes a different fault confinement region [11]. Furthermore, if k failures of a functional unit in a system must be tolerated, then k+1 replicas of that unit are needed as long as they are fail silent. If the replicas are fail uncontrolled, then $3k + 1$ will be required. Thus, the use of fail silent nodes also reduces the complexity of designing fault-tolerant systems. Usually, in wired networks, fail silence is enforced by bus guardians, which are autonomous devices with respect to the node network controller and host processor, which act as failure mode converters, i.e., the failure modes of the component are, at the interface to other components, replaced by the failure modes of the guardian. In wireless communications, one can think in medium guardians as devices that protect non-faulty wireless network nodes from erroneous ones.

In order to be fail-independent with respect to the interface it monitors the medium guardian must belong to a separate fault confinement region. A guardian would be of no use if it failed whenever the node that it is guarding also failed. Some potential sources of common mode failures are: clocks, CPU/hardware, power supply, protocol implementation, operating system, etc. Designing a medium guardian with independent hardware, with no common components and design diversity can help to avoid common failure modes. Despite the possible design compromises made between independence, fault coverage and simplicity/cost in any medium guardian architecture it is mandatory for the guardian to have some a priori knowledge of the timing behaviour of the node it is policing. In time-triggered (TDMA) networks this implies that each medium guardian needs to have its own copy of the schedule and an independent knowledge of the time.

In master-slave networks, as V-FTT, fail silence in the slave nodes (OBUs) could be enforced using either bus guardians or internal replication and temporized agreement. For the case of the master nodes (RSUs) it is necessary to enforce fail-silence both in the time and value domains. This is mandatory for the

master nodes, to guarantee the correctness of the EC-schedules that are broadcast to the network. A medium guardian policing functionality cannot be used in RSU nodes because of the causal relations between the RSUs computed schedules and the medium guardian operation. Fail silence enforcement in V-FTT roadside units require a replicated processing/voting scheme.

OBUs (slaves) also require fail-silence behaviour and although one could adopt the same mechanism used in master nodes, that would be expensive. Thus, slave nodes fail-silence enforcement both in time and value domain should only be adopted in special cases where the slave node information (value and timing) is absolutely essential, e.g., for police and emergency vehicles. In other cases, limiting OBUs ability to transmit uncontrollably will suffice. This corresponds to enforce fail-silence behaviour in the time domain only. From the OBU's perspective, a schedule is valid only within the scope of an elementary cycle, thus the medium guardian policing a node only needs to be aware of the node schedule in a EC by EC basis. In this way the medium guardian decodes every trigger message contents and blocks any unscheduled transmission from the node. The architecture of the medium guardian for OBU nodes is outside the scope of this paper.

The rationale to provide fault-tolerance mechanisms at the lower levels of the OSI stack, is based on the observation that moving them higher in the OSI stack, increases both the application design complexity and potentiates design defects. This is an important factor to take into account and whose relevance keeps increasing with the application size: the more complex it is, the higher is the number of potential software defects and more difficult becomes to identify them. This is why some other authors also claim [6][8] that the fault-tolerance mechanisms should be implemented as low as possible within the OSI stack to partially hide them to the designers. Those mechanisms are carefully designed and properly validated once and may be used thereafter with some guarantees. The drawback of providing fault-tolerance mechanisms at the lower levels of the OSI stack is the overhead introduced by these mechanisms in applications that do not require them.

2.2 System structure and assumptions

We make the following assumptions about the vehicular communication system and the components it contains:

- The system consists of a set of fail silent wireless nodes, both RSUs and OBUs, equipped with ETSI ITS G5 radios.
- The RSUs act as masters in the sense they schedule the transmissions of the OBUs (slaves) according to the V-FTT protocol, controlling the medium access.
- The RSUs are interconnected through a reliable wired network, using some resource reservation protocol, e.g. SRP, to perform cooperative scheduling of the OBUs.
- Non V-FTT compliant nodes are only able to transmit during the free period.

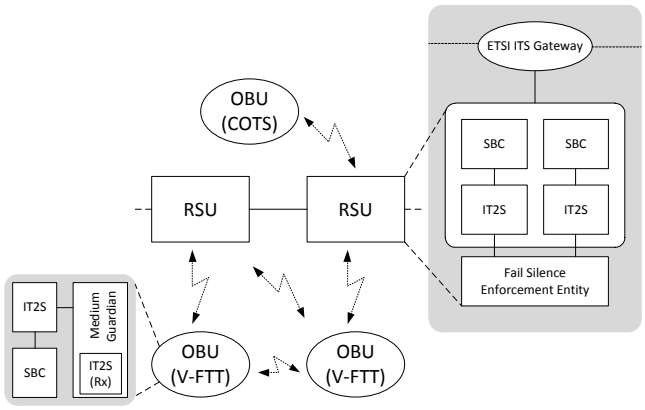

Fig. 3. V-FTT network based on road side infrastructure.

