Considering Safety Requirements in Design Phase of Future E/E Architectures

Hadi Askaripoor, Morteza Hashemi Farzaneh, and Alois Knoll
Chair of Robotics, Artificial Intelligence and Real-time Systems
Technical University of Munich
Boltzmannstr. 3, 85748 Garching bei München
Email:{hadi.askari,morteza.hashemi}@tum.de, {knoll}@in.tum.de

***Abstract*—Without meeting safety requirements in the design of electric/electronic (E/E) architectures, achieving fully-automated vehicles is infeasible. However, considering architecture-related safety requirements (e.g. redundancy for fail-operational) in the design phase is a time-consuming task that requires domain-specific knowledge. This paper tackles this challenge by proposing a novel approach under development that takes topological safety aspects into account and transforms them into an optimization problem to generate safe topologies. We aim at accelerating the architecture design process while reducing unnecessary verification efforts as well as avoiding undesired functional safety violations.**

***Index Terms*—Autonomous Driving, E/E Architectures, Functional Safety, Optimization, Redundancy, Topology.**

I. Introduction

Development of self-driving cars brings new challenges for the E/E-architecture designers. Especially, the safety requirements, for instance, fail-operational guarantees must be considered at the beginning of the architecture design. Functional safety as a prominent aspect plays a pivotal role in the automotive domain. The most important international safety standard is ISO 26262 (derived from IEC 61508) which is applied to E/E systems. It consists of 10 parts including the automotive safety integrity level (ASIL) to measure the risk of a specific component in the system. In other words, this standard covers the functional safety in the incident of system failures; however, excludes the safety hazards that cause without system failure. In contrast, SOTIF/PAS21448 (Safety of The Intended Functionality) covers the safety hazards without system failure and is an appropriate complement for ISO 26262 as well as a guarantee for artificial intelligence decisions in autonomous vehicles [1], [2].

However, being an autonomous vehicle opens a significant number of challenges corresponding to vehicle safety at runtime and design-time. The topology of the architecture has significant impact on various aspects of the car including safety (e.g. redundancy for fail-operational) and cost. Moreover, the design of the vehicle E/E architecture in compliance with functional safety standards is an elaborate task for automotive designers. For example, finding of adequate routes and traffic schedules in architectures with fixed topology are well-known intractable NP-problems [3] with an exponential or even over-exponential computations times. It is obvious that the creation of a safe topology increases the complexity of the problem.

Our work in progress paper presents a framework under development to generate the architecture topology based on predefined constraints including e.g. route redundancy to facilitate the architecture design procedures of self-driving vehicles.

This paper is organized as follows: the next section looks at the related works relevant to vehicle topology, functional safety, and networking focused on the E/E architecture. Section III describes our approach and its formulation. Finally, section IV provides further steps and the conclusion of this work.

II. Related Work

Mody [4] presents an explanation regarding AD/ADAS topologies by analyzing two system topologies as two examples. Besides, the authors have compared these two topologies by considering several parameters including bandwidth, functional safety, number of ECU, cost, etc. Also, system partitioning with focus on the AD/ADAS domain corresponding to each topology was explained in this work. Zerfowski et al. [5] discuss upcoming E/E architectures within the automotive domain as well as related aspects to the E/E architecture components comprising cyber security, energy management, appropriate middleware, etc.

The network used within the automotive domain, specifically the Ethernet network, has been analyzed in terms of safety aspects by this paper [6]. Also, a bit flip comparison between CAN based and Ethernet-based networks has been done. Asim et al. [7] propose a safety approach to improve the safety of the architecture utilized in an autonomous car and assess the architectural design of the self-driving system at the development process.

A method is declared for safety-critical applications within the automotive domain which is appropriate for a centralized ECU by Yoneda et al. [8]. It is a NOC (Network-on-Chip) platform and is designed to mitigate link faults as well as handle delay faults. Adina et al. [9] has proposed an approach for safety verification of autonomous systems that the static verification technique, at design time, is combined with dynamic one, at run time, to transfer the results of static verification to run time environment.

This study [10] points out to safety requirements at the software level of a self-driving car consisting of listed methods

978-1-7281-8956-7/20/$31.00 ©2020 IEEE

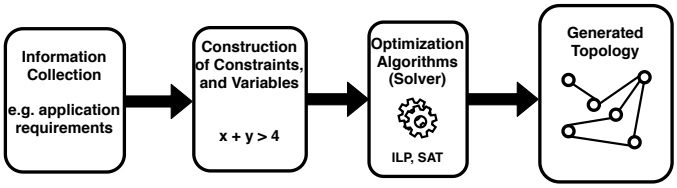

Fig. 1. The approach procedures including three steps.

