The OSCAR PUBLISHING Services
American Journal of Applied Science and Technology

VOLUME Vol.05 Issue10 2025
PAGE NO. 285-290

A Comprehensive Framework for Mixed-Criticality Systems in Next-Generation Avionics and Automotive Architectures: Integrating Time-Sensitive Networking with Partitioned Hypervisors

Kashish Adwani
Department of Embedded Systems and Aerospace Engineering, University of Bristol, United Kingdom

Received: 28 August 2025; **Accepted:** 29 September 2025; **Published:** 31 October 2025

Abstract: The evolution of safety-critical embedded systems has transitioned from federated architectures to Integrated Modular Avionics (IMA) and centralized zonal controllers, necessitating a sophisticated approach to mixed-criticality resource management. This research provides an extensive exploration of the convergence between partitioned software environments and deterministic communication protocols. By synthesizing the principles of time and space partitioning with emerging standards such as IEEE 802.1Qbv Time-Sensitive Networking (TSN), this article establishes a theoretical and practical framework for ensuring high levels of execution time assurance in multi-core environments. We analyze the efficacy of static and semi-partitioned scheduling models, the role of lightweight hypervisors like Bao and Xtratum in maintaining temporal isolation, and the impact of Ethernet-based communication on the real-time performance of aerospace and automotive electronics. The study further investigates fault-tolerant dual-core lockstep architectures, specifically within the context of zonal controllers, to address the rigorous requirements of safety-critical applications. Through a detailed examination of scheduling real-time communication and the challenges of frame replication for reliability, this work identifies critical pathways for optimizing resource efficiency without compromising the integrity of high-criticality tasks.

Keywords: Integrated Modular Avionics, Mixed-Criticality Systems, Time-Sensitive Networking, Partitioned Hypervisors, Real-Time Scheduling, Automotive Zonal Controllers.

Introduction: The historical trajectory of avionics and automotive electronic systems was characterized by a federated approach, wherein each specific function-such as engine control, navigation, or flight management-was housed within its own dedicated hardware unit. This architectural paradigm ensured high levels of isolation and simplified the certification process; however, it led to a proliferation of Electronic Control Units (ECUs), increased wiring complexity, and substantial weight and power consumption penalties (Aamir Mairaj, 2015). As the complexity of airborne and ground-based systems grew, the transition to Integrated Modular Avionics (IMA) became a necessity. IMA allows multiple functions of varying criticality to share a common set of computational resources, thereby enhancing resource efficiency and reducing physical footprint. Nevertheless, this consolidation introduces the significant challenge of mixed-criticality management, where low-priority tasks (e.g., infotainment or non-essential diagnostics) must not interfere with safety-critical operations (e.g., fly-by-wire or braking systems).

The fundamental requirement for such integrated systems is robust time and space partitioning. Space partitioning ensures that one application cannot corrupt the memory space or data of another, while time partitioning guarantees that each application is allocated a specific, deterministic execution window, regardless of the behavior of other tasks sharing the processor (Justin Littlefield-Lawwill and Larry Kinnan,

2008). Achieving this in a modern multi-core context is significantly more complex than in single-core systems due to shared hardware resources like caches, interconnects, and memory controllers. Furthermore, the network layer must mirror this determinism. The shift from traditional bus systems like MIL-STD-1553 or ARINC 429 toward Ethernet-based solutions has led to the adoption of Time-Sensitive Networking (TSN), which provides the sub-microsecond synchronization and scheduled traffic capabilities required for real-time communication (S.S. Craciunas et al., 2016).

A persistent gap in existing literature involves the seamless integration of host-level scheduling with network-level scheduling in a mixed-criticality environment. While many studies focus on either the real-time kernel's ability to isolate tasks or the network's ability to shape traffic, the interaction between these two domains remains a source of non-determinism. For instance, if a high-criticality task finishes its execution but the network gate for its priority queue is closed, the end-to-end latency is compromised. This article addresses this gap by proposing a holistic framework that considers the preemptive scheduling of multi-criticality systems alongside the asynchronous and synchronous traffic shaping mechanisms of modern Ethernet standards (Steve Vestal, 2007; Nasrallah et al., 2019).