- Each node contains a set of behavioural based error detection mechanisms: the fail silence entity for the case of RSUs and a medium guardian for the case of OBUs.
- The hidden terminal and exposed terminal problems are only relevant during the V-FTT' free period, redundant RSU scheduling solves these problems for the case of synchronous OBU window.
- The V-FTT scheduling is dynamic, in an elementary cycle basis, thus the OBUs only become aware of it after decoding the trigger messages sent by the RSUs.

The system structure is depicted in 3. Each vehicular RSU is composed by and ETSI ITS gateway connected to two processing units (single board computers - SBC), two IT2S platforms connected to the SBC via USB interfaces and a fail silence enforcement entity. OBUs integrate a SBC, an IT2S platform and a medium guardian equipped with an ITS2 receptions chain to autonomously decode the trigger message.

The fault model assumed for the system is not a restrictive one: hardware faults, both transient and permanent, and software faults are possible within each node. We also assume fault independence between the processing elements (SBC and IT2S) and the medium guardian, for the case of OBUs, or the fail silence enforcement entity. We do not consider Byzantine faults, notably intrusions, occurring at the RSU level. Such kind of faults need to be handled at the ETSI ITS gateway level.

3 Related work

The alternatives to enforce fail silent behaviour may be generically divided in two main groups; the ones that result from adding redundancy to each node and ones that rely on behavioural error detection techniques [11].

Using replicated processing within a node with output comparison or voting calls for the use of mechanisms to keep the replicas perfectly synchronized and to avoid replicas to diverge due, e.g. to asynchronous events. Synchronization at processor instruction level is the most obvious way to achieve replica synchronism, driving identical processors with the same clock source and evaluating their outputs (either comparing or voting) at critical instants, e.g. every bus access. Special care must be taken with asynchronous events that must be delivered to the processors so that all perceive the same event at the same point of their instruction streams.

Over the years many systems were designed based in double-processor fail silent nodes such as Sequoia [2] and Stratus [13]. However, these systems have some drawbacks [3]. First of all the processors must exhibit the same deterministic behaviour every clock cycle and don't care states are not allowed so that they produce identical outputs. Secondly, the use of special purpose hardware as comparators or voters, reliable clock sources and asynchronous event handlers greatly increases the design complexity. Finally, due to their operation in lock step, a transient fault could affect both processors in the same way, making the node susceptible to common mode failures. An alternative approach to eliminate the hardware level complexity of the solutions referred above is to transfer the replica synchronism to a higher level using software protocols over a set of standard processors operating independently of each other in a node. Task synchronization approaches were used in SIFT [14] and Voltan [9].

Behavioural error detection mechanisms, either in software or in hardware, are another alternative for enforcing fail-silence behaviour. Mechanisms such as checksums, watchdog timers and processor monitoring, are usually implemented using COTS components. Error detection latency is the major bottleneck of these systems since the error detection mechanisms are only able to detect errors a relatively long time after they occur, possibly forcing other nodes to put in place some sort of error recovery policy.

Bus guardians, which are autonomous devices with respect to the node network controller and host processor, also implement behavioural error detection mechanisms that contribute to reduce possible residual fail silent violations resulting from the error detection latency by enforcing an adequate timing in the node transmissions.

To the best of authors' knowledge, there is no prior work of enforcing fail silence behaviour in wireless distributed embedded systems. Most of the work in the area is for wired industrial, or automotive systems [12][1]. Although the principles are similar, nodes of wireless vehicular networks pose some additional problems implementing fail silent behaviour, arising both from the open nature of such networks, in contrast with closed wired networks found in industrial and automotive systems, and from the physical impossibility of having a radio receiv-

ing the transmission of another one located a few centimetres away. Furthermore, the use of centralized bus guardians as in star topologies is not possible.

4 Fail-silent RSU design

As already stated, in order to implement a fail-silent RSU, all the possible device failure modes must be converted in fail-silent failure mode, enforced by simpler, thus less prone to failures, component. Figure 4 depicts two complete sets of SBCs and IT2S platforms, and their possible output verification points. Validation at each of these points implies the comparison of the values and timings of the outputs of each module. Each checkpoint location has its advantages and drawbacks, requiring a careful analysis to reach best compromise for the implementation of the fail silence mechanism. Fail silence enforcement should be performed by an external entity, ideally with no common mode failure mode, i.e., with separated power supply and clock source. However, this is costly and one may consider several degrees of coupling between the fail silent enforcement components and the node being policed.

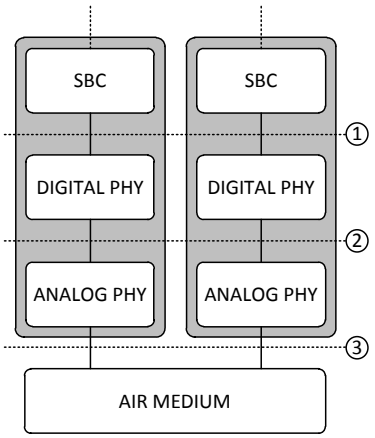

Fig. 4. Possible checkpoints inside the RSU

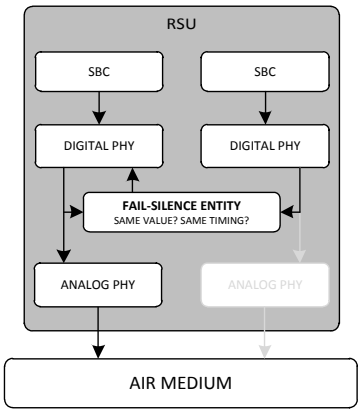

Fig. 5. Fail silent master scheme - basic proposal

Output comparison at the checkpoint 1 could be attained using a voting scheme between both SBCs on high level outputs, e.g., the V-FTT scheduling. At this point, however, no verification is possible as to the state of the lower levels, such as the MAC layer.

Output comparison at checkpoint 2, already encompasses the results of the complete digital baseband chains and, any discrepancy between the outputs of

each module can be detected at this stages. If, for instance, the SBCs attempts to transmit different messages, resulting, e.g., from different views of the network or from inconsistent scheduling computations, the outputs at this checkpoint will differ and that error can be detected. It is also possible to detect, at this stage, hardware faults occurring in one of the digital PHYs, e.g., stuck at or bit flipping.

Ideally, checkpoint 3 is the best place to verify the correctness of the outputs of both transmission chains, because we are already observing the radio signals that are being received by the other nodes. However, implementing a verification algorithm of high frequency analog signals is a very difficult task and, since both transmission chains could produce correct signals although slightly different, due to minor discrepancies on the digital to analog conversion and on the radio frequency amplifiers.

From the analysis of these three possible verification points, checkpoint 2 seems to be the most promising approach. The analysis of the data at the this level may, only by itself, ensure the correctness of the complete digital system, including both the SBC and the FPGA implementation. However, possible faults on the RF modules are not covered.

A basic implementation, presented in figure 5, would be to perform a runtime comparison of the output produced by both digital PHYs and signal an error that would abort the ongoing transmission whenever a mismatch is detected. This method, however, would not prevent the medium from being occupied during at least part of an incorrect frame transmission, a small tolerance in the moment when the results are produced must be allowed. This implementation would not result then in a true fail silent entity.

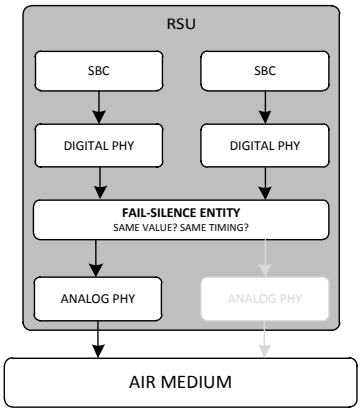

Fig. 6. Fail silent master scheme - fail silence at sample level

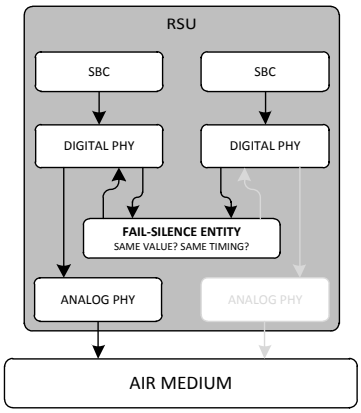

Fig. 7. Fail silent master scheme - true fail silence

A possible improvement on the previous scheme is presented in figure 6, where each of the samples produced by the digital PHY is only allowed to proceed to the analog PHY after successful validation. This solution, although providing protection against frames that are completely different or out of time, still does not provide true fail silence behaviour, because a single error occurring at the middle of a message invalidates the complete message. The medium could therefore be occupied with samples that although correct when analysed independently, were incorrect when in the context of a full frame. Beyond that, given that under V-FTT the RSU coordinates all the communications, a complete elementary cycle would be wasted. The implementation of a true fail-silent system requires knowledge of the system as a whole, namely, the fact that it implements a packet oriented communication system, and that an error at any part of a packet jeopardizes it completely.

The proposed solution for this problem, which can be observed in figure 7, is to modify the transmission chain to produce the samples that are to be sent over the air in advance, relative to the moment when they are supposed to be sent. If enough advance is provided, the samples of an entire frame can be compared and validated before that frame transmission has even started. In this scheme, the validated samples are then fed back to the digital PHY that will, using a very simple mechanism, apply them at the correct moment to the analog PHY for transmission. If any difference is observed at any part of the frame or if the measured delay between the instants when samples are provided is greater than a certain fixed tolerance, the fail silent entity will not allow any of two units to transmit. In this case, the medium will not be occupied inadequately, contrarily to the previous proposals.

A possible implementation of the fail-silence entity can be made using an FPGA external to the IT2S board, using general purpose IO pins to connect both platforms, along with adequate changes to the sample generation timing.

5 Conclusions and future work

The main contribution of this paper is a proposal of an architecture to implement fail-silent road side units (RSU) in the scope of infrastructure to vehicle safety communications. Although generic and applicable to other white box implementations of IEEE 802.11 A6 / ETSI ITS G5 controllers, the proposed architecture is described within the scope of the Vehicular Flexible Time-Triggered protocol and takes advantage of a flexible, FPGA-based softcore implementation of a IEEE 802.11 A6 / ETSI ITS G5 controller, the IT2S platform.

Future work in this line of research, includes the implementation of the fail silence enforcement component on a FPGA board and its validation using fault injection techniques. We also plan to address the internal replication of the RSU to increase its availability. For this purpose we are considering using an active replication protocol to minimize the inaccessibility time upon failure of the active replica. Concerning the V-FTT compliant OBUs, we plan to design

medium guardians that, with an independent knowledge of the TDMA schedule, will be able to protect the network against babbling idiot failures of the OBUs.

Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7) under grant agreement n. 3176711 and by BRISA, under research contract with Instituto de Telecomunicações - Aveiro.

References

1. Belschner, R. *et al.* FlexRay Requirements Specification, version 2.0.2. *FlexRay Consortium, http://www.flexray-group.com*, 2002.
2. P. Bernstein. Sequoia: A Fault-Tolerant Tightly Coupled Multiprocessor for Transaction Processing. *IEEE Computer*, 21(2):37–45, 1988.
3. Francisco V. Brasileiro, Paul Devadoss Ezhilchelvan, Santosh K. Shrivastava, Neil A. Speirs, and S. Tao. Implementing Fail-Silent Nodes for Distributed Systems. *IEEE Transactions on Computers*, 45(11):1226–1238, November 1996.
4. ETSI. Final draft ETSI ES 202 663 V1.1.0 (2009-11), ETSI Standard, Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band, November 2011.
5. IEEE. 802.11-2012 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2012.
6. H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Press, 1997.
7. D. Powell. *Delta-4 - A generic Architecture for Dependable Distributed Computing*. 1991.
8. J. Proenza and J. Miro-Julia. MajorCAN: A modification to the Controller Area Network to achieve Atomic Broadcast. *IEEE Int. Workshop on Group Communication and Computations. Taipei, Taiwan*, 2000.
9. S. Shrivastava, P. Ezhilchelvan, N. Speirs, S. Tao, and A. Tully. Principal Features of the Voltan Family of Reliable Node Architectures for Distributed Systems. *IEEE Transactions on Computers (Special Issue on Fault-Tolerant Computing)*, 41(5):542–549, 1992.
10. Meireles T., Fonseca J., and Ferreira J. Vehicular Flexible Time-Triggered Protocol (V-FTT). Technical Report 1, Instituto de Telecomunicações - Embedded Systems Group, March 2013.
11. C. Temple. Avoiding the babbling-idiot failure in a time-triggered communication system. *Fault Tolerant Computing Symposium*, pages 218–227, 1998.
12. TTTech. Time-Triggered Protocol TTP/C High-Level Specification Document (edition 1.0). *http://www.ttagroup.org*, 2002.
13. S. Webber and J. Beirne. The Stratus Architecture. *Digest of Papers FTCS-21*, pages 79–85, 1991.
14. Wensley, J. *et al.* SIFT: Design and Analysis of a Fault Tolerant Computer for Aircraft Control. *Proceedings of IEEE*, 66(10):1240–1255, 1978.