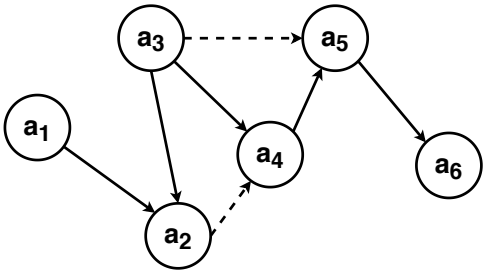

Fig. 2. Application graph explaining the data flow among the applications.

1. Redundancy at different levels of system 2. Software and hardware architecture proposal based on redundancy requirements 3. Usage of freedom from interference (FFI) technique 4. Run time monitoring of the software and hardware due to detect and avoid run-time failures. A safety monitoring method at run time in an autonomous system is introduced by Haupt [11]. This method utilizes a set of safety rules to detect safety-critical violations in the system.

Mulkul et al. [12] propose three models at the architectural level for driverless cars which assure functional safety of E/E components based on ISO26262. A model-based method to analyze the safety of an automated driving system is addressed by this group of researchers [13]. In this approach environment as well as perception, fusion, and control units used in an autonomous car have been modeled.

Time-Sensitive Networking (TSN) standards [14] developed by the Institute of Electrical and Electronics Engineers (IEEE) address the hard real-time requirements of Ethernet-based distributed applications. The main objective of these standards is to support the implementation of distributed applications with different levels of criticality in the same network infrastructure. Especially for the automotive domain developed IEEE 802.1DG TSN standard [15], specifies profiles for secure, highly reliable (e.g. using IEEE 802.1CB for seamless redundancy [16]), deterministic latency, automotive in-vehicle bridged IEEE 802.3 Ethernet networks. Traffic planning and verification of these networks require advanced expertise and are time-consuming; a motivation to develop adequate scheduling approaches [3], modeling and verification tools (including experimental setups) [17], [19]–[21]. Moreover, recent works also deal with consideration of safety aspects in routing decisions [22], [23] to increase the reliability of the critical applications.

The mentioned solutions assume a predefined topology and analyze the fulfillment of the safety requirements (e.g. timing and routing). However, they do not address requirement-based generation of the topology.

III. Approach

To create a requirement-based E/E topology, our approach consists of three major procedures (see Fig. 1) listed in the following:

1) Collection of system information regarding the requirements of the application (e.g. data redundancy)
2) Defining variables and constraints of the topology for solving
3) Generate topology based on the defined constraints using solving technology including e.g. ILP (Integer Linear Programming) or SAT (Boolean satisfiability problem)

To illustrate our approach, we use a visualized example. In this example, topologies are created based on an application graph with safety requirements (see Fig. 2) such as the routing redundancy, also the cost of link (to be defined in terms of length, weight, and material) in the topology.

As an example, an application graph (see Fig. 2) is considered. This graph consists of six applications in which their data flow are displayed by directed edges in Fig. 2. Two dashed line arrows illustrate the high-criticality data flow existing between a_3,a_5 and a_2,a_4 applications which means that these data flows must be redundant.

A topology is displayed in Fig. 3 which comprises nodes, links, and links costs as well as the applications. All six applications are assigned to specific nodes. The redundancy requirement of the data flow connection between a_3 and a_5 applications is not met. There is only one path between N_{12} and N_{11}. Green arrow in Fig. 3 shows two links (L_{11}, L_{12}) with the cost including 17 units in total.

Furthermore, for the other two critical applications a_2 and a_4 assigned to the N_9 and N_3 nodes respectively, there are four various routes to transmit data, visualizing two of them with blue and red arrows regarding the Fig. 3, cause in meeting the safety requirement (routing redundancy) of these applications.

Therefore, an example of a generated optimized architecture with utilizing the same assets in Fig. 3 topology, and based on the predefined safety and cost requirements is presented in Fig. 4a. It includes twelve nodes including the six assigned

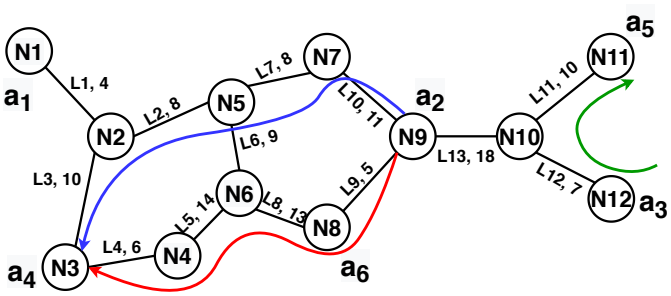

Fig. 3. A topology graph displaying nodes, links, links costs, and assigned applications.