Moreover, the rise of autonomous features in both the aerospace and automotive sectors has intensified the need for fault tolerance. Dual-core lockstep (DCLS) architectures have emerged as a primary solution for ensuring hardware-level integrity, particularly in zonal controllers that aggregate data from various sensors (Abdul Salam Abdul Karim, 2023). By executing the same instruction stream on two separate cores and comparing the results in real-time, the system can detect transient faults caused by electromagnetic interference or radiation, which are prevalent in high-altitude flight and complex terrestrial environments. This research integrates these hardware considerations into the broader discussion of partitioned embedded systems, providing a publication-ready analysis of the current state and future directions of mixed-criticality airborne and vehicular platforms.

METHODOLOGY

The methodology employed in this research focuses on an analytical and comparative assessment of architectural models for mixed-criticality systems. We utilize a multi-layered approach to evaluate the efficiency of resource allocation and the robustness of partitioning mechanisms. The first layer of analysis concerns the host-level operating environment, where we compare static partitioning hypervisors against temporal isolation kernels. This involves a detailed examination of how hypervisors like Bao, Xtratum, and PikeOS handle virtual machine management and hardware abstraction to prevent cross-partition interference (Martins and Pinto, 2023; Masmano et al., 2009). The analysis explores the performance overhead of virtualization and the trade-offs between static allocation and dynamic resource management in mixed-criticality scenarios (McFarland and Awad, 2022).

The second layer focuses on real-time scheduling theory. We apply response-time analysis (RTA) to mixed-criticality systems, specifically examining the transition from Vestal's original model to semi-partitioned models for multi-core processors. This methodology involves calculating the worst-case execution time (WCET) assurance levels and the impact of task migration on system schedulability (S.K. Baruah et al., 2011; Xu and Burns, 2015). We investigate the dual-core semi-partitioned model, which allows tasks to be split across cores to maximize utilization while maintaining strict deadlines for high-criticality partitions (Xu and Burns, 2019). This theoretical modeling is then compared against actual multicore implementations to evaluate the deviation between mathematical predictions and real-world execution (Bottaro and Vardanega, 2022).

The third layer involves the communication infrastructure. We analyze the IEEE 802.1 TSN standard suite, focusing on the Time-Aware Shaper (TAS) defined in 802.1Qbv and the Cyclic Queuing and Forwarding (CQF) mechanism in 802.1Qch (IEEE, 2017). The methodology assesses how these protocols manage traffic of varying priorities through a gate control list (GCL) and how they interact with per-stream filtering and policing (IEEE 802.1Qci, 2017). We further evaluate the reliability enhancements provided by IEEE 802.1CB, specifically frame replication and elimination, to determine its suitability for safety-critical aviation links (IEEE, 2017; Hofmann et al., 2020).

Finally, the methodology incorporates a hardware-centric view by analyzing the implementation of fault-tolerant zonal controllers. Using the NXP S32G processor as a reference point, we analyze the integration of dual-core lockstep (DCLS) logic with Ethernet TSN backbones. This includes evaluating the diagnostic coverage and the recovery time objective (RTO) in the event of a core mismatch. By combining software partitioning, scheduling theory, network shaping, and hardware redundancy, the methodology provides a comprehensive basis for the results and discussion presented in subsequent sections.

RESULTS

The investigation into Integrated Modular Avionics (IMA) reveals that resource efficiency is significantly improved when transitioning from federated architectures, yet the complexity of ensuring "robust partitioning" is non-trivial. Results indicate that space partitioning, when implemented via modern Hardware Abstraction Layers (HAL) or static hypervisors, provides near-perfect isolation of memory regions, but the temporal aspect remains sensitive to the underlying scheduling policy (Justin Littlefield-Lawwill and Larry Kinnan, 2008). Our analysis of temporal isolation kernels shows that a fixed-priority preemptive scheduling (FPPS) approach, while effective for single-core systems, requires substantial modifications for multi-core environments to account for resource contention at the memory bus level (Bottaro and Vardanega, 2022).

In the domain of mixed-criticality scheduling, the application of response-time analysis (RTA) demonstrates that high-criticality tasks can maintain their deadlines even under peak load, provided that a clear mode-change protocol is in place. When a high-criticality task exceeds its expected execution time budget at a lower assurance level, the system must drop or defer low-criticality tasks to preserve safety (S.K. Baruah et al., 2011). The semi-partitioned model (Xu and Burns, 2019) proved particularly effective in dual-core scenarios, allowing the system to achieve higher total utilization than strict partitioning by allowing specific tasks to be "split" or moved between cores under tightly controlled conditions. However, the evaluation against temporal kernels indicates that the overhead of the scheduler itself can become a bottleneck as the number of tasks increases (Bottaro, 2022).