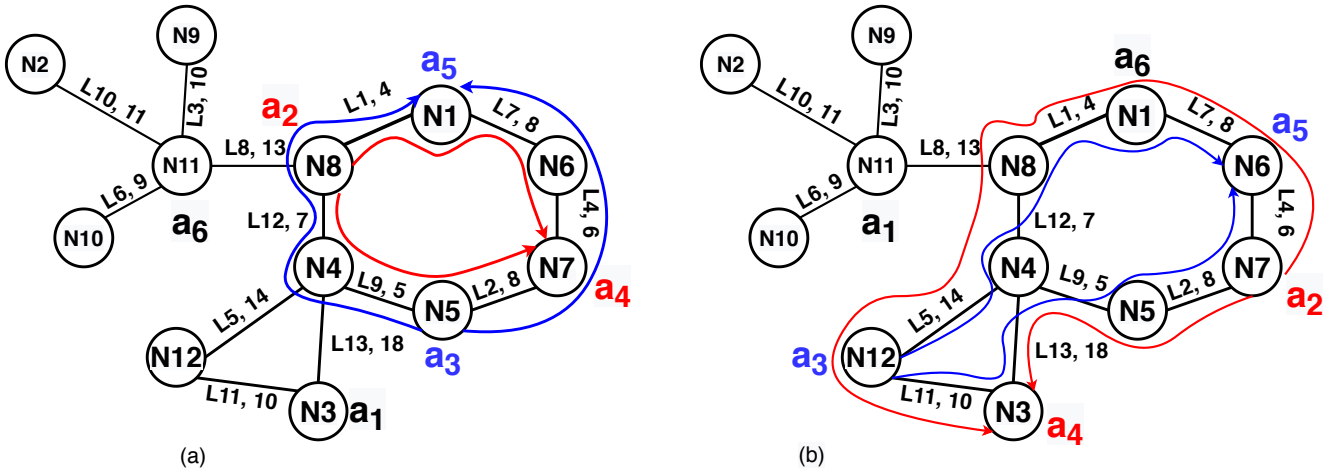

Fig. 4. The generated graphs: a) Cost-effective topology b) Cost-ineffective topology

applications and thirteen links comprising cost weights. It can be seen that the routes, allocated nodes for each app, and the shape of the topology are changed considerably rather than the previous topology. Also, a redundant path is devoted to data flow between a_3 and a_5 applications (shown with blue color in Fig. 4.a) with a cost of 22 and 16 units. This topology meets our predefined safety requirements including routing redundancy for a_3 and a_5 rather than the last topology and is still affordable in terms of the links cost.

Moreover, in this generated topology, the cost of the routes between critical applications a_2 and a_4, has been optimized 50 percent approximately compared to the last topology in Fig. 3. For instance, the blue and red paths cost 37 and 38 units respectively (see Fig. 3) while the critical paths for these two applications cost 18 and 20 according to the red directed edges in Fig. 4a.

As a result, the topology visualized in Fig. 4.a is known as an optimized architecture compared to Fig. 2 judged based on meeting our predetermined requirements (cost and routing redundancy). However, finding an optimal topology based on the requirements and constraints is not always feasible.

For instance, Fig. 4.b describes another generated topology for the same architecture properties as Fig. 4.a. It meets similar safety requirement (routing redundancy) and also includes similar topology aspects comprising same utilized links between the nodes, and architectural shape as Fig. 4.a. However, bringing attention to the various applications assignment among the nodes, illustrates that new paths for a_3,a_5 and a_4,a_2 (blue and red arrows respectively in Fig. 4.b) including 47, 33 and 49, 31 cost units for blue and red routes respectively (see Fig. 4.b).

With comparison between two graphs in Fig. 4, it is realized that Fig. 4.a is obviously more optimized than the generated topology in Fig. 4.b in terms of the routing expenses.

Finding an optimized generated graph is not a trivial task. For instance, the number of possible paths between two arbitrary nodes in a complete graph has a worst-case complexity of $O(n!)$ which significantly increases the computation time

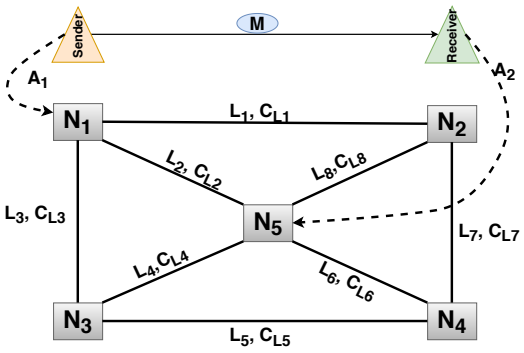

Fig. 5. A graph visualizing variables to create the system constraints

of solving the path-related constraints.

A. Formulating of topology variables

In order to visualize the input variables of our approach, a graph in Fig. 5 is presented. In this graph, we consider a network consists of nodes (N), links (L), senders and receivers as processes, cost of each link as well as each node (C_L,C_N), process assignment to each node (A), and routed message between the sender and receiver (M). We utilize the graph-based model approach introduced in [23].

In the following, the variables of our proposed topology are listed and shortly described.

- $\mathbf{A}\epsilon[0,1]$: assignment A is utilized in the architecture graph to allocate a specific node to a process
- $\mathbf{N}\epsilon[0,1]$: node N is allocated in the architecture graph
- $\mathbf{L}\epsilon[0,1]$: link L is devoted in the architecture graph
- $\mathbf{M_N}\epsilon[0,1]$: message M is sent over node N
- $\mathbf{M_{(N,P)}}\epsilon[0,1]$: the transmission signal of message M to process P is sent over node N
- $\mathbf{M_L}\epsilon[0,1]$: message M is sent over the link L
- $\mathbf{M_{(L,P)}}\epsilon[0,1]$: the transmission signal of message M to process P is sent over the link L
- $\mathbf{C_L}\epsilon\mathbb{R}$: the cost of link L is allocated in the architecture graph

- $\mathbf{C_N}\epsilon\mathbb{R}$: the cost of node N is devoted in the architecture graph

Furthermore, three equations are considered as our optimization goals to generate optimized topology concerning our defined variables. The objective of Eq. 1 is a minimization of the total number of nodes as well as their costs in the topology whereas Eq. 2 expresses the same concept for the links. Finally the Eq. 3 goal as a multi-objective equation is lessening the total cost of the topology following predefined variables comprising link and node costs.

Optimization Goals:

$$\min_{N*C_N} \quad \sum_{i=1}^{h} N_i * C_{N_i},\ i = 1, \ldots, h. \tag{1}$$

$$\min_{L*C_L} \quad \sum_{j=1}^{k} L_j * C_{L_j},\ j = 1, \ldots, k. \tag{2}$$

$$\min_{C_L, C_N} \quad \sum_{j=1}^{k} C_{L_j} + \sum_{i=1}^{h} C_{N_i} \tag{3}$$

IV. Further Steps & conclusion

In this paper, we propose an approach to produce requirement-based E/E topologies taking into account the routing redundancy as well as financial aspects at the beginning of the E/E architecture design phase. We use an illustrative example to explain the approach and formulate required variables for our constraint-based optimization system as well as defining three optimization objectives. Our future work consists of the following steps, the concrete constraints for routing will be formulated as an ILP optimization problem. Secondly, an appropriate ILP solver will be selected to implement the formulated constraints. Thirdly, we plan to generate problem sets with a various number of applications, data flow, and different redundancy requirements. Finally, the performance of our optimization approach will be evaluated by defining different metrics such as computation time, and the number of allocated nodes and edges.

Based on this evaluation, the constraints formulation will be improved iteratively. Moreover, the modular architecture of our optimization framework will guarantee the capability for future extensions to cover other functional safety aspects related to routing. For example, node capabilities including the operating system and the link properties can be considered and formulated as safety-related constraints.

References

[1] Koopman, P. and Wagner, M., 2017. Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intelligent Transportation Systems Magazine, 9(1), pp.90-96.

[2] Richard Bellairs,'What Is ISO 26262? An ISO 26262 Overview', 2019. [Online]. Available: https://www.perforce.com/blog/qac/what-is-iso-26262/. [Accessed: 18- May- 2020].

[3] Craciunas, S. S., Oliver, R. S., Chmelík, M., and Steiner, W. (2016, October). Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks. In Proceedings of the 24th International Conference on Real-Time Networks and Systems.