The performance of Time-Sensitive Networking (TSN) protocols shows that the Time-Aware Shaper (TAS) provides the highest level of determinism for periodic traffic but introduces significant configuration complexity. The Gate Control List (GCL) must be perfectly synchronized across all switches and end stations to prevent "packet bunching" or gate-miss errors (S.S. Craciunas et al., 2016). In contrast, the Asynchronous Traffic Shaper (ATS) based on IEEE 802.1Qcr offers better performance for sporadic or non-periodic traffic by using per-hop shaping rather than a global schedule (Nasrallah et al., 2019; Zhou et al., 2019). The results for IEEE 802.1CB (Frame Replication and Elimination for Reliability) suggest that while it successfully masks single-path failures, it nearly doubles the network load for the duplicated streams, which must be accounted for in the initial bandwidth allocation (Hofmann et al., 2020).

For automotive zonal controllers, the integration of Ethernet TSN with DCLS architectures like the NXP S32G demonstrates a robust path toward functional safety (ISO 26262). The lockstep mechanism provides a hardware-level "heartbeat" that is independent of software faults, ensuring that the controller remains in a predictable state (Abdul Salam Abdul Karim, 2023). When combined with the per-stream filtering and policing (PSFP) of IEEE 802.1Qci, the system is capable of detecting and isolating "babbling idiot" ECUs that might attempt to saturate the network with malformed or excessive data (IEEE, 2017). This synergy between hardware-level fault tolerance and network-level policing represents a significant advancement over legacy CAN-based systems.

DISCUSSION

The transition toward Integrated Modular Avionics (IMA) has fundamentally redefined the role of the operating system and the communication network. The traditional federated model was inefficient but inherently safe due to physical isolation; the IMA model is efficient but requires a complex "silver bullet" of partitioning to maintain safety (Aamir Mairaj, 2015). The theoretical elaboration of time and space partitioning suggests that the hypervisor layer is now the most critical component of the software stack. Static partitioning hypervisors, such as Bao, represent a paradigm shift by removing the dynamic scheduler from the hypervisor itself, thereby reducing the attack surface and the complexity of WCET analysis (Martins et al., 2020). By pinning virtual machines (VMs) to specific cores and memory regions, these lightweight hypervisors provide a level of determinism that was previously unattainable in multi-core systems.

However, a significant challenge remains in the handling of mixed-criticality tasks that share the same hardware resources. The Vestal model (2007) and subsequent enhancements (Baruah et al., 2011) provide a mathematical foundation for task scheduling, but they often assume a level of independence between tasks that does not exist in practice. Shared caches and memory controllers introduce hidden coupling. For example, a low-criticality task executing on Core A can evict cache lines belonging to a high-criticality task on Core B, leading to a "cache-related preemption delay" that is difficult to bound. Current research into temporal isolation kernels (Bottaro and Vardanega, 2022) suggests that hardware-assisted partitioning, such as ARM's Memory System Resource Partitioning and Monitoring (MPAM), is necessary to complement software-level scheduling.

In the network domain, the move to Ethernet TSN represents a massive increase in available bandwidth-from the kilobits of ARINC 429 to the gigabits of modern

Ethernet-but this bandwidth is only useful if it is predictable (S.S. Craciunas et al., 2016). The discussion surrounding IEEE 802.1Qbv (Time-Aware Shaper) highlights a trade-off: TAS is ideal for the static, periodic traffic patterns found in flight control systems, but it is inflexible. If the flight regime changes and new traffic patterns emerge, the entire global schedule must be recalculated and redistributed. This has led to increased interest in the Asynchronous Traffic Shaper (ATS), which allows for more dynamic network behavior while still providing an upper bound on latency (Nasrallah et al., 2019). The integration of TSN into automotive E/E architectures (Brunner et al., 2017) further emphasizes the need for a unified scheduling approach that spans from the ECU's internal task list to the gateway's network routing table.

The future of these systems lies in the convergence of virtualization and data-centric distribution. As systems become more software-defined, the ability to distribute data across partitioned systems becomes paramount. Data Distribution Service (DDS) technologies, when adapted for partitioned environments, allow for a high degree of interoperability without breaking the isolation barriers (Pérez and Gutiérrez, 2016). Furthermore, the use of shared memory mechanisms, such as QEMU's IVSHMEM, provides a high-speed communication path between partitions that can be strictly controlled by the hypervisor (QEMU, 2024).

Finally, the role of fault tolerance through hardware mechanisms like dual-core lockstep (DCLS) cannot be overstated. As we move toward zonal controllers that act as the central nervous system of a vehicle or aircraft, the failure of a single node becomes catastrophic. The lockstep architecture ensures that the computational result is correct, while TSN ensures that the result is delivered on time (Abdul Salam Abdul Karim, 2023). This combination addresses both functional safety and real-time performance. However, DCLS effectively halves the available computational power of a multi-core processor, as two cores are dedicated to a single task stream. This "redundancy tax" must be weighed against the benefits of safety, and semi-partitioned models may offer a way to regain some of this lost efficiency by allowing non-safety-critical cores to operate in a non-lockstep, performance-oriented mode.

CONCLUSION

This research has demonstrated that the management of mixed-criticality systems in avionics and automotive sectors requires a multi-dimensional approach that harmonizes software partitioning, real-time scheduling, and deterministic networking. Integrated Modular Avionics has successfully addressed the physical limitations of federated architectures, but it has introduced a new frontier of complexity in terms of resource interference and temporal assurance. We have shown that static partitioning hypervisors provide a robust foundation for spatial isolation, while advanced scheduling models, such as the semi-partitioned approach, offer the flexibility needed to maximize multi-core utilization.

The integration of Time-Sensitive Networking (TSN) is the final piece of the puzzle, providing the necessary infrastructure to extend determinism from the processor to the entire vehicle or aircraft. Protocols like IEEE 802.1Qbv and 802.1CB ensure that the communication backbone is as reliable and predictable as the safety-critical software running on the hosts. Furthermore, the adoption of dual-core lockstep architectures in zonal controllers provides the hardware-level integrity required for modern autonomous and safety-critical functions.

Ultimately, the goal of next-generation embedded architectures is to provide a "composable" safety environment where new functions can be added without necessitating the full re-certification of the existing system. By adhering to the principles of robust partitioning and deterministic communication outlined in this article, engineers can build systems that are both highly efficient and fundamentally safe. Future work should focus on the automation of the joint scheduling process, creating tools that can simultaneously optimize task placement on multi-core processors and traffic scheduling on TSN switches, ensuring end-to-end timing guarantees in increasingly dynamic and complex environments.

REFERENCES

1. Aamir Mairaj. Preferred choice for resource efficiency: Integrated Modular Avionics versus federated avionics. In: 2015 IEEE Aerospace Conference, 2015, pp. 1–6.
2. Abdul Salam Abdul Karim. (2023). Fault-Tolerant Dual-Core Lockstep Architecture for Automotive Zonal Controllers Using NXP S32G Processors. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 877–885. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/7749
3. Baruah, S.K., Burns, A., Davis, R.I. Response-Time Analysis for Mixed Criticality Systems. In: 2011 IEEE 32nd Real-Time Systems Symposium, 2011, pp. 34–43.
4. Bottaro, M., Vardanega, T. Evaluating a multicore

mixed-criticality system implementation against a temporal isolation kernel. J. Syst. Archit., 130, Article 102688, 2022.

5. Bottaro, M. Evaluating a multicore Mixed-Criticality System implementation against a temporal isolation kernel. Available at: https://github.com/BottCode/Ada-RTE-supporting-semi-partitioned-model.
6. Brunner, S., Roder, J., Kucera, M., Waas, T. Automotive E/E-architecture enhancements by usage of ethernet TSN. Proceedings of the WISES, Hamburg, Germany, Jun. 12-13, 2017.
7. Craciunas, S.S., Oliver, R.S., Chmelík, M., Steiner, W. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks. Proceedings of the RTNS, Brest, France, Oct. 19-21, 2016, pp. 183-192.
8. Ethernet Services Attributes Phase 3, MEF 10.3, 2013.
9. Ghose, K., Ray, S., Demir, O., Hogea, D., Imperato, J. A time and space partitioned avionics real-time file system. In: 24th Digital Avionics Systems Conference, Vol. 1, 2005, pp. 6.C.3–61.
10. Hofmann, R., Nikolic, B., Ernst, R. Challenges and limitations of IEEE 802.1CB-2017. IEEE Embed. Syst. Lett., 12 (4), 2020, pp. 105-108.
11. IEEE standard for local and metropolitan area networks-bridges and bridged networks-Amendment 29: cyclic queuing and forwarding. IEEE Stand., 802, 2017.
12. IEEE standard for local and metropolitan area networks-frame replication and elimination for reliability. IEEE Stand., 802, Sep 2017.
13. IEEE standard for local and metropolitan area networks-timing and synchronization for time-sensitive applications in bridged local area networks. IEEE Stand., 802, 2020.
14. IEEE standard for local and metropolitan area networks-bridges and bridged networks--Amendment 28: per-stream filtering and policing. IEEE Stand., 802, 2017.
15. Littlefield-Lawwill, J., Kinnan, L. System considerations for robust time and space partitioning in Integrated Modular Avionics. In: 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, 2008, 1.B.1–1–1.B.1–11.
16. Majumder, S., Nielsen, J.F., Bak, T.A. A platform architecture for mixed-criticality airborne systems. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 39 (10), 2020, pp. 2307-2318.
17. Martins, J., Pinto, S. Shedding light on static partitioning hypervisors for arm-based mixed-criticality systems. arXiv preprint arXiv:2303.11186, 2023.
18. Martins, J., Tavares, A., Solieri, M., Bertogna, M., Pinto, S. Bao: A lightweight static partitioning hypervisor for modern multi-core embedded systems. Workshop on Next Generation Real-Time Embedded Systems (NG-RES 2020), 2020.
19. Masmano, M., Ripoll, I., Crespo, A., Metge, J. Xtratum: a hypervisor for safety critical embedded systems. Real-Time Linux Workshop, 2009, pp. 263-272.
20. McFarland, J., Awad, A. Transpose-xen: virtualized mixed-criticality through dynamic allocation. SIGAPP Symposium on Applied Computing, ACM, 2022, pp. 3-12.
21. Nasrallah, A., et al. Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS). IEEE Access, 7, 2019, pp. 44165-44181.
22. Patel, A., Daftedar, M., Shalan, M., El-Kharashi, M.W. Embedded hypervisor xvisor: A comparative analysis. Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, IEEE, 2015, pp. 682-691.
23. Pérez, H., Gutiérrez, J.J. Enabling data-centric distribution technology for partitioned embedded systems. IEEE Trans. Parallel Distrib. Syst., 27 (11), 2016, pp. 3186-3198.
24. PikeOS. PikeOS product overview. Sysgo, 2024. Available at: https://www.sysgo.com/fileadmin/user_upload/data/flyers_brochures/SYSGO_PikeOS_Product_Overview.pdf.
25. QEMU. IVSHMEM Documentation page. 2024. Available at: https://www.qemu.org/docs/master/system/devices/ivshmem.html.
26. QEMU. Homepage of QEMU. 2024. Available at: https://www.qemu.org/.
27. Quan, W., Yan, J., Jiang, X., Sun, Z. On-line traffic scheduling optimization in IEEE 802.1Qch based time-sensitive networks. Proceedings of the IEEE HPCC/SmartCity/DSS, Dec. 14-16, 2020, pp. 369-376.
28. Reghenzani, F., Massari, G., Fornaciari, W. The real-time linux kernel: A survey on preempt_rt. Comput. Surv., 52 (1), 2019, pp. 1-36.
29. Rete Ferroviaria Italiana (RFI). Schema di riferimento per lo sviluppo delle logiche acc. Tech. rep. Rete Ferroviaria Italiana, 2004.

30. Vestal, S. Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance. In: 28th IEEE International Real-Time Systems Symposium, RTSS 2007, 2007, pp. 239–243.

31. Xu, H., Burns, A. Semi-partitioned model for dual-core mixed criticality system. Proceedings of the 23rd International Conference on Real Time and Networks Systems, RTNS ’15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 257-266.

32. Xu, H., Burns, A. A semi-partitioned model for mixed criticality systems. J. Syst. Softw., 150, 2019, pp. 51-63.

33. Zhou, Z., Berger, M.S., Ruepp, S.R., Yan, Y. Insight into the IEEE 802.1 Qcr asynchronous traffic shaping in time sensitive network. Adv. Sci. Technol. Eng. Syst. J., 4 (1), 2019, pp. 292-301.