[4] Mody, M., Jones, J., Chitnis, K., Sagar, R., Shurtz, G., Dutt, Y., Koul, M., Biju, M.G. and Dubey, A., 2018. Understanding Vehicle E/E Architecture Topologies for Automated Driving: System Partitioning and Tradeoff Parameters. Electronic Imaging, 2018(17), pp.358-1.

[5] Zerfowski D., Lock A. (2019) Functional architecture and E/E-Architecture – A challenge for the automotive industry. In: Bargende M., Reuss HC., Wagner A., Wiedemann J. (eds) 19. Internationales Stuttgarter Symposium. Proceedings. Springer Vieweg, Wiesbaden

[6] van Dijk, L. and Sporer, G., 2018. Functional safety for automotive ethernet networks. Journal of Traffic and Transportation Engineering, 6(4), pp.176-182.

[7] Abdulkhaleq, A., Wagner, S., Lammering, D., Boehmert, H. and Blueher, P., 2017. Using stpa in compliance with iso 26262 for developing a safe architecture for fully automated vehicles. arXiv preprint arXiv:1703.03657.

[8] Yoneda, T., Imai, M., Saito, H., Mochizuki, A., Hanyu, T., Kise, K. and Nakamura, Y., 2019. Network-on-Chip Based Multiple-Core Centralized ECUs for Safety-Critical Automotive Applications. In VLSI Design and Test for Systems Dependability (pp. 607-633). Springer, Tokyo.

[9] Aniculaesei, A., Arnsberger, D., Howar, F. and Rausch, A., 2016. Towards the verification of safety-critical autonomous systems in dynamic environments. arXiv preprint arXiv:1612.04977.

[10] Chitnis, K., Mody, M., Swami, P., Sivaraj, R., Ghone, C., Biju, M.G., Narayanan, B., Dutt, Y. and Dubey, A., 2017. Enabling functional safety ASIL compliance for autonomous driving software systems. Electronic Imaging, 2017(19), pp.35-40.

[11] Haupt, N.B. and Liggesmeyer, P., 2019, September. A Runtime Safety Monitoring Approach for Adaptable Autonomous Systems. In International Conference on Computer Safety, Reliability, and Security (pp. 166-177). Springer, Cham.

[12] Gosavi, M.A., Rhoades, B.B. and Conrad, J.M., 2018, April. Application of Functional Safety in Autonomous Vehicles Using ISO 26262 Standard: A Survey. In SoutheastCon 2018.

[13] Tlig, M., Machin, M., Kerneis, R., Arbaretier, E., Zhao, L., Meurville, F. and Van Frank, J., 2018, June. Autonomous Driving System: Model Based Safety Analysis. In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W).

[14] Institute of Electrical and Electronics Engineers, Inc, 'Time-Sensitive Networking (TSN) Task Group'. [Online]. Available: https://1.ieee802.org/tsn/. [Accessed: 21- May- 2020].

[15] Institute of Electrical and Electronics Engineers, Inc, 'TSN Profile for Automotive In-Vehicle Ethernet Communications'. [Online]. Available: https://1.ieee802.org/tsn/802-1dg/. [Accessed: 15- May- 2020].

[16] Institute of Electrical and Electronics Engineers, Inc, 'Frame Replication and Elimination for Reliability'. [Online]. Available: https://1.ieee802.org/tsn/802-1cb/. [Accessed: 7- May- 2020].

[17] Farzaneh, M. H., Kugele, S., and Knoll, A. (2017, September). A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking. In 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).

[18] Farzaneh, M. H., and Knoll, A. (2016, October). An ontology-based plug-and-play approach for in-vehicle time-sensitive networking (tsn). In IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

[19] Farzaneh, M. H., Shafaei, S., and Knoll, A. (2016, December). Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming. In IEEE Vehicular Networking Conference (VNC).

[20] Farzaneh, M. H., and Knoll, A. (2017, November). Time-sensitive networking (TSN): An experimental setup. In IEEE Vehicular Networking Conference (VNC).

[21] Ashjaei, M., Patti, G., Behnam, M., Nolte, T., Alderisi, G., and Bello, L. L. (2017). Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support. Real-Time Systems, 53(4), 526-577.

[22] Smirnov, F., Glaß, M., Reimann, F., and Teich, J. Optimizing message routing and scheduling in automotive mixed-criticality time-triggered networks. In 2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC).

[23] Smirnov, F., Reimann, F., Teich, J., Han, Z., and Glaß, M. (2018, May). Automatic optimization of redundant message routings in automotive networks. In Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems.