Specification and Analysis of AeroRing -A Full Duplex Ethernet Ring Network for New Generation Avionics Systems

Ahmed Amari

Université Fédérale

Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace

Présentée et soutenue par :
Ahmed AMARI

le jeudi 21 septembre 2017

Titre :

Conception et validation d'AeroRing - un réseau de communication Ethernet en double anneau pour les systèmes avioniques de nouvelle génération

Specification and Analysis of AeroRing - A Full Duplex Ethernet Ring Network for New Generation Avionics Systems

---

École doctorale et discipline ou spécialité :

ED MITT : Réseaux, télécom, système et architecture

Unité de recherche :

Équipe d'accueil ISAE-ONERA MOIS

Directeur(s) de Thèse :

M. Jérôme LACAN (directeur de thèse)
Mme Ahlem MIFDAOUI (co-directrice de thèse)

Jury :

M. Emmanuel GROLLEAU, Professeur ISAE-ENSMA - Président, Rapporteur
M. Fabrice FRANCES, Professeur Associé ISAE-SUPAERO
M. Marc GATTI, Directeur R&T Thales Avionics
M. Jérôme LACAN, Professeur ISAE-SUPAERO - Directeur de thèse
Mme Ahlem MIFDAOUI, Professeure ISAE-SUPAERO - Co-directrice de thèse
Mme Pascale MINET, Chargée de recherche INRIA Paris - Rapporteur

Acknowledgements

I would like to express my sincere gratitude and appreciation to my Principal Supervisor, Ahlem Mifdaoui, whom I am very fortunate to have worked with, for her constant support, guidance and encouragement in my thesis journey. To my PhD director Jérôme Lacan and co-supervisor Fabrice Frances for their care and support.

I would like to thank all my family members, my parents, my brothers, Salim and Farouk, and my sisters, Soumia and Hadjer, who have always supported and encouraged me in my thesis and my life.

I would like to thank Younes, Mohamed, Billal, Sid Ahmed, Houssam, Driss, Yasmina, Rafik, Imad, Aziz, Abla and all my friends for the fun and their support, as well as my friends and colleagues that I worked with at the DISC department of ISAE-SUPAERO, Emmanuel, Tanguy, Anass, Vatsal, Anais, Sophie, Luca, Erwan, Jonathan, Bastien, Rami, Karine, Frederic, Antoine, Victor and all the rest.

Ahmed Amari

Toulouse, September 2017

Abstract

The inherent complexity and bandwidth requirement of avionics communication architectures are increasing due to the growing number of interconnected end-systems and the expansion of exchanged data. The Avionics Full Duplex Switched Ethernet (AFDX) has been introduced to provide high-speed communication (100Mbps) for new generation aircraft. However, this switched network is deployed in a fully redundant way, which leads to significant quantities of wires, and thus increases weight and integration costs. To cope with these arising issues, integrating ring-based Ethernet network in avionics context is proposed in this thesis as a main solution to decrease the wiring-related weight and complexity. In this context, our main objective is to design and validate a new avionic communication network, called AeroRing, based on a Gigabit Ethernet technology and supporting a Full Duplex ring topology.

To achieve this aim, first, a benchmarking of the most relevant Real-Time Ethernet (RTE) solutions supporting ring topologies vs avionics requirements has been conducted, and we particularly assess the main Performance Indicators (PIs), specified in IEC 61784-2. This benchmarking reveals that each existing RTE solution satisfies some requirements better than others, but there is no best solution in terms of all the requirements.

Therefore, we have specified a new RTE solution, AeroRing, to guarantee a high timing performance and availability levels, while keeping the IEEE802.3 compatibility and reducing the configuration efforts. The main innovative features of AeroRing are: (i) distributed access mechanism allowing simultaneous data exchange, to increase the offered bandwidth and resource usage efficiency; (ii) distributed fault management mechanism avoiding any central point of failure, to provide high reliability and availability levels; (iii) event-triggered communication enhancing the system flexibility and decreasing the implementation complexity, through avoiding any need of synchronization; (iv) QoS management handling heterogeneous data constraints, through QoS-aware routing algorithm.

To analyze the effects of such a proposal on the avionics timing performance, we have modeled this solution using the Network Calculus formalism, which defines an arrival curve for each input flow and a service curve for each crossed node. Based on the existing iterative Network Calculus approaches supporting ring topologies, we have computed end-to-end delay bounds to verify the system timeliness. Preliminary performance evaluation through small-scale test cases reveals that these conventional methods lead to overly pessimistic upper bounds, decreasing the network scalability (number of interconnected nodes) and resource efficiency (network utilization rate).

To enable the computation of tighter end-to-end delay bounds, we have introduced a new global analysis approach, Pay Multiplexing Only at Convergence points (PMOC). This consists in considering the flow serialization phenomena along the path of a flow of interest (f.o.i), by paying the bursts of interfering flows only at the convergence points. Hence, we have defined and proved the guaranteed end-to-end service curve of any f.o.i crossing such a network. Then, the methodology to compute delay bounds have been presented for one ring and generalized to multiple-ring topologies. Finally, the first numerical results have highlighted the accuracy of our proposed approach, in comparison to conventional methods, which yields enhanced performance, in terms of resource efficiency and network scalability.

Afterwards, to analyze the reliability level of AeroRing, we have conducted a dependability study where we have analytically quantified the reliability level of AeroRing depending on several aspects such as: the network size, the equipment reliability (MTTF) and the mission time. For this, we have first described how we have modeled the failures with their occurrence and impact, as well as the AeroRing system model. The models are then built up using Stochastic Active Networks (SANs), a stochastic extension of Petri Nets (PNs). Finally, we have highlighted the high reliability level of AeroRing under different scenarios through a sensitivity analysis. The obtained results have shown the high reliability level of AeroRing which meets the Avionics required level.

Finally, the validation of AeroRing through a representative avionics setup of an A380 communication network has been conducted. The obtained results highlight the ability of AeroRing to guarantee the avionics requirements in terms of timeliness, scalability, resource efficiency and reliability.

Résumé

La complexité et le besoin en bande passante des architectures de communication avionique ne cessent de croitre avec le nombre des calculateurs et l'expansion des données échangées. La technologie AFDX a été introduite pour offrir des communications haut débit (100Mbps) pour les avions de nouvelle génération. Cependant, ce réseau commuté est déployé de manière entièrement redondante, ce qui conduit à des quantités importantes de câbles, augmentant le poids et les coûts d'intégration. Pour faire face à ces problèmes, on propose dans cette thèse l'intégration d'un réseau Ethernet en anneau comme une solution principale pour diminuer le poids et la complexité liés au câblage. Dans ce contexte, notre objectif est de concevoir et valider un nouveau réseau de communication avionique, AeroRing, basé sur de l'Ethernet Gigabit avec une topologie anneau.

Pour atteindre cet objectif, un benchmarking des solutions Ethernet (RTE) les plus pertinentes supportant les topologies anneau vis-à-vis des besoins en avionique a été réalisé, en évaluant en particulier les principaux indicateurs de performance (IP) spécifiés dans le document IEC 61784-2 [1]. Ce benchmarking a révélé que chacune des solutions RTE existantes ne satisfait que certaines exigences, mais qu'il n'y a pas de meilleure solution en termes de toutes les exigences.

Par conséquent, nous avons spécifié une nouvelle solution RTE, AeroRing, pour garantir les niveaux requis de performances et de disponibilité, tout en conservant la compatibilité IEEE802.3 et en réduisant les efforts de configuration. Les principales caractéristiques innovantes d'AeroRing sont les suivantes: (i) mécanisme d'accès distribué permettant l'échange simultané de données, pour augmenter la bande passante offerte et l'utilisation des ressources; (ii) un mécanisme distribué de gestion des pannes évitant tout point de défaillance central, ce qui permet de fournir des niveaux de fiabilité et de disponibilité élevés; (iii) communication à base d'événement améliorant la flexibilité du système et diminuant la complexité de l'implémentation, en évitant tout besoin de synchronisation; (iv) Gestion de la QoS (Quality of Service) prenant en compte des contraintes hétérogènes sur les données, grâce à un algorithme de routage orienté QoS (qualité de service).

Pour analyser les effets d'une telle proposition sur les performances temporelles de l'avionique, nous avons modélisé cette solution en utilisant le formalisme du Calcul Réseau (Network Calculus), en se basant tout d'abord sur des approches itératives existantes pour les topologies anneaux. L'évaluation de performance préliminaire a révélé que ces méthodes conduisent à des bornes excessivement pessimistes, et par conséquent à un passage à l'échelle et une utilisation de ressources limités.

Pour permettre le calcul des bornes maximales plus précises sur les délais de bout en bout, nous avons introduit une nouvelle approche d'analyse globale, Pay Multiplexing Only at Convergence points (PMOC), qui prend en compte les phénomènes de sérialisation de flux, en considérant l'impact des flux interférents seulement aux points de convergence. Les premiers résultats ont mis en évidence l'amélioration des bornes calculées avec notre approche, par rapport aux autres méthodes. Ceci a permis d'améliorer les performances, en termes de passage à l'échelle et d'utilisation des ressources.

Par la suite, pour analyser le niveau de fiabilité d'AeroRing, nous avons mené une étude de fiabilité où le niveau de fiabilité d'AeroRing a été quantifié analytiquement, en fonction de plusieurs paramètres. Les résultats obtenus ont montré le niveau de fiabilité élevé d'AeroRing, satisfaisant les exigences de l'avionique. Enfin, la validation d'AeroRing via une configuration représentative d'un réseau de communication avionique d'un A380 a été menée. Les résultats obtenus ont mis en évidence la capacité d'AeroRing à garantir les exigences avioniques, en termes de déterminisme, passage à l'échelle, utilisation des ressources et fiabilité.

Contents

List of Figures

List of Tables

Chapter 1

Problem Statement

Contents

Nowadays, the complexity of embedded systems is growing in several domains, such as civil and military avionics, aerospace and railways. This complexity, particularly in avionics, is due to the increasing number of functions and the amount of exchanged data over the last few decades. As shown in Fig. 1.1, since the A300, the number of electronic equipment is constantly increasing [2] to offer new functionalities, improving performance, safety, maintenance and passenger comfort.

To cope with these emerging avionics needs, high speed communication networks have been integrated in new generation aircraft. For instance, the 100Mbps Avionics Full-Duplex Switched Ethernet (AFDX) [8] is used as a backbone network, to interconnect the critical avionics systems in the A380 and A350. However, low rate buses such as ARINC 429 [9, 10] and CAN [11], are still used for the sensors/actuators and the cabin communications.

Although this communication architecture meets the main avionics requirements, it leads at the same time to an inherent heterogeneity of the interconnection means and a large amount of cables and connectors; thus high integration costs. The costs related to cabling during the manufacturing and installation are actually between 14 million dollars for an A320 to 50 million dollars for an A787 [12]. Moreover, the cabling complexity is considered as one of the main reasons behind the production delays of the A380, where the cost overruns has been estimated at 2 billion dollars [12].

To handle these limitations, our objective in this thesis is to specify and validate a new high speed ring-based network meeting the avionics requirements, while limiting the cabling complexity and the deployment costs.

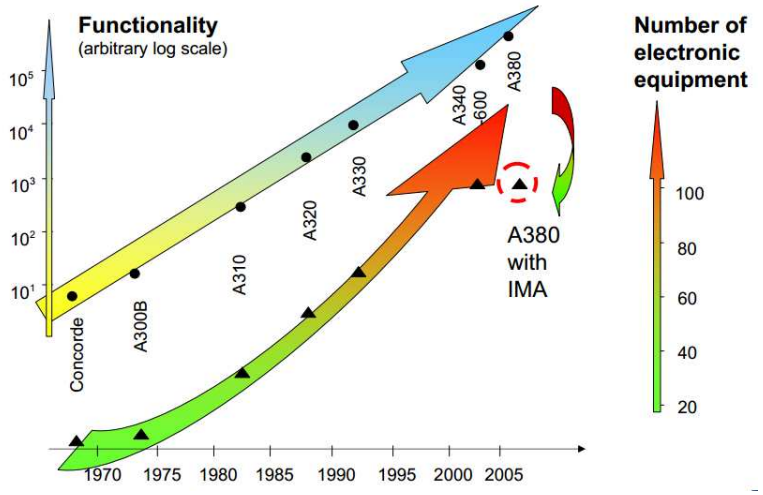


Figure 1.1: Evolution of the functionality and number of electronic equipment in avionics [2]

In this chapter, we first describe the avionics context to highlight the main characteristics and requirements that have to be fulfilled by an alternative solution. Afterwards, we give an overview of the most relevant solutions to improve avionics performance and relate them to our proposal. Finally, we detail our followed methodology and main contributions to design and validate our proposed solution.

1.1 Context and Problematic

As shown in Fig. 1.2, the current avionics network architecture consists of a fully redundant backbone network, based on the AFDX, which interconnects the critical avionics subsystems. These subsystems are responsible for flight control, cockpit, engines and landing gears. However, some legacy systems for I/O and cabin management are still connected to low rate buses. We describe herein these different networks and the main avionics requirements.

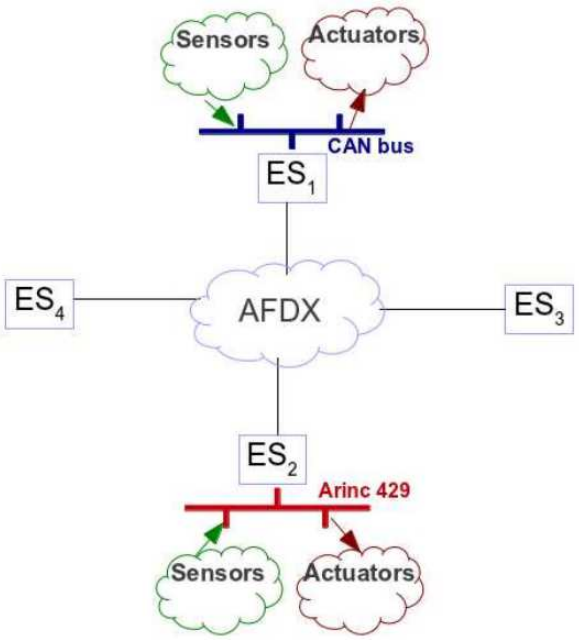


Figure 1.2: Avionic network architecture [3]

1.1.1 Backbone Network: ARINC 664

The Avionics Full-DupleX Switched Ethernet (AFDX), known also as ARINC 664 [8], is an Ethernet-compliant technology, which has been certified to meet the main avionics requirements. The main concepts of such a technology are: the Virtual Link (VL), the static data forwarding and the redundancy mechanisms.

Virtual Link The Virtual Link (VL) concept is a way to reserve a guaranteed bandwidth to each traffic flow. It uses a 16-bits value called a Virtual Link ID (VLID) to route the frames in the AFDX network. A VL is unidirectional and must always originate at one end-system and destined to a predetermined set of subsystems (unicast or multicast). Each VL defines a bandwidth contract expressed in terms of: i) BAG (Bandwidth Allocation Gap), ranging in powers of 2 from 1 to 128 milliseconds, which represents the minimal inter-arrival time between two consecutive frames; ii) MFS (maximum frame size), ranging from 64 to 1518 bytes, which represents the size of the largest frame that can be sent during each BAG.

Static Data Forwarding AFDX uses the standard Ethernet frame where the destination MAC address field contains the VLID. In addition, there is 1-byte sequence number to identify the redundant frames sent in the backup network. Fig. 1.3 shows the structure of an AFDX frame.

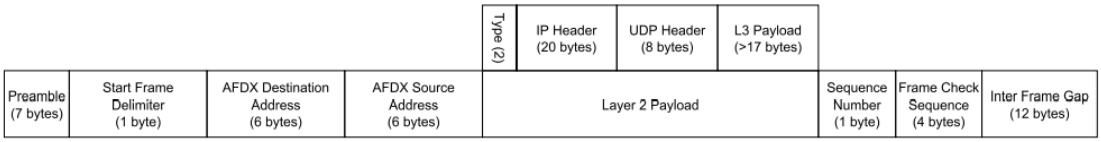


Figure 1.3: ARINC 664 frame structure

The end-to-end communication of a message using AFDX requires a static configuration of the source and destination end-systems and the AFDX switches.

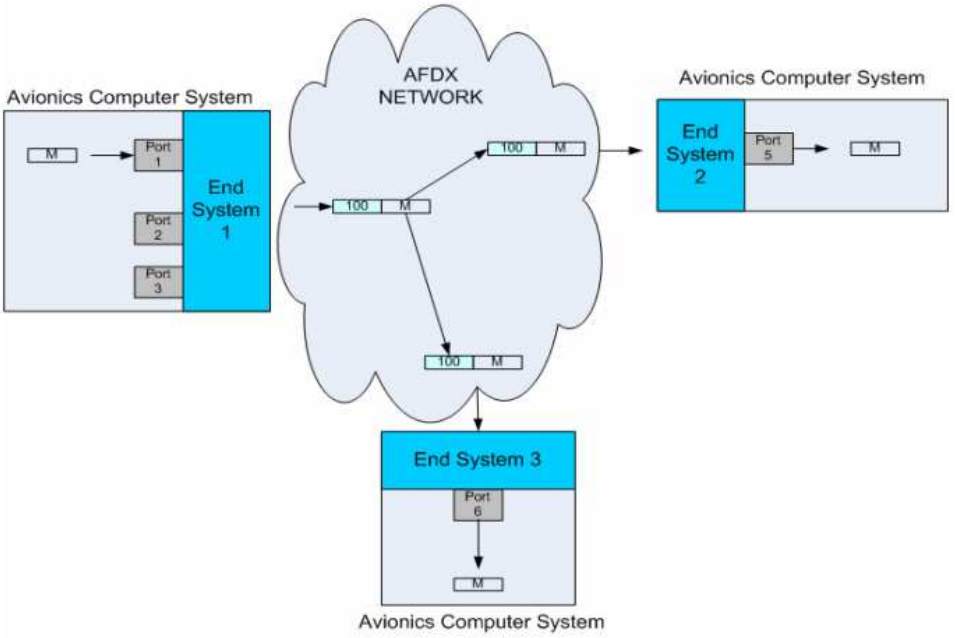


Figure 1.4: End-to-end message transmission in the AFDX [4]

Fig. 1.4 shows a message M being sent from Port 1 of End-system 1. The message is encapsulated in an AFDX frame and sent within the Virtual Link 100 (the Ethernet destination address specifies VLID 100). The forwarding tables within switches are configured to deliver the AFDX frame to both end-systems 2 and 3. The end-systems that receive the frame are configured to be able to determine the destination ports. In the case shown in Fig. 1.4, the message is delivered by end-systems 2 and 3 to ports 5 and 6, respectively. The source and destination port numbers are conveyed in the source and destination fields of the UDP header.

Redundancy Mechanisms ARINC 664 specifies static redundancy mechanisms, based on two dedicated networks (network A and B), i.e., a fully redundant network. Each subsystem

is connected to an end-system with two interfaces, connected to both AFDX networks. Each packet transmitted by an end-system is duplicated and sent on both networks. Therefore, if no transmission errors occur, each end-system will receive two copies of each packet. The first frame with a valid checksum to arrive to the end-system is consumed. Then, subsequent frames with a duplicate sequence number can be identified and discarded.

1.1.2 I/O networks: ARINC 429 and CAN

ARINC 429 [9] is one of the first standards specifically developed for avionics applications. It has been deployed in various avionics applications for decades, and is still used in actual aircraft as a low rate I/O data bus.

ARINC 429 implements serial line communication. A line is a unidirectional and simplex connection, connecting one sending station LRU (Line Replaceable Unit) and multiple receivers (up to 19), as shown in Fig. 1.5. A station may be attached to multiple buses and operates as either sender or receiver, thus hierarchical layouts are possible.

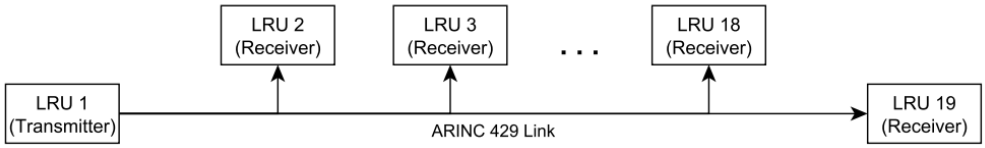


Figure 1.5: An ARINC 429 layout with just one transmitting LRU and up to 19 receivers [5]

ARINC 429 bus can operate with two transmission speeds: low or high speed. Low speed uses a variable clock rate with a throughput of 12-14 kbps, while the high speed mode requires a fixed clock rate and allows 100 kbps. The data unit transmitted in the medium is called a word, and has a length of 32 bits. Various types of data are specified for the ARING 429 communication, e.g., discrete and character data.

Controller Area Network (CAN) [11] is a broadcast digital data bus, designed in the 80s by bosh for automotive applications, to replace the complex wiring harness with a two-wire bus. It was standardized by International Standard Organization (ISO) to provide 1 Mbps and 125 Kbps data rate for cable lengths up to 40 and 500 meters, respectively.

CAN networks have been successfully used to replace point-to-point connections in many application domains, including avionics. This fact is due to its low cost, deterministic resolution of contentions, and error detection and retransmission mechanisms.

CAN supports two versions of protocols. The first uses the standard frame format with an 11-bits identifier, while the second uses an extended frame format with a 29-bits identifier. Since the bus is a shared medium, the connected controllers use the Carrier Sense Multiple Access with Collision Resolution (CSMA/CR) mechanism to avoid and solve collisions [13].

1.1.3 Requirements

The main avionics requirements concern both technical and costs aspects:

- **High Rate-** the number of embedded devices and functions is more and more important, which increases the amount of exchanged data. Hence, to cope with this growing expansion of the network, a high rate is required. Indeed, like Moore's law for processor power, the complexity of avionics systems doubles every 5 years, and to guarantee a long life for avionics systems (20 years on average), there is a need for a high rate to enable future development;

- **Predictability-** the network has to behave in a predictable manner and guarantees minimum and maximum delays for any type of traffic. Thus, the system must be able to deliver accurate information within a bounded time. Moreover, avionics systems are hard real-time systems where critical messages need to be transmitted on time, even in the presence of non-critical messages. Then, a quality of service management has to be provided;

- **Modularity-** This requirement is related to the flexibility and exchangeability of software and hardware components. An important step towards enhancing the avionics system modularity has been fulfilled with the adoption of the IMA approach [14], i.e., common elementary components can be configured to fit different avionic applications. This feature aims to minimize the (re) configuration effort to facilitate system maintenance and its progress over the years. In the specific case of the AFDX, the implementation of an event-triggered paradigm is favoring such a requirement;

- **Reliability and Availability-** The network must be fault tolerant and fulfill required reliability and availability levels to prevent failed nodes from affecting the normal operations. For instance, redundancy mechanisms are implemented for the AFDX network to recover packet losses and faulty nodes during operation time;

- **Physical and electromagnetic resistance-** avionics devices are subject to severe physical constraints such as vibrations, the large range of temperature degrees and electromagnetic interferences. Therefore, the network must be very resisting physically and particularly at the level of connectors and cables.

Furthermore, the choice of an avionics network has to be efficient to meet the design requirements for the least amount of money. Thus, the economic requirements are mainly:

- **Cost-** today, the communication network can reach 30% of the total cost of an aircraft, and this number will continue to grow. Thus, a good choice of avionics network is crucial to optimize the overall cost of the aircraft. The flexibility and configurability of components reduce development cycle duration, and ease incremental design and maintenance processes. Furthermore, the use of Commercial Off-The Shelf (COTS) technologies and components infers development and deployment costs reduction.

- **IEEE 802.3 compatibility-** to facilitate its adoption and its interoperability with the actual backbone network, i.e., AFDX;

1.2 Related Work: Improving Avionics Performance

Improving the performance of the Avionics Data Communication Network (ADCN) still is an ongoing work for already several decades for academia and industrial. There are several aspects related to such an objective, such as increasing the flexibility, resource efficiency, scalability and reliability, while reducing complexity, heterogeneity and costs related to cabling, fuel consumption and integration. We identify herein two main classes in this specific domain. The first class is based on cable-less solutions [14, 15] to reduce the weight and deployment costs. An interesting hybrid solution, denoted Wireless Safety-Critical Avionics Network (WSCAN), based on High Rate Ultra Wideband (HR-UWB) [16] and switched Ethernet technologies, to replace the backup network of the AFDX backbone has been proposed in [14]. The network is divided into two clusters of subsystems, each gathering the end-systems of the same avionics bay, interconnected through UWB technology. Then, the inter-cluster communication is enabled via gateways, connected to a Gigabit Ethernet switch. The choice of such a hybrid architecture is related to scalability issues. Moreover, the guarantees in terms of timeliness and reliability of such a solution have been proved under very specific conditions of isolation of each avionics bay from electromagnetic interferences. Hence, although the interesting advantages of using such a solution in terms of reducing weight and costs, and increasing flexibility and efficiency, the security still is a main challenge, due to its sensitivity to interference and jamming attacks. On the other hand, we have either solutions working on the actual ADCN architecture as [3] and considered as short term solutions, or defining new architecture solutions guaranteeing the avionics requirements as [17], which are considered as long term solutions. A design of a new CAN-AFDX RDC device to enhance the network bandwidth utilization while meeting the timing constraints has been proposed in [3]. Two major functions were adopted to reach this goal: i) the new gateway applies a frame packing on the upstream flows coming from the CAN buses, to reduce communication overheads, and consequently decrease the AFDX bandwidth utilization; ii) hierarchical Traffic Shaping (HTS) applied on downstream flows destined to actuators on CAN buses, to guarantee isolation between upstream and downstream flows on each I/O CAN bus, and consequently to favor the frame packing process. Although this solution improves the ressource efficiency, it does not address the key avionics requirements as the reduction of complexity, heterogeneity and cabling-related costs.

Furthermore, there is the Time-Triggered Ethernet (TTE) [17], which is based on Ethernet technology. The latter is considered as one of the most mature and cost effective technologies, allowing scalable and arbitrary topologies and supporting high speed communication and Quality of Service (QoS) features. It has been used for decades in various application domains and proved to be a robust and flexible technology. This fact will facilitate the interoperability with the actual backbone network, i.e., AFDX, and its adoption process in the near future. Moreover, the high communication speed of Ethernet technology, i.e., 1Gbps, will favor the transmission of mixed criticality-data on the same physical links; thus decrease the wiring and installation costs of the global architecture. Hence, TTE [17] is an embedded safety-critical Switched Ethernet network based on time-triggered communications [18]. It uses time scheduling, i.e., TDMA, with an off-line configuration to guarantee predictable and

deterministic communications for the highest priority traffic, and it combines different types of data flows on the same network to reduce the heterogeneity of the actual ADCN. Although TTE reduces heterogeneity and guarantees deterministic communications, it follows a completely different communication paradigm from the actual AFDX network, which is a distributed network based on event-triggered communications. This characteristic will decrease the modularity level due to the need of synchronization, and increase the reconfiguration effort.

Unlike TTE, our solution is based on Ethernet with a ring topology. The recent research efforts towards defining new communication solutions for cyber-physical systems (CPS) to guarantee high availability level, while limiting cabling costs, have actually renewed the interest in ring-based networks. Therefore, introducing Ethernet-compliant solutions supporting ring topology for avionics has become feasible, but also advisable for the following reasons:

- the ring topology will decrease the cabling complexity, in comparison to the switched one, thus an inherent weight reduction and an increase of system efficiency, e.g., less fuel consumption;
- the high availability level offered by the ring topology due to the various redundancy solutions, which have been specified in the documents IEC62439-1/7. This topology provides actually an implicit redundant path by introducing only one additional connection between the two end nodes, compared to line or star topologies [19].

However, the main challenge for Ethernet-compliant solutions supporting a ring topology is reconciling the different avionics requirements, and especially predictability and availability, while reducing the reconfiguration effort and deployment costs. To achieve this aim, we have followed a specific design methodology, detailed in the next section.

1.3 Methodology and Outline

In this section, we detail our methodology to design and validate a ring-based Ethernet solution. We have followed a 'Top-Down' approach, which starts from high-level specifications of the target avionics solution to gradually converge to the design and validation.

- **Evaluation of the existing ring-based Ethernet solutions:** Before specifying a customized solution to fulfill the avionics requirements, we have started with a deep analysis of existing solutions, which may cope with this need. Therefore, we have conducted a qualitative and quantitative benchmarking of the most relevant Real-Time Ethernet (RTE) solutions supporting ring topology vs the main avionics requirements. The analysis of the key Performance Indicators presented in the standard IEC 61784 [1] has revealed the non-existence of a perfect solution meeting all the requirements. However, this step has allowed us to identify the most efficient mechanisms and infer a high specification level of our solution. This step is detailed in **Chapter 2**.
- **Specification of a new RTE network for avionics:** The main idea is to bridge the gap between the existing RTE solutions to satisfy the avionics constraints. This fact mainly consists in guaranteeing high reliability, availability and timing performance levels,

while keeping the IEEE802.3 compatibility and reducing the costs and configuration efforts. Hence, the proposed solution must integrate various aspects:

  - ***The network architecture:*** replacing the current avionics topology with a ring topology raises some questions concerning the nodes grouping to guarantee a physical isolation between the functions of different criticality levels;
  - ***The MAC protocol:*** the choice of the MAC protocol will affect directly the determinism of communications, a key requirement for avionics systems. Therefore, the MAC protocol must be well defined to ensure this requirement while keeping a low reconfiguration effort;
  - ***The availability mechanisms:*** the new network must fulfill the required availability level. Hence, we have to address such an issue through defining adequate mechanisms to properly meet this requirement.

  This step is presented in **Chapter 3**.

- **Performance and Dependability Analyses:** For avionics embedded systems, it is essential that the communication network meets the certification requirements where real-time constraints and reliability level must be guaranteed. Hence, we need to investigate the timing performance and dependability of our solution using adequate methods covering the worst-case behaviour.

  - To conduct the timing performance, we have selected the Network Calculus framework [20]. The high modularity and scalability of such a framework make it particularly efficient to conduct timing analysis of complex communication networks [21], e.g., it has been recently used to certify the AFDX [8]. Such a key issue is detailed in **Chapter 4**.
  - To infer the dependability analysis of our proposal, we have used Stochastic Active Networks (SANs), a stochastic extension of Petri Nets (PNs). This analysis has to take into account several aspects, such as the network size, the equipment reliability (MTTF) and the mission time. This step is detailed in **Chapter 5**.

- **Validation:** To have the proof of concept of our proposal, we need to validate its timing performance and dependability through a realistic avionics case study. To achieve this aim, we have considered a representative avionics network of an A380, and conducted comparative analyses with the current AFDX network and the most relevant RTE solutions. This analysis is detailed in **Chapter 6**.

1.4 Contributions

The main contribution of this thesis are as follows:

- **Design and specification of AeroRing**. The main innovative features of AeroRing are: (i) distributed access mechanism allowing simultaneous data exchange, to increase

the offered bandwidth and resource usage efficiency; (ii) distributed fault management mechanism avoiding any central point of failure, to provide high reliability and availability levels; (iii) event-triggered communication enhancing the system flexibility and decreasing the implementation complexity, through avoiding any need of synchronization; (iv) QoS management handling heterogeneous data constraints, through QoS-aware routing algorithm;

- **Timing analysis of AeroRing**. To deal with the performance evaluation of AeroRing network, accurate timing analysis to compute worst-case delays or at least upper bounds has to be considered. For the most common ring-based Real-Time Ethernet (RTE) profiles, conducting such performance analyses has been greatly simplified due to their implemented time-triggered communication scheme, e.g., Master/slave or TDMA. Unlike these existing approaches, AeroRing is an event-triggered ring-based networks, which guarantees high resource utilization efficiency and (re)configuration flexibility, at the cost of increasing timing analysis complexity due to cyclic dependencies, i.e., there exist interfering flows with paths forming cycles;

  To cope with this arising issue of cyclic dependencies, only few techniques have been proposed in the literature, mainly based on Network Calculus framework [20], which defines an arrival curve for each input flow and a service curve for each crossed node (a background on the Network Calculus is given in Appendix B). Existing approaches are based on *iterative local analysis*, by successively computing the delay bound in each crossed node either directly, i.e., *Delay-based* methods [22][23] [24], or from the backlog bound, i.e., *Backlog-based* methods [25] [20]; and summing these delays up results in end-to-end delay bounds. However, these lead to overly pessimistic upper bounds, decreasing the network scalability and resource efficiency, as it will be illustrated in Chapter 4. To enable the computation of tighter end-to-end delay bounds, we have introduced a new global analysis approach, Pay Multiplexing Only at Convergence points (PMOC). This consists in considering the flow serialization phenomena by paying the bursts of interfering flows only at the convergence points. Hence, we have defined and proved the guaranteed end-to-end service curve of any f.o.i crossing such a network. Then, the methodology to compute delay bounds have been presented for one ring and generalized to multiple-ring topologies. Finally, the first numerical results have highlighted the accuracy of our proposed approach, in comparison to conventional methods, which yields enhanced performance, in terms of resource efficiency and network scalability;

- **Dependability Analysis of AeroRing**. To analyze the reliability level of AeroRing, we have conducted a dependability study, where we analytically quantify the reliability level of AeroRing depending on several aspects. First, we modeled our system using the Stochastic Active Networks and solved it using the Mobius tools. Then, a sensitivity analysis has been conducted to highlight the high reliability level of AeroRing;

- **Validation of AeroRing**. Finally, we validate our proposed approach, by investigating the

offered timing performance and reliability level through a realistic avionics case study of an A380. The latter consists of 8 AFDX switches connecting 54 end-systems, where each switch is connecting between 6 and 13 endsystems. Each end-system sends 8 flows within 3 different traffic classes. AeroRing has been compared to the AFDX network and conventional RTE solutions under the same scenario, to show its timeliness, scalability, resource efficiency and reliability.

Chapter 2

Ring-based Real-Time Ethernet Solutions: State of the Art

Contents

In this Chapter, we first give a general description of standard Ethernet, and particularly the Gigabit Ethernet, i.e. 1000BASE-T, and some experimental results of its performance in a hostile environment. Then, we describe the most relevant Real-Time Ethernet solutions and their pros and cons vs avionics requirements. Finally, we detail some quantitative benchmarking of the main performance indicators, which helps us to have some insights into an ultimate solution for new generation avionics.

2.1 Standard Ethernet

In May 1973, Robert Metcalfe and his colleagues at the Xerox Palo Alto Research Center wrote a memo describing a local area network (LAN) technology to interconnect stations, servers and peripheral devices within the same building using a common bus system [26]. This fact led to the creation of the Ethernet media access control protocol, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [6], inspired from the Aloha protocol [27].

The CSMA/CD works following two steps. First, the station has to listen continuously to the medium. If the station has data to transmit, then it has to wait until the medium is released to send the data. The second step consists in detecting collisions while transmitting the data. If it is the case, the node stops the transmission and waits for a random time before a new attempt.

During the last decades, Ethernet has been successfully used in various applications domains. However, it has not been conceived to support real-time applications because of its unpredictable CSMA/CD protocol, which can lead to collisions and an unbounded medium access delay [28], e.g., it can even lead to a complete drop of the frame, if the maximum number of attempts is reached, i.e. 16 attempts. Several works have been conducted to ensure a real-time behaviour on top of the standard Ethernet, by providing solutions to the collision problem due to the CSMA/CD mechanism. Among the existing approaches, there are on the first hand, methods based on the modification of the MAC layer, such as the CSMA/DCR [29], the Virtual Time CSMA [30] or Window Protocol [31], and on the other hand, those adding a transmission control layer using several techniques at the data link or at the application layers.

Afterwards, in the 1990s, Switched Ethernet with full-duplex links was introduced to solve the collision and shared medium issues. Each device is connected to a switch port, which can send and receive data simultaneously due to the full-duplex links. However, these mechanisms do not guarantee the determinism requirement, where the output multiplexing queues may introduce unknown (non deterministic) delays.

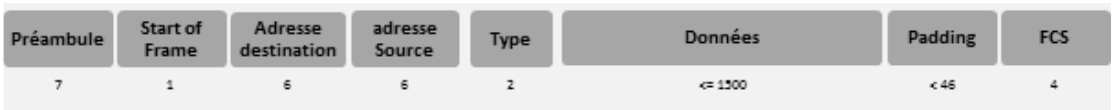


Figure 2.1: Ethernet frame structure

As shown in Fig. 2.1, an Ethernet frame starts with a preamble field to ensure synchronization, followed by Start Of Frame field (SOF) indicating the start of the frame. Then comes the 6-bytes destination and 6-bytes source addresses to identify the receiver and transmitter,

respectively. The type of encapsulated protocol is specified by the Type field. The data is contained in the payload field with a length varying from 0 to 1500 bytes. To detect collisions, there must be a minimum length of 64 bytes and padding is used for short frames to reach this minimum length. Finally, the Ethernet frame ends with a FCS (Frame Check Sequence) for error detection.

2.1.1 Gigabit Ethernet

The Gigabit (also named 1000BASE-T) generation of Ethernet networks, i.e 1Gbps, is the successor of the Fast Ethernet, i.e. 100Mbps. His appearance is due to the growing number of users and the strong demand in terms of bandwidth. In 1997, the working group 802.3ab was created for the standardization of the Gigabit Ethernet, based on a four twisted pair non-shielded cable of category 5 [7, 6], respecting the following objectives [6]:

1. Support the CSMA/CD MAC;
2. Comply with the specifications for the Gigabit Media Independent Interface (GMII);
3. Support the 1000 Mb/s repeater;
4. Provide line transmission that supports full and half duplex operations;
5. Meet or exceed FCC (Federal Communications Commission) and Class A/CISPR (International Special Committee on Radio Interference) or better operation;
6. Support operation over 100 meters of copper balanced cabling;
7. Bit Error Ratio less than or equal to 10^{-10};
8. Support Auto-Negotiation.

Fig. 2.2 shows the relationship between the 1000BASE-T PHY layer, the Open Systems Interconnection (OSI) Reference Model and the IEEE 802.3 CSMA/CD model. In order to meet the requirements, 1000BASE-T uses a full duplex baseband transmission on four twisted pair cable of category 5. This requires a data rate of 250 Mbps in each direction of a pair of cables, as illustrated in Fig 2.3. The main features of the 1000BASE-T PHY layer, as well as the different functions of its sublayers are described in Appendix A

The data are transformed into symbols by using the 4D-PAM5[1] method before being sent. Thanks to the PAM5 modulation, this transformation allows an optimal use of the bandwidth. Each symbol represents one of the five levels of modulation (-2, -1, 0, + 1, + 2). Four levels are used for the transmission of data to encode two bits of data (two bits have four different values) and the fifth is used for control, error correction (FEC) and improving performance. Thus, we obtain a rate of 125 M symbol/s per pair, which corresponds to a transmission of eight bits all at once.

The combination of four symbols gives 625 (5^4) possible cases. However, to transmit eight bits of data, we need only 256 (2^8) combinations. The 369 remaining codes are used for control

[1] 4D-PAM5 = Four-Dimensional 5-Level Amplitude Modulation

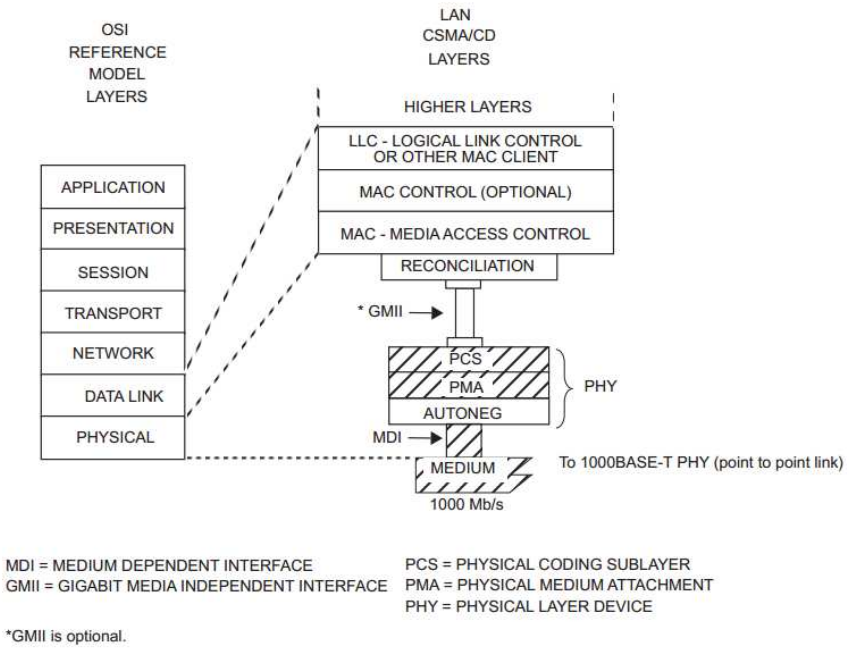


Figure 2.2: Type 1000BASE-T PHY relationship to the ISO Open Systems Interconnection (OSI) Reference Model and the IEEE 802.3 CSMA/CD LAN Model [6]

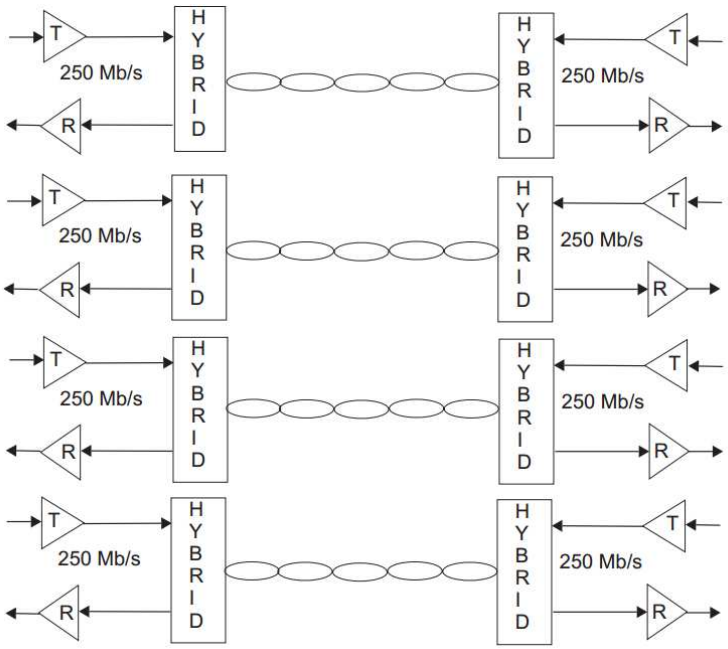


Figure 2.3: 1000BASE-T Transmissions [6]

and improving the performance (256 codes for a 100% redundancy and 113 codes for control) using a trellis coded modulation at the transmission, and a Viterbi decoder at the reception. The Viterbi decoder allows not only to detect errors, but also to correct them. The use of this type of encoder/decoder allows having an effective gain of 6dB.

The 1000BASE-T uses a continuous signaling system. Even between the transmission of two frames or at the lack of transmission, it continues to transmit a set of control symbols to improve the synchronization. Fig. 2.4 shows an example of a multilevel signaling in base-band.

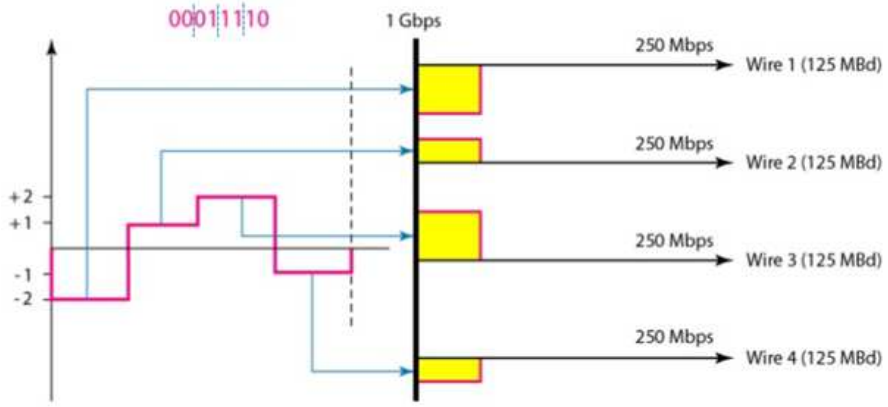


Figure 2.4: Baseband Multi-level Signaling

2.1.2 Experimental Results Under Electromagnetic Interference

Standard Ethernet has shown its effectiveness in non-critical domains, which have a bit error rate lower or equal to 10^{-10}. However, in avionics, networks are exposed to high interference that can degrade or prevent the normal functioning of the networks and increase the bit error rate.

In order to measure the robustness and performance of the Gigabit Ethernet technology in such environments, we have conducted some electromagnetic interference (EMI) tests on a simple system setup. This system consists of two PCs connected by 1000BASE-T. The idea is to expose the medium to different degrees and types of interference and measure the resistance of the 1000BASE-T.

We have used three types of cables for tests: an unshielded twisted pair cable of category 5, a shielded twisted pair cable of category 6 and a double AFDX-cable, i.e., we have used two AFDX cables to form the four pairs of a Gigabit Ethernet cable.

Afterwards, to measure the loss rate, we have developed a client/server traffic generator that generates, sends and receives frames and provides statistics on traffic (speed, number of frames received or lost and their sequences). Interferences are generated in two types of environments: a cylindric pipe that canalizes the interferences, or an anechoic room containing a powerful antenna. We have conducted an increasing frequency scanning from 30

Mhz to 150 Mhz with three different powers, where each frequency is generated within two phases: first the non-modulated frequency is sent for a certain time, then 80% of it modulated in a sinusoidal signal with a frequency of 1 kHz.

We have obtained the following results:

- In the anechoic room: the antenna directs the signal to a specific point on the cable. The tests were done on the AFDX cable and the STP cat 6 cable with a power of 3, 10 and 20 V/m. The generated interferences on both cables have not altered the network behaviour and no loss has been detected regardless of the frequency and power.
- In the cylindric pipe: the tests were done on the AFDX and the UTP cat 5 cables with a power of 20 V/m. The generated interference on the AFDX cable did not alter the network behaviour and no loss has been detected. However, in the case of UTP cat 5, the first losses were detected at a frequency of 70 Mhz. The losses have increased when increasing the frequency, until a complete loss of connection with frequencies around 100 Mhz. Afterwards, losses have started to decrease until disappearance with frequencies greater than 120 Mhz.

Fig. 2.5 shows the error ratio according to the number of a sliding window, where each window contains 100000 packets.

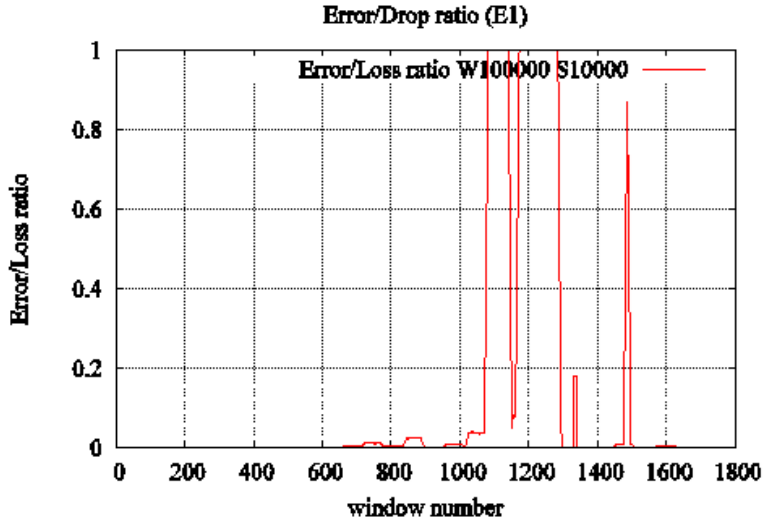


Figure 2.5: Average error ratio according to the window number

As we can see, the losses are observed for a window number between 625 and 1600, and they increase until a total loss of communications, i.e. 100% of error.

These results can be explained due to the code groups used in the Gigabit Ethernet to control the flow of communication and correct errors. If the PHY can not correct the errors and detects that the quality of the communication flow is degraded, then the 1000BASE-T switches from 1000 Mbps mode to a lower mode (100 Mbps or 10 Mbps) to better control

the communication and generates more redundancies to correct the data. The high error rate, i.e., equals or near to 1 in Fig. 2.5, corresponds to a loss of connection, where both PHYs perform the auto-negotiation mechanism to negotiate a transmission mode able to resist to the interferences, i.e. reduces the data rate and increases the redundancy. Hence, the disappearance of losses can have two explanations: 1) the two PHYs were able to negotiate an operating mode able to resist to the interferences; 2) the generated interferences were decorrelated from the transmitted signal, i.e., did not affect the signal.

2.2 Ring-based RTE Solutions vs Avionics requirements

During the last two decades, a wide range of RTE solutions have been proposed by industrials and academia to fulfill the requirements of real-time applications. The most relevant ones have been cited in [1, 32, 33, 34].

In this section, we first identify the main parameters impacting the RTE solutions performance and reliability. Then, we introduce a new classification of the existing solutions from an avionics perspective. Finally, we detail a qualitative benchmarking of the most interesting approaches through highlighting the pros and cons of each one of them.

2.2.1 Taxonomy

To identify the most relevant RTE solutions from an avionics perspective, we distinguish herein two main characteristics: the communication paradigm and the redundancy protocols.

2.2.1.1 Communication Paradigm

This parameter is of utmost importance to quantify the reconfiguration effort needed by the alternative RTE solution, in comparison to the AFDX-based one. This indicator conditions the modularity level offered by the selected solution, a key requirement in avionics. We consider the two main paradigms [18], i.e., event-triggered and time-triggered. The event-triggered paradigm is known as highly flexible and facilitates the system reconfiguration, but it infers at the same time an indeterminism level and needs further proofs to verify the predictability requirement. On the other hand, the time-triggered paradigm is highly predictable, but presents some limitations in terms of system reconfigurability.

2.2.1.2 Redundancy Protocols

This parameter impacts especially the availability level of the communication network, but also the deployment costs to support the introduced redundancy level. We mainly identify two classes of redundancy solutions, static and dynamic. The former is generally based on a fully duplicated network, where both are used in parallel to increase the fault detection coverage. This solution offers a zero switchover time when a failure occurs, through guaranteeing two redundant paths for each transmitted data. This fact infers a high availability level, but also high deployment costs. On the other hand, the dynamic redundancy solutions have been introduced to decrease the installation costs, through using a backup path in case of failures, but they need to offer a bounded switchover time to guarantee availability.

Various redundancy mechanisms for ring-based RTE solutions have been proposed and

cited in IEC62439-1/7. The most relevant static protocols are the Parallel Redundancy Protocol (PRP) [35] and High-availability Seamless Redundancy protocol (HSR) [35]; whereas, the main dynamic protocols are Distributed Redundancy Protocol (DRP) [36], Media Redundancy Protocol (MRP) [37] and Ring-based Redundancy Protocol (RRP) [38].

Both **PRP** and **HSR** offer a zero switchover time when failure, through guaranteeing two redundant paths for each transmitted data. The PRP handles this feature due to a fully redundant network, i.e., two parallel networks, where most of the equipment are attached to both parallel networks, and each data is duplicated at the transmission and filtered at the reception using a specific device; whereas the HSR protocol achieves the same purpose through a daisy-chain ring topology and sending duplicated data on both directions, then the destination consumes only the first valid one. The unicast messages are filtered by the destination and the broadcast messages by the source using the MAC addresses to avoid infinite message looping.

The **MRP** is based on a manager, called Media Redundancy Manager (MRM), that monitors the status of the network and the other nodes, called Media Redundancy Clients (MRCs). Each equipment integrates an internal switch with two ports, and supports three status: *disabled*, when the port is down; *blocked*, the forwarding function is disabled; *forwarding*, the port can receive and forward messages. In the nominal case, the ring is closed and all MRCs are forwarding the data, except the MRM which blocks one of its ports to create a logical line topology and to avoid the infinite message looping. Furthermore, the MRM monitors the status of the network by sending periodically *Test* frames on both ports, and if the frames are received on the opposite ports, then the ring is closed. However, if the frames are lost, then the MRM concludes that the network is faulty. In addition to that, each MRC monitors the local connection with its neighbours, if it detects a failure or a recovery, then it announces it to the MRM by a *LinkChange* frame. In both scenarios, the MRM activates both ports to transmit data and informs the MRCs about the topology change by sending *TopoChange* frames.

The **DRP** implements a local fault detection mechanisms, where each equipment can check the status of its neighbors by sending a link test frame *"LinkCheck"* to detect failures. It transforms the ring topology into a line topology by disabling a port of an elected device (the device with the highest ID is elected) to avoid infinite packet looping. In addition to the local fault detection, DRP implements a centralized fault detection mechanism to check the ring status in a cyclic manner, i.e., during each cycle, only one equipment can check the ring status via a ring test frame *"RingCheck"*, gather and broadcast the information to the rest of equipment in case of a change. When a failure occurs, the device with the blocking port activates its port to allow packets transmission. It is worth noting that, an accurate synchronization protocol is required to manage such a cyclic process.

The **RRP** manages the fault detection and network configuration dynamically. Based on the physical layer mechanisms as specified in ISO/IEC 8802-3:20000 Clause 24, the devices can detect their neighbours and also occurred faults, then they share these informations through the network. Based on these informations, all network devices will build their routing tables in a distributed way. However, RRP transforms the ring topology into a line topology to avoid infinite packet looping, through the selection of two adjacent devices, called Ring Network

Managers (RNMs), and disabling one of their ports.

Protocol	Costs	Flexibility	Resource Efficiency	Reliability
HSR	Low	High	Low	High
PRP	High	High	Low	High
MRP	Low	Low	Low	Medium
DRP	Low	Low	Medium	Medium
RRP	Low	High	Medium	Medium

Table 2.1: Benchmarking of redundancy protocols supporting ring topology

Characteristic	PRP	HSR	MRP	DRP	RRP
Topology	Redundant Network	Ring	Line	Line	Line
Frame Redundancy	Yes	Yes	None	None	None
Fault detection	N.A.	N.A.	Local + Global	Local + Global	Local
Reconfiguration	N.A.	N.A.	Centralized	Centralized	Distributed
Frame Filtering	Using a specific device	Within the nodes based on MAC @	Manager	Elected node	2 ring network managers

Table 2.2: Main characteristics of ring-based redundancy protocols

Discussion The characteristics summary of the main ring-based redundancy protocols redundancy protocols, e.g., PRP, HSR, MRP, DRP and RRP, described above is illustrated in Table 2.2, and their benchmarking vs the main avionics requirements is illustrated in Table 2.1.

Both static protocols, PRP and HSR offer a high flexibility and reliability levels, since they do not require any (re)configuration mechanism and offer a zero switchover time when a failure occurs. However, both protocols limit the resource efficiency, since the available utilization capacity is only of 50% due to the duplication of all packets on both networks (resp. ring directions) for PRP (resp. HSR). However, HSR was designed to reach the same reliability level as PRP using redundancy within the same network, instead of using fully redundant one, which allows reducing the deployment costs.

On the other hand, the dynamic protocols allow reducing the deployment costs through enabling redundancy within the same network. However, they degrade the reliability level by transforming the ring into a line topology to avoid infinite message looping. Moreover, RRP offers the best flexibility level due to its dynamic mechanisms to detect faults and configure the network, in comparison with DRP and MRP. Finally, the MRP guarantees the lowest resource efficiency level, in comparison with DRP and RRP, due to its high overhead to detect faults, i.e., local and global mechanisms.

We can notice that each protocol satisfies some requirements better than others, but there is no best protocol in terms of all the requirements. Hence, the new solution needs to bridge the gap between these aforementioned protocols, through guaranteeing a high reliability level as HSR and high flexibility level as RRP, while increasing the resource efficiency by taking more advantage of the multiple path and reducing the data control overhead.

2.2.2 Classification

In [32], an interesting classification of the main RTE solutions has been detailed based on the implementation level of each proposed solution, as represented in Fig. 4.1. Hence, a first class with an implementation on top of the transport/network layer has been identified which does not require a special hardware, e.g., P-NET [39], V-NET [40, 41] and Modbus-RTPS [42, 43, 44]. These solutions are usually easier to implement and configure, but they lead at the same time to important latencies (about 10ms), which make them more effective for soft real-time applications. Then, a second category has been defined, which provides a realization on top of the MAC layer while keeping the IEEE802.3 compatibility, e.g., TCNET [45], Ethernet/IP [46, 47] with Device Level Ring (DLR) [48, 49] and PowerLink [50]. Finally, the third category of solutions modifies the standard implementation, by introducing a set of hardware modifications, to guarantee a better real-time behaviour.

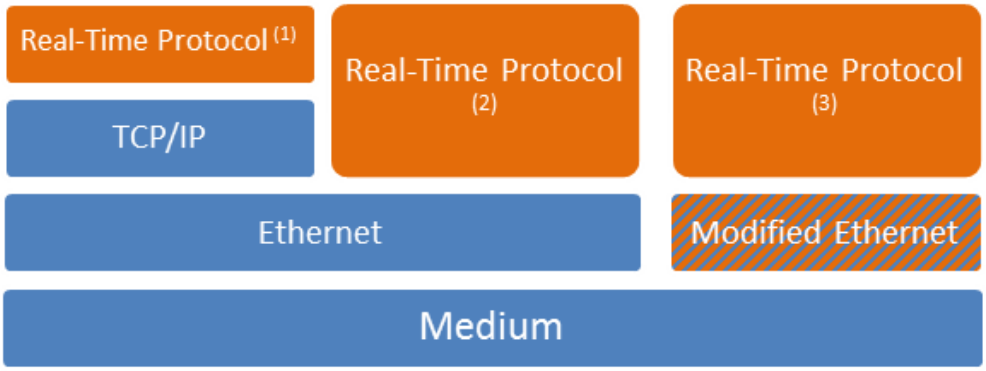


Figure 2.6: Different implementation levels of Real-Time Ethernet

Most of these solutions have been standardized and cited in the second part of IEC 61784 standard [1], entitled "Additional fieldbus for real-time networks based on ISO/IEC 8802-3", published by the working group SC65C/WG11. This working group has been established to refine a classification of needs for a real-time Ethernet and to define profiles of communication networks, based on the standard IEEE 802.3.

A different classification is introduced herein to distinguish the main RTE solutions from an avionics perspective, as shown in Fig. 2.7. Four classes of RTE solutions supporting ring topology have been identified:

- Event-triggered with static redundancy: this class represents the current avionics network based on the AFDX standard, which implements an event-triggered paradigm and a fully duplicated network. This solution reduces the reconfiguration effort, while increasing the deployment costs. Hence, it is considered as a reference for the benchmarking of the most relevant RTE solutions. It is worth noting that the current avionics network has been proved as predictable [51] and guarantees a high availability level thanks to its static redundancy solution, similar to the PRP [35];

- Time-triggered with static redundancy: a representative solution in this class is the Time Triggered Ethernet (TTE) [17], which implements a time-triggered paradigm and a static

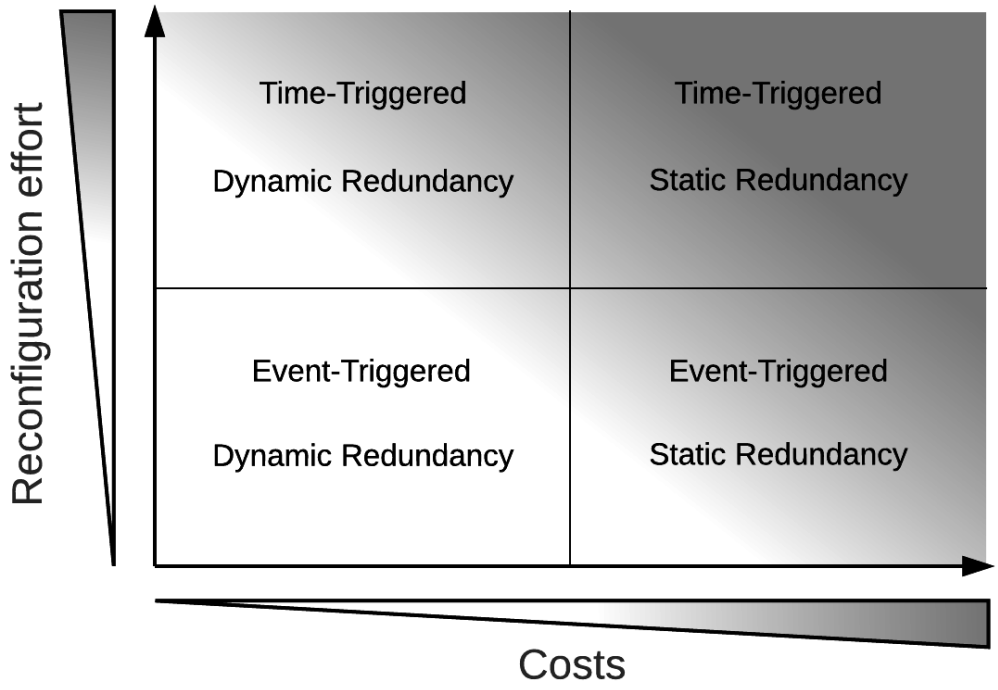


Figure 2.7: Classification of RTE solutions based on Communication paradigm and Redundancy mechanisms

redundancy solution. This solution offers a high predictability and availability levels, but it increases at the same time the deployment costs and the reconfiguration effort. Therefore, this solution will not be detailed in this chapter;

- Time-triggered with dynamic redundancy: two interesting solutions can be identified in this class, EtherCAT [52, 53] and Profinet/IRT [54]. These RTE solutions implement actually a master/slave mechanism based on a time-triggered paradigm, and dynamic redundancy solutions, such as the Media Redundancy Protocol (MRP) [37] for Profinet/IRT. This class of solutions will definitely decrease the deployment costs thanks to the standby mode on a ring topology, but will increase at the same time the reconfiguration effort;

- Event-triggered with dynamic redundancy: the interesting candidate in this class is Ethernet/IP with DLR [48]. The solution implements event-triggered paradigm, which induces a similar reconfiguration effort than the AFDX solution, while implementing a dynamic redundancy solution to reduce the deployment costs. From a practical point of view, this class should actually contain the best solution for the new generation avionics in terms of modularity and costs, but it is also the one introducing the most challenging issues to guarantee predictability and availability.

Therefore, we detail herein a qualitative benchmarking of the most relevant classes of RTE solutions in the avionics context: time-triggered with dynamic redundancy and event-

triggered with dynamic redundancy. A quantitative analysis of their performances will be conducted in the next section.

2.2.3 Time-Triggered Solutions with Dynamic Redundancy

2.2.3.1 EtherCAT

EtherCAT has been defined by Beckhoff GmbH and supported by the EtherCAT Technology Group (ETG). It implements a master/ slave mechanism on top of Fast Ethernet (100Mbps). The main particularity of EtherCAT is the on-the-fly forwarding technique, which allows the slaves to insert the requested data in a standard Ethernet frame crossing the couplers step by step. As shown in the first topology of Fig. 2.9, EtherCAT is a line network. Thanks to the Full-Duplex links, frames are sent by the master until the last slave, which sends back the frame to the master in the opposite direction to form a virtual ring on a line topology. It is worth noting that this technique requires a specific implementation within the slaves, but allows at the same time collecting data from several slaves to be transmitted within the same frame. Therefore, this technique allows reducing the overhead of EtherCAT to one header for many collected data, instead of one header per data in classic Ethernet (see Fig. 2.8).

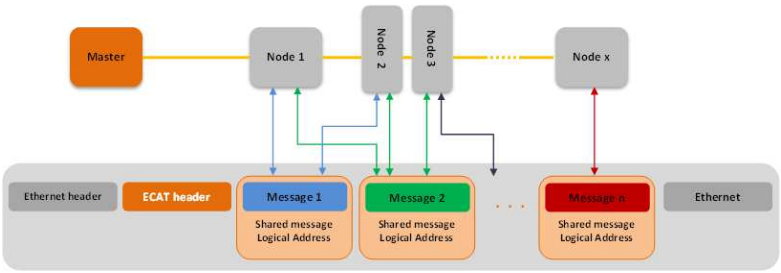


Figure 2.8: EtherCAT frame structure and processing

EtherCAT ensures a great flexibility by adding and reconfiguring the nodes on-the-fly, thanks to the "hot connect" mechanism. Furthermore, to guarantee the reliability requirements, EtherCAT supports the master redundancy due to the hot standby method, and implements a dynamic redundancy solution based on a ring topology. In the case of a link or node failure, first, the slave detecting the failure returns immediately the EtherCAT frame to the master to avoid losing the communication with the rest of the nodes. Afterwards, the master activates its ports and sends the frame on both to be received by all slaves. Furthermore, the master can determine the failure location through analyzing the slaves error counters.

EtherCAT provides interesting timing performance and availability levels due to the on-the-fly mechanism. The latter induces actually short communication latencies, thus a fast failure detection. Furthermore, it implements a specific redundancy mechanism to enhance the reliability level. However, the main drawbacks of this technology are mainly related to: (i) the specificity of its devices, which increases the implementation costs; (ii) the use of a master/slave mechanism, which reduces its compatibility with the AFDX standard and increases its reconfiguration effort.

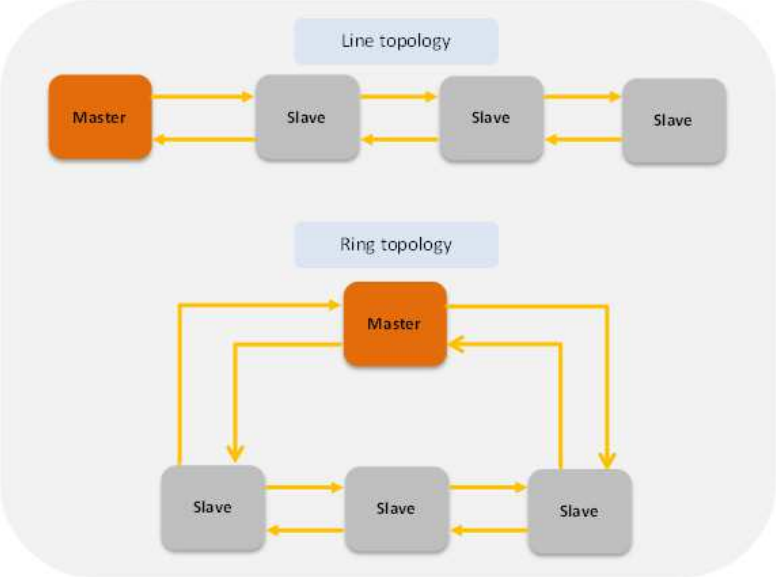


Figure 2.9: EtherCAT network topologies

2.2.3.2 PROFINET IRT

PROFINET/IRT (*Isochronous Real-Time*) is an extended version of PROFINET, which supports real time communications on top of Fast Ethernet (100Mbps). It is a master/ slave network, based on cyclic communication handling two communication channel types: isochronous and asynchronous. These channels are used by slaves to transmit real-time and non real-time data, respectively. The data is relayed using the Cut-through mechanism to reduce the processing time. The first communication channel, i.e. isochronous, is called time scheduled communication, where all packets transmissions are scheduled during the initialization phase. It is worth noting that the isochronous channel requires an accurate synchronization protocol to guarantee the packet transmissions according to a predefined schedule. The second channel is called, SRT channel (Soft Real-Time channel), used to satisfy the real-time automation constraints. It is based directly on Ethernet (Layer 2) which reduces the processing time within the communication stack. The third communication channel is reserved to non-real-time TCP/UDP/IP packets without temporal constraints. Furthermore, in order to reduce the cycle time, PROFINET/IRT implements a slipstream method to transmit data, which consists in sending the packets following the physical order of the nodes from the master point of view: the first packet is for the farthest node and the last packet is for the nearest node. This method decreases inherently the communication latencies. Fig. 2.10 shows a temporal diagram of a PROFINET communication with the slipstream effect. The cycle time is the time between the transmission of the first bit of the first packet by the master until the reception of the last bit of the last packet by the first node. Furthermore, Profinet/IRT supports reliability features through implementing the MRP [37], based on a ring topology.

Profinet/IRT favors predictability and availability requirements thanks to the cut through mechanism and the slipstream method, which infer short communication latencies and fault

detection time. Moreover, it implements the MRP to manage redundancy and enhance reliability. However, it has mainly the same drawbacks than EtherCAT in terms of reconfiguration effort, because of the synchronization protocol and the master/slave paradigm. Nevertheless, Profinet/IRT should be more interesting than EtherCAT in terms of deployment costs, since it does not require specific devices.

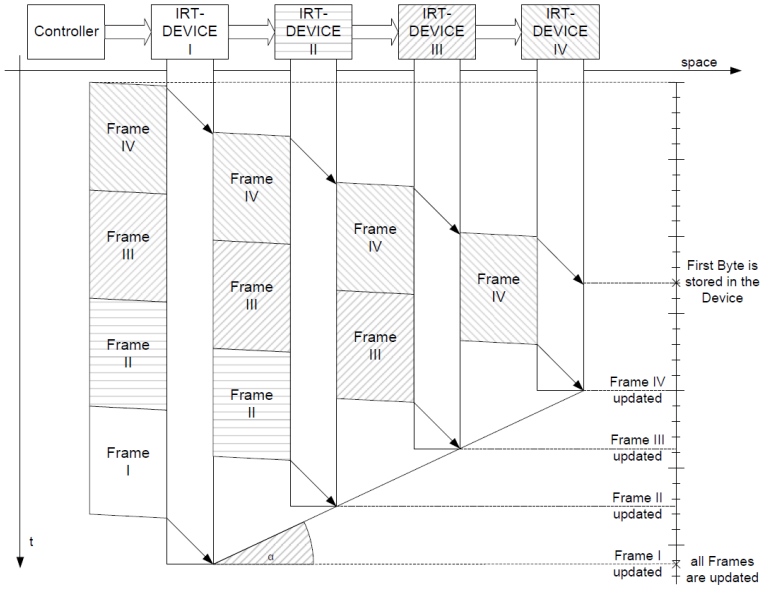


Figure 2.10: Temporal diagram of a PROFINET communication with the slipstream effect

2.2.3.3 SERCOS III

SERCOS III [55] is an extension of SERCOS (SEriell Real time Communication System Interface) on top of Ethernet, a real-time communication protocol which has similar characteristics than EtherCAT, i.e. frame summation, access mechanism (master/slave) and the network topology. However, the number of slaves cannot exceed 254.

SERCOS III communications are based on cycles, where each cycle consists of two logical communication channels: the RT channel for real-time communications and IP channel for non-real time communications. As shown in Fig. 2.11, each cycle is initiated by the master and consists in sending up to four Master Data Telegrams (MDT) and four Acknowledge Telegrams (AT) during the RT channel, and then the IP channel where slaves can send their non-real-time data in standard frames. The number and duration of MDTS and ATs are configured during the initialization phase.

The master sends to the slaves MDT frames to convey the synchronization and control information, and AT frames which are empty frames with predefined fields to allow slaves inserting their data and status information at their allocated fields. Afterwards, the slaves can send these telegrams to the master and/or to the other slaves. The reception and writing of data are done "on the fly" when the frame passes through the slave; whereas, the non-real-time data are sent in the IP channel in standard frames, respecting the maximum duration of the

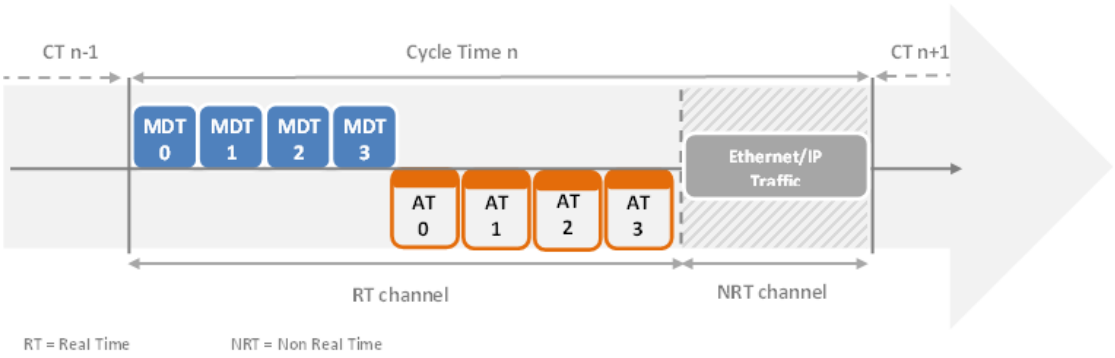


Figure 2.11: Communication cycle of SERCOS III

channel. It is worth noting that a specific software or hardware (FPGA) component needs to be integrated at each device (master or slave) to separate both communication channels.

As EtherCAT, the main drawbacks of this technology are due to the specificity of the device, and the master/slave mechanisms, which increase the implementation costs and the reconfiguration efforts. Moreover, the master presents a central point of failure, which decreases the reliability level. Finally, it seems more suitable for small data sizes, since the cycle is limited to four MDT and four AT frames.

2.2.3.4 VABS

VABS (Very High Performance Automation Bus System) is an academic RTE solution proposed in 2013 by [56]. Unlike many RTE protocol using the standard Ethernet frame to convey their data, VABS has its own frame format and its own data link layer protocol, as shown in Fig 2.12. This choice limits VABS compatibility with standard Ethernet devices.

START Frame Delimiter	Frame Header	Start Address	DATA	1. End Delimiter	CRC	2. END Delimiter
1	2	2	max. 1024	1	2	1

Figure 2.12: VABS frame structure

VABS is based on the master/slave mechanism to convey synchronous traffic and a token passing for asynchronous traffic. As EtherCAT and SERCOS III, VABS supports a line network topology that forms a logical ring through sending back the frames by the last node due to the full-duplex links. Moreover, the star topology can be handled by using specific equipment, i.e. VABS switch.

The idea of this protocol is to transmit the real-time traffic on-the-fly, and if the node is transmitting an asynchronous traffic, then it preempts the transmission to send the real-time traffic. The transmission of the preempted traffic will be resumed at the preempted byte after the real-time traffic. The synchronous data transmission is similar to EtherCAT, the master sends an empty frame with predefined fields to enable slaves inserting their data on the fly.

This protocol is similar to EtherCAT but the transmission of the non-real-time traffic can

be preemptited and is managed through token passing mechanism. Hence, VABS presents the same pros and cons than EtherCAT and SERCOS III in terms of costs, reconfiguration efforts and reliability.

2.2.4 Event-Triggered Solutions with Dynamic Redundancy

Ethernet/IP (for Industrial Protocol) is a 100Mbps network developed by Rockwell Automation in 2001 and supported by the Open DeviceNet Vendor Association (ODVA). Ethernet/IP uses CIP (Common Industrial Protocol) [47], which allows the use of off-the-shelf products that are compatible with the TCP-UDP/IP stack and the IEEE 802.3 [6] standard. Ethernet/IP is based on CIP connections, which define the type of packet that will be produced on the network. Two categories of connections are defined: Explicit Messaging and Implicit Messaging. The former is used for generic communications between two nodes, whereas the latter is specific to I/O applications and uses UDP rather than TCP protocol. Ethernet/IP uses the 802.1Q tag to affect a VLAN-ID and a priority level to the real-time messages, to enable transmission within switches before non-real-time messages.

To favor the real-time communication on top of Ethernet/IP and support safety requirements, OADV have introduced in 2008 the Device Level Ring (DLR) mechanism, based on ring topology. The DLR mechanism is based on a ring controller, called ring supervisor, which collects data from the other interconnected nodes on only one port to avoid infinite traffic loop, except some specific frames, i.e., beacons. Each equipment has two Ethernet interfaces and an integrated switch, which implements Store & Forward mechanism and Static Priority service policy. Moreover, fault detection and reconfiguration mechanisms are handled within the controller via specific messages, i.e., beacon and announce, similar to the MRP. Fig. 2.13 shows the nominal operating mode of DLR.

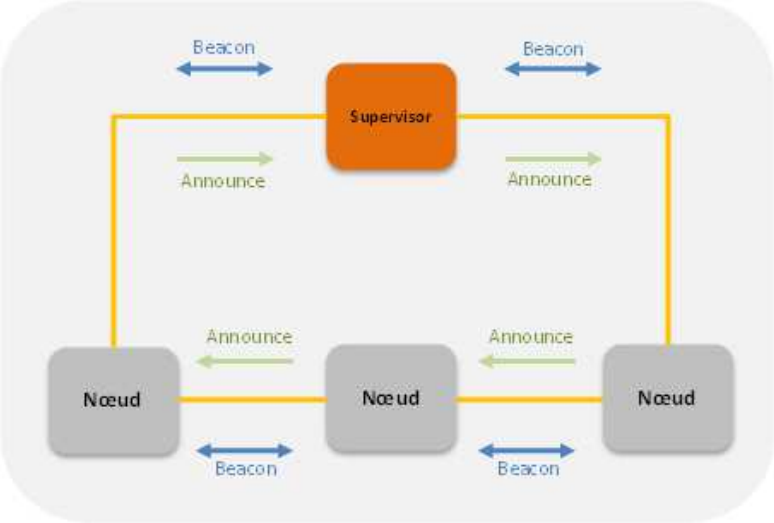


Figure 2.13: Nominal operating mode of DLR

This protocol has interesting features in terms of reliability due to the fault detection mechanism within the controller, and reduced costs due to standard devices. Beacon messages

are sent in both directions of the ring to detect failures. In case of lost and after a timeout, the non-nominal mode is triggered. The supervisor is informed by the location of the fault due to the status messages, which are used by nodes to report the status of their neighbours. The supervisor releases its blocked port to allow data retransmission, which transforms the ring into a line topology. Afterwards, it sends an announce message to inform all the nodes of this topology change. Fig. 2.14 shows the network reconfiguration in case of a link or node failure. If the network is restored, then the supervisor starts receiving again the beacon frames in its both ports. Hence, it sends immediately announce messages to reconfigure the network according to the ring mode.

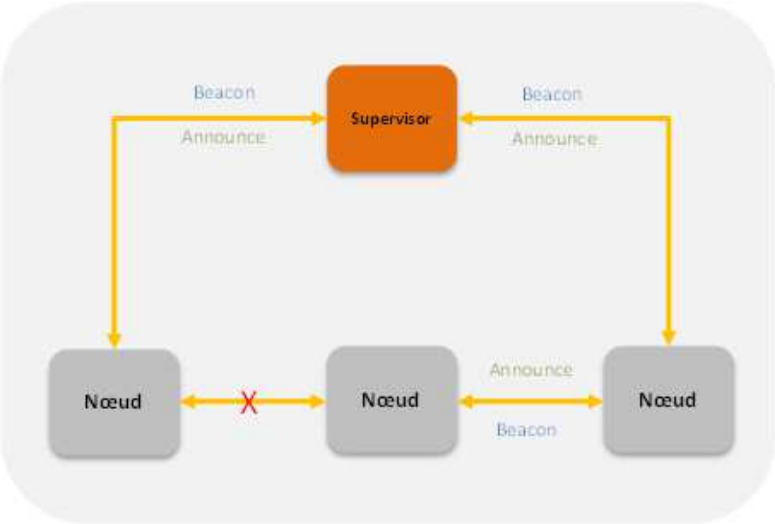


Figure 2.14: DLR behaviour after a failure

Ethernet/IP with DLR has interesting features in terms of reliability due to the fault detection mechanisms within the controller, and of reduced costs due to its standard devices. However, the non-nominal case needs the reconfiguration of the supervisor, which increases the reconfiguration effort. Furthermore, integrated switches based on Store & Forward mechanism induce high transmission latencies, which decrease the offered real-time performance and availability levels.

2.2.5 Discussions

The summary of the main characteristics of the most relevant RTE solutions, i.e. EtherCAT, Profinet and Ethernet/IP with DLR, described above is illustrated in Table 2.4, and their benchmarking vs the main identified avionics requirements is illustrated in Table 2.3.

EtherCAT and Profinet/IRT imply higher costs due to the specificity of the implemented devices and synchronization protocol, and lower reliability due to the master/slaves mechanism, than Ethernet/IP with DLR. The latter is based on standard devices and implements fault detection and reconfiguration mechanisms, which enhance costs and reliability. Concerning predictability and availability, EtherCAT and Profinet/IRT allow short latencies due to on-the-fly and Cut Through mechanisms, whereas Ethernet/IP with DLR induces high

Protocols	Reliability	Availability	Predictability	Modularity	Costs
EtherCAT	Medium	High	High	Low	High
PROFINET/IRT	Medium	Medium	Medium	Low	High
Ethernet/IP with DLR	High	Low	Low	High	Medium

Table 2.3: Benchmarking of RTE solutions supporting ring topology

Characteristic	EtherCAT	PROFINET IRT	Ethernet/IP
Rate (Mbps)	100	100	100
Topology	Bus or ring	Bus or ring	Daisy-chain ring
Media	100Base-TX	100Base-TX	100Base-TX
Control Mechanism	Master/slaves	Master/slaves	Event-triggered with SP policy
Robustness management	centralized (specific)	centralized (MRP)	centralized (DLR)
QoS management	no	no	yes
Standardization	Open standard	Open standard	By OADV
Pros	On-the-fly transmission Short transmission cycle	Cut-through transmission Short transmission cycle	Efficient faults detection QoS Management
Cons	Specific devices Central point of failure	Specific devices Central point of failure	Complexity due to integrated switches High latency

Table 2.4: Specifications Comparison of RTE solutions supporting ring topology

latencies because of the Store & forward one. Moreover, these transmission latencies have a direct effect on the fault detection time, and consequently the availability level. Hence, the offered predictability and availability levels of EtherCAT and Profinet/IRT are higher than Ethernet/IP. Finally, concerning modularity, Ethernet/IP offers higher modularity level thanks to the implemented event-triggered paradigm, in comparison to EtherCAT and Profinet/IRT due to the master/slave mechanism.

As we can notice, each RTE solution satisfies some requirements better than others, but there is no solution satisfying all them. Hence, the new RTE solution has to guarantee high reliability and modularity levels as Ethernet/IP with DLR, while enhancing the predictability and availability levels to be comparable to EtherCAT and Profinet/IRT. This comparative analysis based on qualitative criteria will be consolidated through quantitative benchmarking in the next section.

2.3 Quantitative Benchmarking

The document IEC 61784-2 [1] has introduced a set of Performance Indicators (PIs) to evaluate the RTE networks abilities. In this section, we first describe the most effective PIs in an avionics context. Then, we describe a representative avionics case study, considered as a reference to assess the PIs of the relevant RTE solutions described in Section 2.2. Finally, we detail and discuss the obtained results for each solution.

2.3.1 Performance Indicators

The main PIs to compute have been defined in [1] and the most relevant ones in avionics are:

- Maximum Delivery Time: indicating "the time needed to convey an APDU containing data (message payload) that has to be delivered in real-time from one node (source)

to another node (destination)" when considering the worst-case scenario. This PI is of utmost importance in avionics to conclude on the network *predictability*, since we need to guarantee that the maximum delivery time of any type of traffic is lower than its associated temporal deadline;

- Maximum number of end-stations: in [1], this indicator represents the maximum number of stations that can be supported by the RTE solution. In the avionics context, such an indicator has to give an idea on the network *scalability*, while respecting the time constraints, i.e., the maximum number that still respects the temporal deadlines of any type of flow exchanged on the network;

- RTE Throughput: it "shall indicate the total amount of APDU data (in bytes) on one link per second". This parameter allows assessing the resource utilization *efficiency* of the alternative solution, thus to evaluate its maintainability during the long lifetime of an avionics system (about 20 to 30 years), which needs an easy incremental design process for adding functions along this duration.

- Non-RTE Bandwidth: it "shall indicate the percentage of bandwidth, which can be used for non-RTE communication on one link". This parameter can be considered as a complementary one to evaluate the effectiveness of the alternative solution, in terms of resource utilization *efficiency*;

- Redundancy recovery time, indicating "the maximum time from failure to become fully operational again in case of a single permanent failure". This indicator is essential to evaluate the network *availability*, a key requirement in avionics.

Table 2.5 describes the used notations in this part to compute the PIs of the main RTE solutions, i.e., EtherCAT, Profinet IRT and Ethernet/IP.

Terms	Notations	Units
Minimum Cycle Time	MCT	s
Transmission time	τ	s
Technological latency	l	s
Propagation time	δ	s
Rate	R	bit/s
Period	T_{period}	s
Data payload	x	$bytes$
Number of nodes	n	–

Table 2.5: Notations for PIs computation

The technological latencies of these solutions are grouped in Table 2.6. Moreover, the propagation delay is typically less than 50 ns per 10m for cable of category 5 [57, 58].

Solution	Latency
Ethernet/IP	$3\mu s$
Profinet IRT	$3\mu s$
EtherCAT	$1.35\mu s$

Table 2.6: Technological latencies

2.3.1.1 Maximum Delivery Time

Calculation of the maximum delivery time shall include the transmission time as well as any waiting time. In cyclic network communication, we are interested in the Minimum Cycle Time (MCT). To model the RTE solutions, we consider the following assumptions:

- Only cyclic communications are enabled;
- the network is initiated, i.e. initiation phase is not counted;
- all transmitted data have the same size;

EtherCAT

Fig. 2.15 shows a spatio-temporal diagram of an EtherCAT frame transmission. According to this figure, the MCT is the sum of [59, 58, 60, 57, 61]:

- the transmission time;
- technological latencies of all crossed slaves (round trip);
- links propagation delays (round trip);

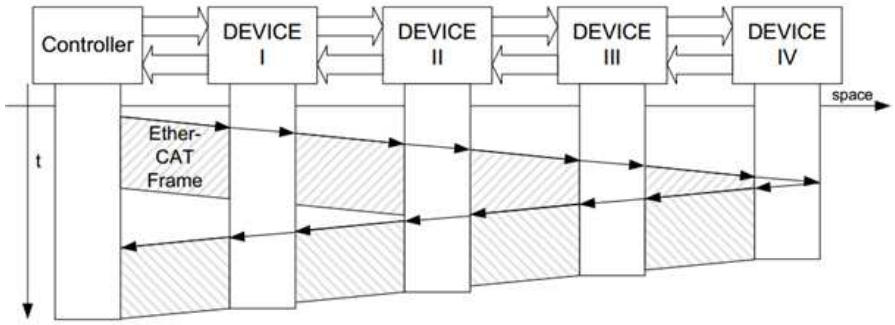


Figure 2.15: EtherCAT spatio-temporal diagram

To compute the transmission time, we need to express the EtherCAT frame size. An EtherCAT frame has an overhead of 40 bytes (26 bytes for the Ethernet header, 2 bytes for the EtherCAT header and 12 bytes for the inter-frame gaps) and contains several data telegrams, where each one has a size x and a header of 12 bytes. To respect the minimum Ethernet frame

payload size, i.e. 46 bytes, a padding is added if the sum of telegram sizes is less than 44 bytes (46 bytes minus 2 bytes of the EtherCAT header). Hence, the transmission time is:

$$\tau = \frac{8(40 + \max(44, n(12 + x)))}{R}$$

The cycle time is then computed as follows:

$$\begin{aligned} MCT &= 2nl + 2n\delta + \tau \\ &= 2nl + 2n\delta + \frac{8(40+\max(44,n(12+x)))}{R} \end{aligned} \tag{2.1}$$

This formula corresponds to the transmission of a single EtherCAT frame. However, it can be used only if the number of telegrams is less then: $n \leq n_{max} = \left\lfloor \frac{1500-2}{12+x} \right\rfloor$. If the number of telegrams is greater than n_{max}, then, the master needs to send more than one frame. Therefore, the number of needed frames to convey n telegrams is given by:

$$k = \left\lceil \frac{n}{n_{max}} \right\rceil$$

The first $(k-1)$ frames are sent encapsulating n_{max} telegrams and the last one conveys the remaining ones. Padding is added to the latter if necessary to reach the minimum Ethernet frame size. These facts induce the following formula:

$$\begin{aligned} MCT = 2nl + 2n\delta \quad &+ \tfrac{8}{R}(k-1)(40 + n_{max}(12 + x))) \\ &+ \tfrac{8}{R}(40 + \max(44, (n - (k-1)n_{max})(12 + x))) \end{aligned} \tag{2.2}$$

Profinet IRT

To model the Profinet IRT cycle time, we consider the slipstreamimg effect [62, 58, 57, 61], which consists in sending frames in the same physical order of the destinations from the master.

Fig. 2.10 shows a spatio-temporal diagram of Profinet. The slipstreaming effect is beneficial and considered only when it is positive, which means $\tau \geq \delta + l$, i.e., $\alpha \geq 0$. Consequently, the Profinet IRT MCT is:

$$MCT = \delta + l + n\tau \tag{2.3}$$

Profinet IRT uses Ethernet frames with two additional fields within the standard payload field: a 2-bytes identifier and a 4-bytes status information. Then, padding is added if necessary to respect the minimum Ethernet frame size. The transmission frame size is as follows:

$$\tau = \frac{8(38 + \max(46, 6 + x))}{R}$$

Therefore, formula (2.3) becomes:

$$MCT = \delta + l + n\frac{8}{R}(38 + \max(46, 6 + x)) \tag{2.4}$$

However, if the slipstreaming effect is not beneficial, i.e., $\tau < \delta + l$, then, the MCT formula becomes:

$$MCT = n(\delta + l) + \frac{8}{R}(38 + \max(46, 6 + x)) \tag{2.5}$$

Ethernet/IP with DLR
The MCT of this RTE solution is equal to the sum of [48]:

- transmission time of the beacon frame within each switch;
- transmission time of data within each switch;
- propagation delay and technological latency related to the beacon frame and each data.

Data are transmitted in Ethernet frames, where each frame consists of a 26-bytes Ethernet header, 28-bytes UDP/IP header, 18-bytes CIP header and 12-bytes of IFG. The beacon frame is a minimum Ethernet frame, i.e. 64-bytes frame, plus a 12-byte IFG and 8-bytes preamble.

The MCT is given by:

$$MCT = n(\frac{8(84 + n(84 + x))}{R} + \delta + l) \tag{2.6}$$

2.3.1.2 Throughput RTE

Based on its definition, throughput RTE depends on the links capacity, the data rate and the protocol overhead. Throughput RTE can be computed as follows:

$$R_{RTE} = \sum_{i=1}^{k} (x_i \times packet\_rate_i) \tag{2.7}$$

where:

x_i	is the data size in byte for flow i;
$packet\_rate_i$	is the packet rate of flow i in packets per second (pps);
k	is the number of flows per node;

In cyclic communications, the throughput RTE is given by:

$$R_{RTE} = \frac{\sum_{i=1}^{k} (x_i)}{MCT} \tag{2.8}$$

2.3.1.3 Non-RTE bandwidth

Based on its definition, the non-RTE bandwidth is computed by dividing the remaining time of the period on the minimum cycle time:

$$D_{NRT} = \frac{T_{period} - MCT}{T_{period}} \tag{2.9}$$

2.3.1.4 Redundancy Recovery Time

Redundancy recovery time shall indicate the maximum time from failure to become fully operational again in case of a single permanent failure.

EtherCAT is based on a software implementation at the application layer to detect a failure by the master. The slave that detects a failure returns immediately the message to avoid losing the communication with the rest of the nodes. Once the master detects the failure, it activates the second port to allow the communication with the isolated nodes due to the failure. In the worst case, a single communication cycle is affected [1]. Hence, the maximum recovery time of EtherCAT is as follows:

$$T^{max}_{recovery} = MCT$$

Profinet IRT uses the MRP protocol [37]. The MRM nodes detect the failure after the loss of a minimum of test frames, i.e., "$TSTNR\_max$" of "MRP-Tests". These frames are sent in an interval of time "$TSTShortT$" in the best case if the MRM does not receive an answer from the MRC node. In both cases, the detection time should not exceed a MCT:

$$T^{max}_{detection} = MCT$$

After the detection, the MRM sends several "$MRP\_TopologyChange$" frames to inform the MRC nodes about the new changes. If we consider that these messages are sent in the same cycle and $T_{setupFDB}$ is the time to update the node database, the update time is equal to:

$$T^{max}_{update} = MCT + T_{setupFDB}$$

The maximum recovery time of Profinet IRT is as follows:

$$T^{max}_{recovery} = T^{max}_{detection} + T^{max}_{update} = 2 \times MCT + T_{setupFDB}$$

The recovery Ethernet/IP with DLR time is given by formula (2.10) according to [48]:

$$T^{max}_{recovery} = T^{max}_{detection} + T^{max}_{update} \tag{2.10}$$

where:

- $T^{max}_{detection} = 2 \times 0.5 \times RndTripTime + 3 \times DLR\_Resp\_T = MCT + 3 \times DLR\_Resp\_T$

- $T_{update}^{max} = MCT + RndTripTime^{max} = 2 \times MCT$
- $RndTripTime$ is the time to cross all the ring
- DLR_Resp_T is the time to response

The recovery time of Ethernet/IP with DLR becomes:

$$T_{recovery}^{max} = 3 \times MCT + 3 \times DLR\_Resp\_T$$

Table 2.7 summarizes the maximum recovery time of these three solutions:

	EtherCAT	Profinet IRT	Ethernet/IP with DLR
Maximum Recovery Time	1 MCT	2 MCT	3 MCT + 3 × DLR_Resp_T

Table 2.7: Maximum Recovery Time of the three main solutions

2.3.2 Reference Case Study

The considered case study is a representative avionics communication network setup, which supports three types of flows: the I/O data initially transmitted on the CAN and ARINC 429, the legacy AFDX flows and audio data for cabin management. Furthermore, we consider the following assumptions:

- The network topology is a ring;
- The links speed is $C = 1Gbit/s$ (we enlarge the capacity of EtherCAT, Profinet/IRT and Ethernet/IP to 1Got/ps);
- The network size varies from 5 to 100 nodes;
- All devices are similar and send the same traffic in broadcast mode;
- Each device generates one flow of each type of traffic, described in Table 2.8.

Table 2.8: Traffic Characteristics

	Priority	MFS (byte)	BAG (ms)	Deadline (ms)
I/O data	High	8	2	2
Legacy AFDX	Medium	1300	10	4
Audio	Low	160	20	20

2.3.3 Numerical Results

Fig. 2.16 illustrates the maximum delivery time of the different RTE solutions supporting the ring topology. There are mainly two interesting observations through this figure. The first one confirms the qualitative benchmarking in Section 2.2.5 in terms of predictability requirements. Ethernet/IP has actually the highest delivery time due to the Store & Forward mechanism, in

comparison to EtherCAT and Profinet IRT, which are based on on-the-fly and cut-through mechanisms, and have quite similar performance for I/O data. The second observation concerns the network scalability, where the maximum number of RTE end-systems respecting the most constrained deadline, i.e., I/O deadline of 2ms, is about 8, 70 and 76 for Ethernet/IP, EtherCAT and Profinet/IRT, respectively. This result show the good scalability of EtherCAT and Profinet/IRT.

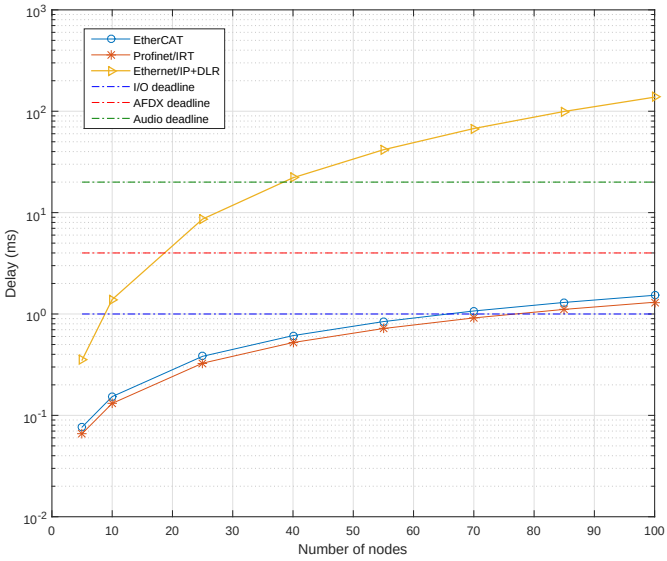


Figure 2.16: Maximum Delivery Time of Ring-based RTE solutions

Concerning resource efficiency of the different RTE solutions, we can observe Figures 2.17, 2.18, 2.19 and 2.20 illustrating the RTE throughput for I/O, AFDX and audio traffic, and the Non-RTE bandwidth of the different solutions, respectively. The obtained results show the high efficiency of EtherCAT and Profinet/IRT, in comparison to Ethernet/IP. This fact is mainly related to the short MCT of EtherCAT and Profinet/IRT due to the on-the-fly and cut-through implemented mechanisms, which allow to deliver the data in a short duration; Thus, increase the Non-RTE bandwidth.

Finally, to assess the availability level of the different RTE solutions, the maximum recovery time is shown in Fig. 2.21. The results confirm again the first qualitative conclusions in Section 2.2.5, where EtherCAT and Profinet/IRT have almost similar availability levels, which are much better than the one offered by Ethernet/IP.

Based on this quantitative analysis of the most relevant PIs, we can adjust the conclusions of Section 2.2.5 concerning the expected behavior of each described RTE solution vs the predictability and availability requirements. For predictability, we can notice that under 1Gbps, Profinet/IRT offers better performance than EtherCAT. This fact is mainly due to the slipstream method of Profinet/IRT and the impossibility of grouping many large-sized data in one frame for EtherCAT. Therefore, we upgrade the predictability level of Profinet/IRT

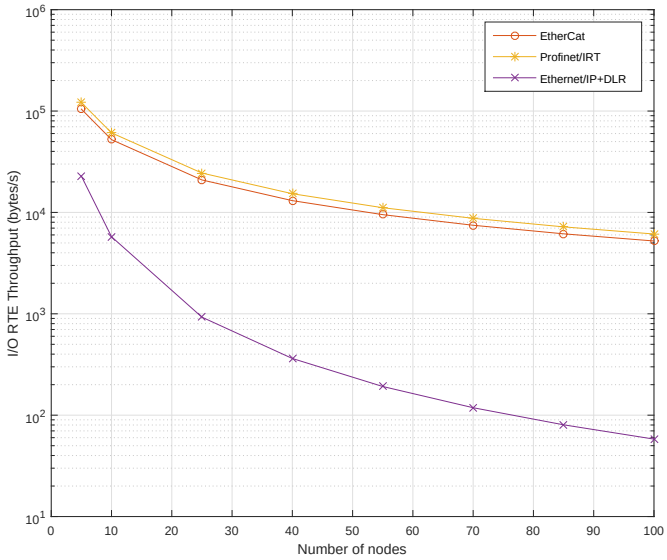


Figure 2.17: RTE Throughput of Ring-based RTE solutions for I/O traffic

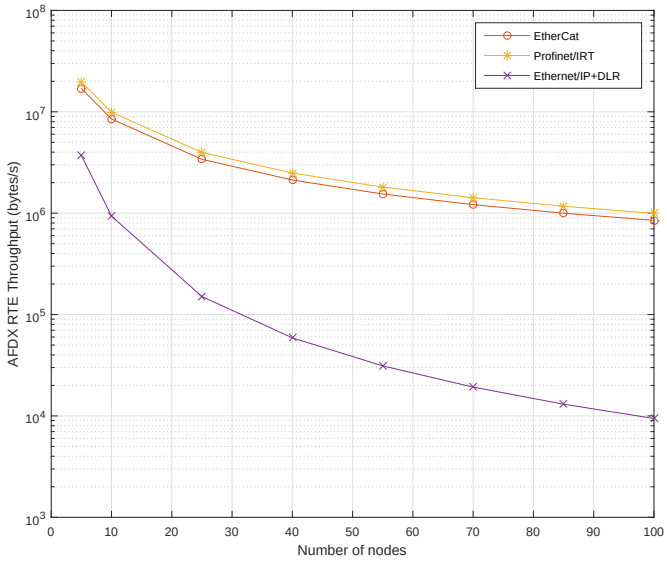


Figure 2.18: RTE Throughput of Ring-based RTE solutions for AFDX traffic

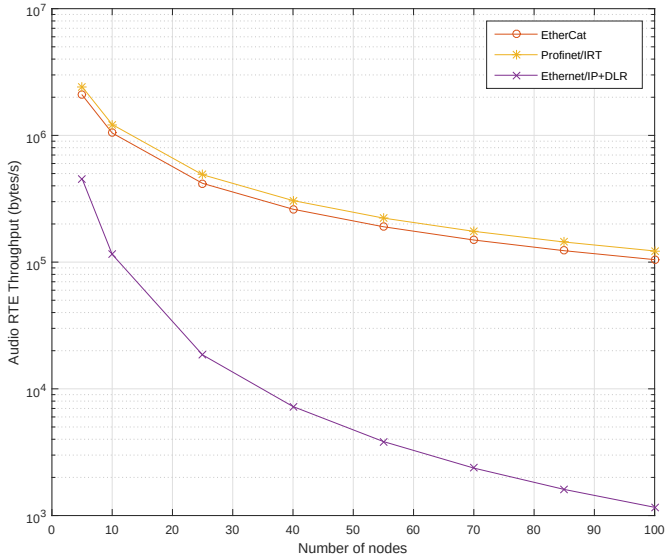


Figure 2.19: RTE Throughput of Ring-based RTE solutions for audio traffic

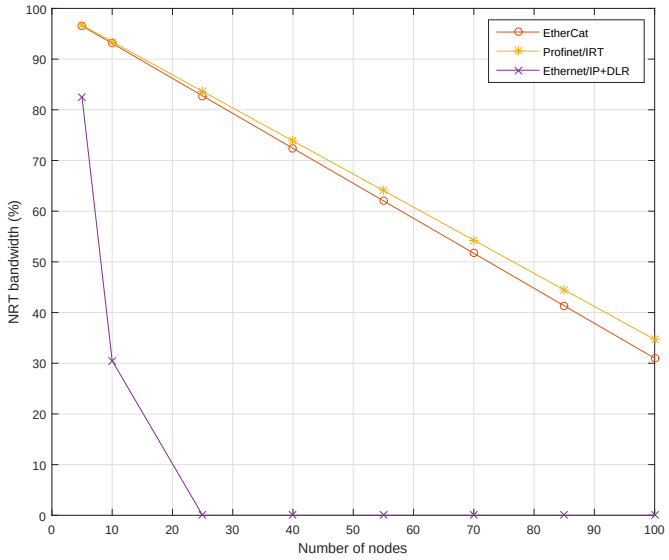


Figure 2.20: Non-RTE Bandwidth of Ring-based RTE solutions

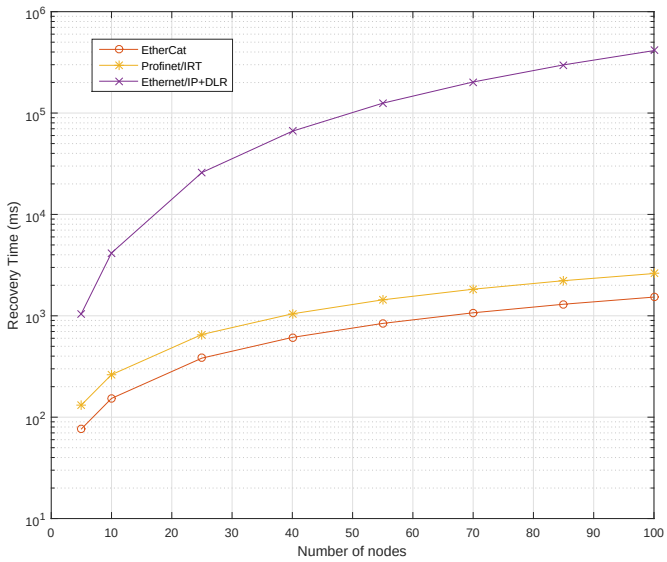


Figure 2.21: Redundancy Recovery Time of Ring-based RTE solutions

from medium to high. For availability, we have exactly the same observations based on the redundancy recovery time. Hence, we obtain the updated Benchmarking Table 2.9. The latter shows that there is no perfect RTE solution for avionics, which offers a high modularity level with a low deployment costs, while guaranteeing high availability, reliability and predictability levels.

Protocols	Reliability	Availability	Predictability	Modularity	Costs
EtherCAT	Medium	High	High	Low	High
PROFINET/IRT	Medium	High	High	Low	High
Ethernet/IP with DLR	High	Low	Low	High	Medium

Table 2.9: Updated Benchmarking of RTE solutions supporting ring topology

2.4 Conclusion

In this chapter, we have presented a brief overview of the standard Ethernet and the most relevant ring-based RTE solutions. The qualitative and quantitative benchmarking of these solutions, according to the most relevant PIs in avionics, reveals that each of the studied solution satisfies some avionics requirements better than others, but there is no best solution regarding all the requirements. EtherCAT and Profinet IRT imply higher costs due to the specificity of the implemented devices, and lower robustness and modularity due to the master/slaves mechanism, in comparison with Ethernet/IP with DLR. However, concerning the predictability, Ethernet/IP offers lower levels due to high latencies because of the Store & Forward mechanism, in comparison to EtherCAT and Profinet/IRT. Moreover, these transmission latencies have a direct effect on the failure detection time, and consequently the availability level.

In the next chapter, we present a new RTE solution to bridge the gap between aforementioned solutions, through guaranteeing similar reliability and modularity levels than Ethernet/IP with DLR, and similar predictability and availability levels than EtherCAT and Profinet/IRT.

Chapter 3

Specification of AeroRing

Contents

In this chapter, we first introduce the main objectives and features of AeroRing and the main supported topologies. Then, we detail the main mechanisms to guarantee high timeliness and availability levels, including real-time and QoS management as well as dynamic redundancy protocol. Finally, we present the analytical formula of the main Performance Indicators of AeroRing.

3.1 Main Objectives

The main objective of AeroRing is to enable a homogeneous communication architecture for avionics. This solution shall integrates different avionics domains, such as Flight Control, Legacy AFDX Systems and In-Flight Entertainment. This fact may bring significant advantages, such as quick installation and maintenance and reduced weight and costs.

To achieve this aim, AeroRing has to guarantee the main avionics requirements, while limiting the implementation costs and (re)configuration efforts.

The benchmarking of the existing solutions, detailed in previous chapter, has inferred important insights into the high-level specifications of an ultimate solution for avionics. Therefore, AeroRing fulfills the most relevant requirements as follows:

- Guaranteeing an easy deployment process and a **cost-effective integration** due to its *IEEE 802.3 Compatibility* and Enabling *various ring-based topologies*, i.e., simple or duplicated mono-ring and multiple-ring topologies, based on auto-configuration mechanisms;
- Providing a high **modularity** level and reducing the (re)configuration effort, through implementing an *event-triggered communication paradigm*;
- Favoring **predictability** using *QoS-aware routing algorithm and traffic policing* mechanisms, to handle heterogeneous data constraints;
- Offering a high **availability** and **reliability** levels thanks to a *Dynamic Redundancy Protocol*, which bridges the gap between the existing static and dynamic redundancy solutions, in terms of costs and flexibility, while improving the resource efficiency, i.e., reducing fault detection overhead and increasing the network utilization rate.

3.2 Main Features

AeroRing is a daisy-chain network that allows any "Ethernet-compliant" equipment to transmit its data in the network via a specific end-system, called T-AeroRing. Each transmitted packet will be forwarded from one T-AeroRing to another until reaching the final destination.

The T-AeroRing is a specific 3 ports Full Duplex Ethernet switch having the internal architecture illustrated in Figure 3.1, and the following main characteristics:

- ***Cut-Through forwarding technique***: the T-AeroRing starts forwarding the packet just after its identification, i.e., only the header of each packet is decoded to determine its destination port. This technique guarantees shorter transmission latency than the Store

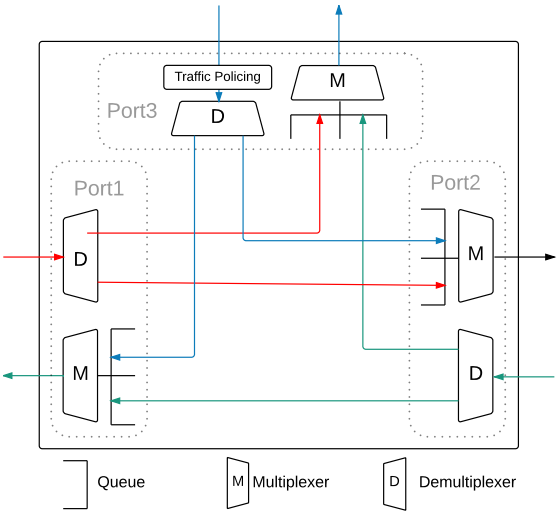


Figure 3.1: *T-AeroRing* internal architecture

& Forward technique (implemented within Ethernet/IP), which waits until the complete reception of the packet before forwarding it to the destination port;

- ***Static Priority service policy*** packets are queued in each output port of T-AeroRing according to their priorities. A queue is selected for transmission only if all traffic classes queues with higher priorities are empty. Then, for each queue, the scheduling order is First In First Out (FIFO) with a non-preemptive transmission. Priorities are defined according to the IEEE 802.1p standard where the 802.1Q tag (3-bits field) is used to manipulate four priority classes: the control traffic class with the highest priority, the Hard Real Time (HRT) class with the second highest priority, the Soft Real Time (SRT) class with medium priority and finally the Non Real Time (NRT) class with the lowest priority;

- ***Traffic policing***: To guarantee real-time performance, the T-AeroRing implements traffic policing mechanisms, based on Leaky Bucket method and particularly greedy method [63], to control each traffic class compliance with its predefined contract to avoid the network saturation. These traffic contracts are defined based on the network designer specifications. Each equipment connected to a T-AeroRing should be aware of these traffic contracts, and may apply traffic shaping to ensure the conformity of its generated traffic and avoid being discarded by the traffic policers. Each traffic exceeding its associated contract may be discarded to guarantee the communication determinism;

- ***QoS-aware routing***: unlike COTS Ethernet switches which relay frames on the basis of the address learning process and the Spanning Tree Algorithm, each T-AeroRing builds its routing table based on control messages exchanged between the interconnected

T-AeroRings, during the initialization phase or when a topology modification occurs (i.e., failure or restoration). Each T-AeroRing implements two routing modes to transmit its generated packets depending on their priorities: (i) sending on both ring ports (Ports 1 and 2 in Fig. 3.1) for high priority traffic classes, i.e., control and HRT data, to allow a high reliability level ; (ii) sending on the port corresponding to the shortest path for medium and low priority traffic classes, i.e., SRT and NRT data, to offer a high performance level i.e., short delay;

- ***AeroRing Redundancy Protocol (ARRP)***: Similarly to RRP [38], ARRP integrates dynamic mechanisms for fault detection and reconfiguration of routing tables, based on control messages. However, ARRP improves resource efficiency, through decreasing the control overhead and using the multi-path feature. Moreover, unlike the main dynamic redundancy protocols, ARRP enables the full use of ring topology, i.e., the ring topology is not transformed into a line by blocking some forwarding ports, due to its filtering mechanisms to avoid infinite message looping.

3.3 Supported Topologies

AeroRing supports several topologies. The first one is a classical ring daisy-chain implementation, called the simple mono-ring and illustrated in Fig. 3.2. In this topology, T-AeroRings are connected in a daisy chain mode using the ring ports, i.e., ports 1 and 2 in Fig. 3.1, whereas port 3 is used to connect the communicating equipment.

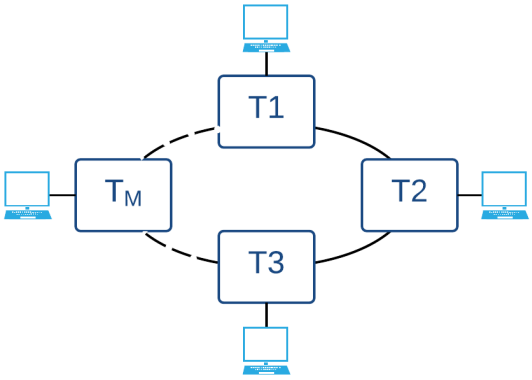


Figure 3.2: *Mono-ring* network architecture

The second AeroRing topology is the simple multiple-ring. The key idea is to gather nodes in peripheral rings according to their exchanged data, as shown in Fig. 3.3. This fact will decrease the end-to-end delays, which depend on the data path length; thus, decreasing the fault detection and reconfiguration times and consequently enhancing the availability level. Moreover, this topology may improve the throughput within each peripheral ring, since it isolates the intra-ring traffic from the inter-ring one.

The peripheral rings are connected to the backbone ring via gateways, which manage the inter-ring communications. The gateway is a specific T-AeroRing and its main function is guaranteeing the QoS-aware routing between the peripheral rings. This feature will be detailed in Section 3.4.2.

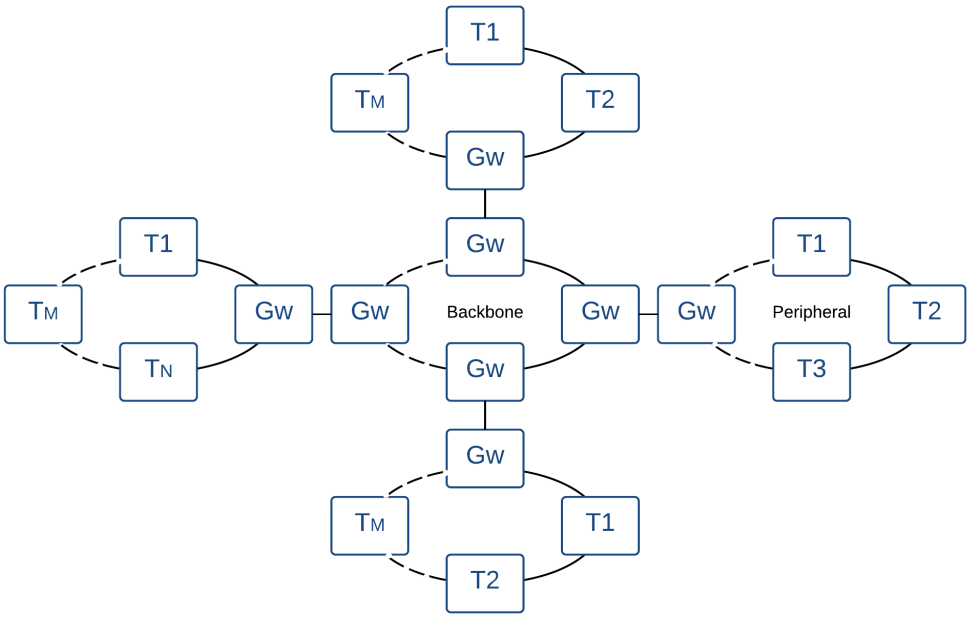


Figure 3.3: *Multiple-ring* network architecture

In addition to these two main topologies, AeroRing supports duplicated mono-ring and multiple-ring topologies. Each equipment may be connected to redundant T-AeroRings and transmit its data on both AeroRing networks (see Fig. 3.4). The redundancy management for such topologies can be handled with classic static redundancy protocols, such as PRP [38].

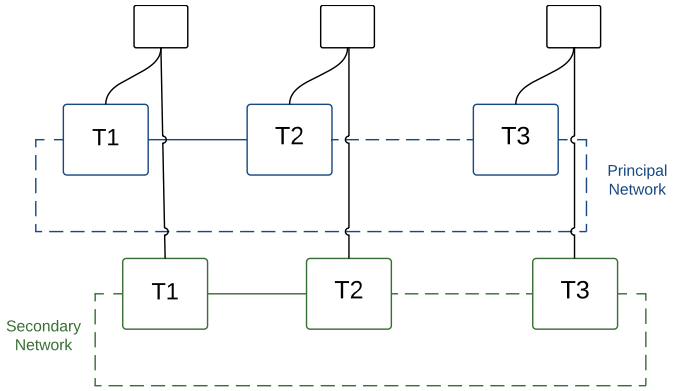


Figure 3.4: Example of a duplicated mono-ring topology

3.4 Real-Time Mechanisms and QoS Management

In this section, we first describe the data flow types supported by AeroRing. Afterwards, we detail the QoS-aware routing algorithm. Finally, we present the real-time mechanisms, which favor the timeliness guarantee.

3.4.1 Data Flow Types

AeroRing guarantees QoS management through the implementation of "Static Priority" policy, which supports the following data flow types:

1. ***HRT data flow***: this traffic has the highest priority level (N1) and is generally generated by real-time applications with hard temporal constraints, i.e., each message must be received before its deadline, otherwise it is considered as lost. This type of data flow is sent on both ring ports to ensure a high reliability level, and is identified by a 2-bytes sequence number, essential to filter redundant messages within the destination T-AeroRing;

2. ***SRT data flow***: this traffic mainly sent by soft real-time applications, such as audio or video transfers, has the medium priority level (N2). This type of data flow is sent on the ring port corresponding to the shortest path to guarantee a high performance level, i.e., short transmission delay;

3. ***NRT data flow***: this traffic corresponds to non real-time applications, such as file transfer, and has the lowest priority level (N3). This type of data flow is sent on the ring port corresponding to the shortest path to guarantee a high performance level.

The T-AeroRing priorities are handled according to the IEEE 802.1Q specification, as shown in Fig. 3.5, where:

- the 3-bits PCP field indicating the priority level of the message can encode up to 8 levels. The mapping between the four priority levels of AeroRing and the standard eight priority levels is shown in Table 3.1;

- the VID field is used to identify each peripheral ring to which the message belongs in the multiple-ring topology.

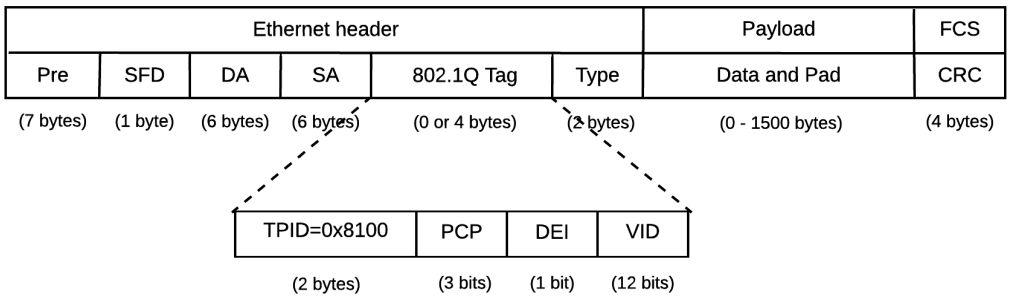


Figure 3.5: *T-AeroRing* frame structure

Field	Priority level	Data type	AeroRing Priority
000	1	Best effort	N3
001	0	Background	N3
010	2	Excellent effort	N3
011	3	Critical App	N2
100	4	Video	N2
101	5	Audio	N1
110	6	Internetwork	N1
111	7	Network control	N0

Table 3.1: AeroRing priority levels

It is worth noting that the AeroRing is compatible with the IEEE 802.3 standard and each *T-AeroRing* can deliver any type of "802.3x-compliant" message from the equipment. Hence, if the message does not include the 802.1Q tag, then it will be treated as a message of NRT data flow type (N3), and transmitted on the ring port corresponding to the shortest path.

AeroRing enables two modes of broadcast:

1. A global broadcast to send data to all the network equipment. Such messages have a default VID value, i.e., $0x000$. Furthermore, broadcast messages without 802.1Q tag are sent following the global broadcast mode;

2. A local broadcast to send data only within a peripheral ring. Such messages have the VID of the corresponding peripheral rings.

3.4.2 QoS-Aware Routing

Based on the description of T-AeroRing ports in Fig. 3.6, each message will be transmitted as follows:

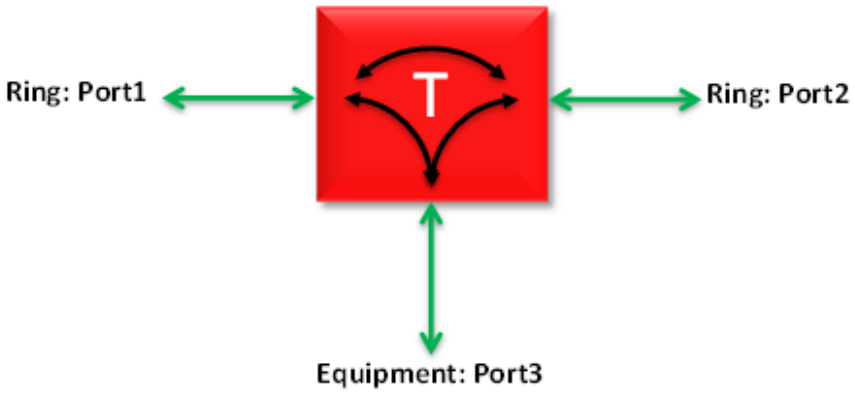


Figure 3.6: *T-AeroRing* different ports

- Messages received from port 3, i.e., from the connected equipment, are transmitted to port 1 or/and 2 according to their priority level, as follows:
  - Messages with priority N1 are sent through both ports.
  - For messages with priority N2 and N3, we distinguish two cases: i) if the final destination belongs to the same peripheral ring than the source, then messages are sent through the port corresponding to the shortest path; ii) else, the messages are sent through the port corresponding to the shortest path to the gateway.
  - Broadcast messages with priorities N2 and N3 are transmitted through a predefined port or a port selected randomly.
- Messages received from port 1 or 2 are treated according to their priority level and destination address. If the destination address corresponds to the connected equipment to the T-AeroRing, then the message is sent to port 3; else, the messages are forwarded to the opposite port. It is worth noting that each message with priority N1 is sent to port 3 only if its replica has not been received yet.

On the other hand, the messages are treated within the gateway as follows:

- For messages received from a ring port, i.e., port 1 or 2, we distinguish three cases:
  - The 802.1Q-tagged (resp. non 802.1Q-tagged) messages are transmitted according to the VID (resp. MAC). If the VID corresponds to the peripheral ring VID (resp. the MAC is within the routing table of the gateway), then the messages are transmitted within the peripheral ring; else, they are transmitted within the backbone ring.
  - The messages compliant with the global broadcast mode are transmitted within peripheral and backbone rings.
- Messages received from the backbone (resp. peripheral) ring are transmitted according to the MAC (resp. VID). If the MAC (resp. VID) is within the routing table of the gateway or is a broadcast address, then the messages are transmitted within the peripheral (resp. backbone) ring, according to their priority levels similarly to a T-AeroRing; else, messages are discarded according to the filtering rules detailed in Section 3.5.3. It is worth noting that for the non 802.1Q-tagged messages received from port 3, we have the same gateway behaviour, except when the MAC address is not within the routing table of the gateway. In this particular case, they are transmitted within the backbone via a ring port selected randomly or a default one.

3.4.3 Real-Time Mechanisms

The real-time behavior of AeroRing and the timeliness guarantee of the delivered data are favored due to the implemented features within the T-AeroRing. First, the "Cut Through" forwarding technique allows a short transmission time along the network, which improves the Maximum end-to-end delivery time. Then, the traffic policing mechanism prevents the network saturation by a deficient equipment, which guarantees the communication determinism.

Furthermore, the implemented QoS-aware routing algorithm supports the transmission of the SRT and NRT data flow on the shortest path, which decreases their transmission delays, and the HRT data flow on both paths to increase the reliability level. Finally, the Static Priority policy ensures the temporal isolation between mixed criticality data with various temporal constraints, and guarantees a bounded delay for the HRT traffic class.

3.5 Safety and Fault Tolerance

AeroRing implements distributed fault detection and reconfiguration mechanisms, which allow improving the reliability and availability of the network. These mechanisms are based on an exchange of control messages, which have the highest priority level N0. Control messages are identified by the type value "0x9000". Figure 3.7 shows the structure of a control message, where the CTL field identifies the type of the control message. Moreover, AeroRing implements filtering mechanisms, which allow detecting the redundant N1 data at the destination nodes and remove invalid messages from the network, to avoid infinite message looping.

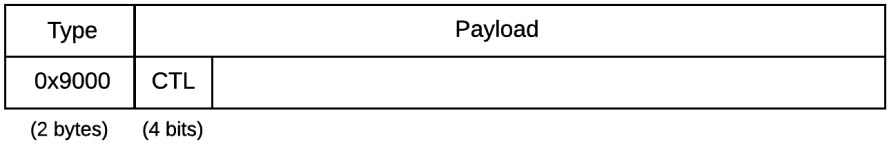


Figure 3.7: Structure of a control message

3.5.1 Fault Detection

To reduce the control messages overhead on the network, AeroRing uses a distributed local fault detection mechanism. A T-AeroRing deduces that its neighbour is operational if it receives any frames from it. Hence, any T-AeroRing has to consider a connection as down with a neighbour, if it does not receive any message from its neighbour during a certain period called "detection period". This detection period can be easily tuned by the network designer. In practice, if a T-AeroRing has no data to transmit to its neighbour, then it announces periodically its status to that neighbour through sending control messages. These messages have the structure represented in Fig. 3.7 with the CTL field set to "0000".

These control messages to announce the status to neighbours are sent periodically on the ring ports when at least one of the following conditions is satisfied:

- The *T-AeroRing* does not have any data to send on one of its ring ports during a period called "*announcing period*" (this period is less than the *detection period* that covers in general the reception of more than one control message);
- The *T-AeroRing* did not receive any data or control message from one of its ring ports for a duration equal to the *detection period*. In this case, the T-AeroRing indicates to its neighbour through a control message that the connection is considered as down.

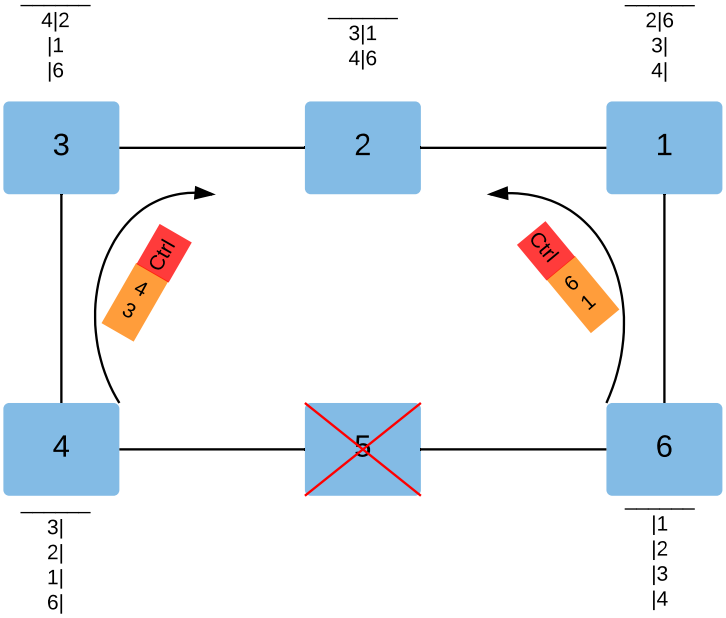


Figure 3.8: Fault detection Mechanism

When a connection is considered as down by one of the interconnected T-AeroRing, the latter sends a first control message (within the faulty ring) to inform the other T-AeroRings about the failure with the CTL code "0010". Then, each T-AeroRing updates its routing tables using the auto-configuration mechanism, detailed in the next section. A down connection is considered operational again (up), if the T-AeroRing starts receiving frames (data or control) from its neighbour. In this case, the auto-configuration mechanism will update the routing tables to consider the ring topology as operational again.

It is worth noting that control messages does not interfere with data messages since they are sent periodically only at the absence of data.

3.5.2 Auto-Configuration Mechanism

To reduce the configuration effort for the network designer and facilitate this new RTE solution adoption in the market, AeroRing offers an auto-configuration service until all the network becomes operational, i.e., updated routing tables within all the T-AeroRings. Each ring performs its auto-configuration mechanism locally and independently from the other rings, i.e., backbone or peripheral.

This service is based on a simple address assignment method and a dynamic network topology discovery process. The address assignment of the connected T-AeroRings method consists in assigning the equipment address to its corresponding T-AeroRing, when it joins the network. This fact facilitates the communication between the connected equipment and avoids a heavy translation addresses step. However, each gateway keeps its predefined MAC address. Moreover, each peripheral ring admits a predefined VID.

Messages are routed within the peripheral rings based on the MAC addresses and within

the backbone ring based on the VID. Hence, peripheral T-AeroRings and gateways routing tables consist of the MAC addresses of the connected equipments, and the backbone gateways routing tables consist of the VIDs. These routing tables allow selecting the port corresponding to the shortest path (ports 1 or 2) for a destination. They are built on the basis of control messages exchanged between the nodes, i.e., T-AeroRings and gateways. The structure of these control messages are illustrated in Fig. 3.9.

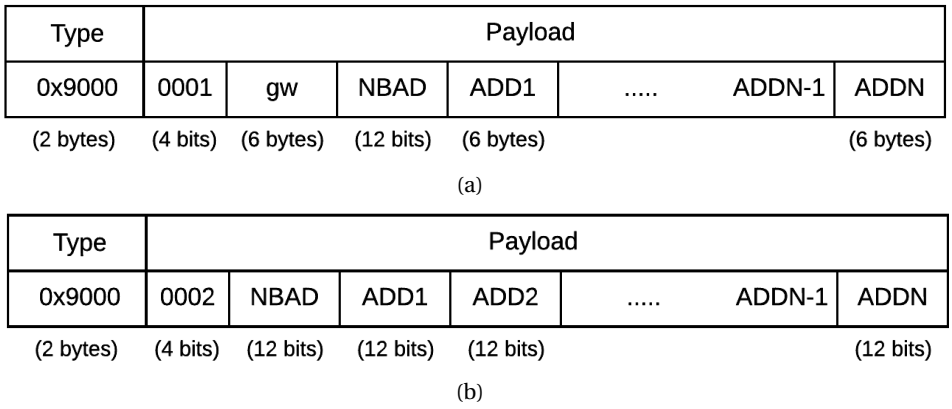


Figure 3.9: (a) Structure of an auto-configuration control message within a peripheral ring; (b) Structure of an auto-configuration control message within a backbone ring.

Control messages used to build the routing tables in a peripheral ring consist of:

- Type field set to "0001";
- Gw field used by the gateway to specify its MAC address;
- NBAD field used as a counter of the addresses inserted in ADDx fields;
- ADDx fields, used by the T-AeroRings to insert their addresses;

On the other hand, control messages used to build the routing tables in the backbone consist of:

- Type field set to "0002";
- NBAD field used as a counter of the VID inserted in ADDx fields;
- ADDx fields, used by the gateways to insert their VIDs;

The control messages to build the routing tables within a peripheral ring are managed as follows:

- At each topology change, i.e., equipment connection, T-AeroRing failure or restoration, the T-AeroRings detecting this event will send periodically control messages on both ring ports with the highest priority, to update the routing tables of the other interconnected T-AeroRings. The NBAD field is set to zero and the ADDx fields are empty;

- Each T-AeroRing contributes in building the routing tables when receiving control messages by:

    1. Incrementing the NBAD counter and inserting its address at the end of the ADDx list to respect the physical order;
    2. Computing the new FCS;
    3. Forwarding it to the next T-AeroRing;
    4. Updating its routing table (i.e., inserts addresses of new equipment and deletes the ones that no longer exist)
    5. Furthermore, the gateway inserts, in addition to its address in the ADDx, its address in the gw field to enable its identification by the other T-AeroRings.

- The T-AeroRing detecting the topology change will stop the periodic transmission when receiving a control message from another T-AeroRing on the same port. This means that it has a neighbour on that side and it is no longer the last node of the segment. This period can be tuned according to the application requirements by the network designer. Then, it stops completely transmitting control messages when detecting both neighbours;

- The control messages transmission stops completely within the ring when the ring loop is closed.

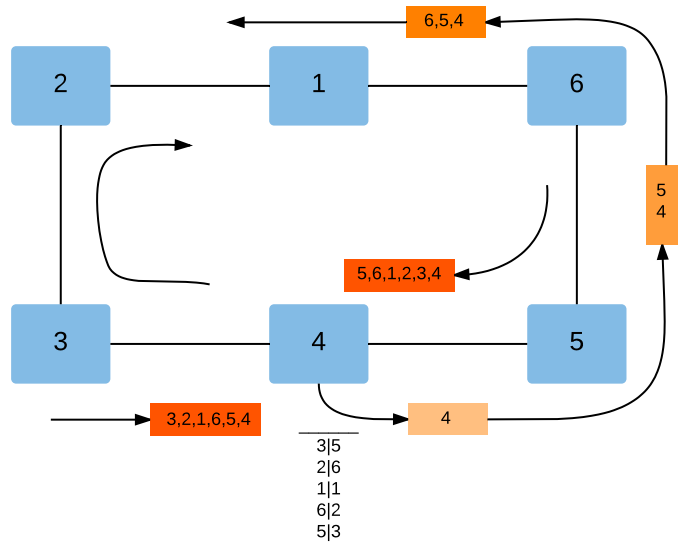


Figure 3.10: Example of routing table building

In the example of Fig. 3.10, node 4 detects a change, so it sends a control message to enable the routing tables update.

When a T-AeroRing receives this control message, it will use the information contained in the ADDx fields to build or update its routing tables to determine the ring port that allows to reach any destination.

Furthermore, each peripheral gateway transmits its routing table to the connected backbone gateway to allow the latter to route inter-rings data, to destination nodes that belong to this peripheral ring, e.g., non 802.1Q-tagged messages.

On the other hand, the auto-configuration mechanism is performed by the gateways within the backbone similarly to the T-AeroRings within a peripheral ring, using the control message with the type 0002 and VIDs instead of MAC addresses.

A control message can contain up to 249 MAC addresses (resp. 998 VIDs) if we respect the maximum Ethernet payload size of 1500 byte. Using the jumbo frames (giant frames) that can go up to 9000 bytes, the size of the peripheral (resp. backbone) ring can be extended to 1499 (resp. 4500) nodes.

3.5.3 Filtering Process

To take a full advantage of redundant paths on ring topologies, mechanisms are needed to avoid infinite messages looping. All dynamic ring-based redundancy protocols, studied in Section 2.2.1.2, deal with this issue by transforming the ring into a virtual line through blocking one or two communication ports. This fact reduces the reliability and resource efficiency of the network. Contrary to these solutions, the ARRP enables communications on both ring directions by implementing some filtering rules to avoid infinite message looping.

Similar to standard Ethernet solution, T-AeroRings support the error detection through the FCS field to discard erroneous frames at their reception. However, if the error is not detected based on the FCS field, and it occurs on the header, then the frame has to be eliminated from the network to avoid infinite packet looping. Messages are filtered from the ring within a T-AeroRing at their reception, if they satisfy one of the following conditions:

- the source MAC address corresponds to the T-AeroRing address;
- the destination MAC address corresponds to the T-AeroRing address;
- neither the source nor the destination is within the ring, i.e., the condition can be checked based on the routing table.

On the other hand, messages are filtered within the gateways if the VID is not within the gateways routing tables. Furthermore, backbone gateways tags messages by a TTL (Time To Live) field equal to the backbone size, which will be decremented by each gateway and removed when it reaches zero. This allows filtering global broadcast messages and non 802.1Q-tagged messages with unknown (erroneous) source or destination MAC address.

In addition to these mechanisms, each T-AeroRing handles the duplicated N1 messages to deliver only the first valid received replica. When a destination receives a message with priority N1, it stores the couple <src MAC, sequence number> in the table, to allow identifying and discarding its replicas. Once the latter is received, or after a timeout, the stored couple <src MAC, sequence number> is removed from the table. It is worth noting that the timeout

is a parameter fixed by the network designer, which must be greater than the maximum end-to-end delay.

3.6 Performance Indicators

The RTE network users have different constraints for different types of real-time applications. Performance indicators are used to specify the ability and guarantees of a RTE network and to define the needs of the real-time applications. The second part of the standard IEC 61784 document [1] defines a set of PIs that allow to chose the communication profiles that guarantee an application needs.

Furthermore, we consider one additional PI, which is: **Fault detection Time**. In this section, we detail the analytical formula of the main PIs detailed in Chapter 2 Sec. 2.3.1 .

3.6.1 Delivery Time

Computing the delivery time for such network based on a ring topology and an event-triggered communication mechanism is considered as a challenging issue due to the cyclic dependencies. The computation of such an indicator is addressed in the next Chapter.

3.6.2 Number of RTE end-stations

Physically, AeroRing with a single ring can support up to 249 nodes using standard Ethernet frames with a maximum payload size of 1500 bytes. Using the jumbo frames (giant frames) that can go up to 9000 bytes, the network size can goes up to 1499 nodes. Using the multiple-ring configuration, several local rings can be connected via the backbone ring. The size of the backbone is equal to the number of local networks, and can go also up to 998 with a maximum payload size of 1500 bytes or 4500 with a maximum payload size of 9000 bytes (which gives a network of 1499×4500 nodes).

However, the guarantees on the communications timeliness, reliability and availability levels have to be proved, which depend on the characteristics of the transmitted flows as well as the size of the network.

3.6.3 Throughput RTE

According to the throughput RTE definition, this indicator depends on the capacity of the link, the real-time flow rate transmitted by each node and the protocol overhead. The real time throughput is calculated by the following formula:

$$D_{RTE} = \sum_{i=1}^{k} (APDU\_SIZE_i \times f_i) \tag{3.1}$$

where:

- $APDU\_SIZE_i$: Payload data size of the flow i;
- ρ_i: rate of the flow i;

- f_i: transmission frequency of flow i packets ($\frac{\rho_i}{packet\_size_i}$);
- k: number of real time flows per node.

An AERORING frame has an overhead of 42 bytes (plus 2 bytes if the frame is HRT): 8 bytes of preamble, 12 bytes of IFG (Inter Frame Gap), 12 bytes for the source and destination address, 4 bytes for the 802.1Q tag, 2 bytes for the type field, 4 bytes for the FCS field and 2 more bytes for the sequence number field for HRT frames.

3.6.4 Non-RTE bandwidth

The non real-time bandwidth is calculated based on the capacity of the link and the total real-time throughput (the percentage of the residual bandwidth). It is given by the following formula:

$$B_{NRT} = \frac{R - \sum_{i=1}^{n} \rho_i}{R} = 1 - U_{RTE} \tag{3.2}$$

where:

- R: links capacity;
- ρ_i: rate of the flow i;
- n: number of real time flows;
- U_{RTE}: RTE utilization rate.

3.6.5 Fault detection Time

A node detects a failure when it loses the connection with his neighbour, or it receives a failure declaration by another node.

The worst case detection time in a network corresponds to the worst case detection time of the node in the middle of the network according to the fault, i.e., the farthest node from failure in both directions. It is equal to the local detection time (detection of the failure by the neighbour) plus the transmission time of a control message to report the fault (64-bytes message), and the maximum delay while crossing the intermediate nodes. The control message can be delayed in the worst case at each node by a maximum packet length of low priority at each crossed T-AeroRing.

$$T_{detection} = T_{local\_detect} + T_{report} + T_{delay} \tag{3.3}$$

The local control messages are sent periodically each T_{local_period} at the absence of traffic. A node detects a failure after the loss of N_{detect} local control messages.

Because of the technological latencies, we add a small time ϵ to the local transmission period to avoid a false loss of a control message.

$$T_{local\_detect} = N_{detect} \times (T_{local\_period} + \epsilon) \tag{3.4}$$

$$T_{report} = \frac{L_{con} \times 8}{R}$$

where:

- L_{con}: size of the control message (64 bytes for the minimum Ethernet frame size and 20 bytes for the preamble and IFG);
- R: links capacity.

$$T_{delay} = \left\lceil \frac{M-1}{2} \right\rceil \times \left(\frac{\max_{pp>0} L_{pp} \times 8}{R} + \epsilon\right)$$

where:

- M: number of nodes;
- L_{pp}: maximal data length of a packet with a priority pp;
- ϵ: technological latency.

3.6.6 Redundancy recovery time

In case of failure, the HRT messages are always delivered to the destination, as they are sent in both directions. The recovery time in this case is zero ($T_{recovery} = 0$).

For other types of messages, $T_{recovery}$ is equal to the sum of: (i) detection time $T_{detection}$, which is the maximum time needed to the neighbors of the faulty node or link to be aware of failure; (ii) the delivery times of control messages for fault declaration T_{decl} and routing tables update T_{tab-up}; (iii) the blocking delay due to low priority messages in each crossed T-AeroRing T_{delay}. Therefore, the recovery time is as follows:

$$T_{recovery} = T_{detection} + T_{decl} + T_{tab-up} + T_{delay} \tag{3.5}$$

where:

- $T_{detection}$ is the local fault detection time and is computed in 3.6.5;
- T_{decl}: is the transmission time of one report control message of minimum size (64 bytes + 20 bytes) for fault detection;
- $T_{tab-up} = \frac{L_{adr-list} \times 8}{R}$ where $L_{add-list}$ is the length of the control message containing the list of MAC addresses, used to update the routing table and is equal to $42 + max(42, 2 + 6 \times (M-3))$ bytes, where 42 bytes is the overhead of the Ethernet header with the 802.1q

tag including 12 bytes for the IFG, 2 bytes to identify the message type, and $(M-3)\cdot 6$ bytes is the size of an Ethernet MAC address multiplied by the maximum number of crossed nodes, i.e., all the nodes apart the failed one and the two detecting the failure.

- $T_{delay} = (M-3) \times (\frac{\max_{pp>0} L_{pp} \times 8}{R} + \epsilon)$.

3.7 Conclusions

To overcome the identified limitations of existing RTE networks supporting the ring topology, we have introduced in this chapter AeroRing, which has the following advantages::

- Improving the resource efficiency due to its QoS-aware routing algorithm: messages are sent on both directions, i.e., high reliability, or only on the shortest path, i.e., high timeliness, according to their priority level;
- Offering high availability and reliability levels through a dynamic redundancy protocol coping with the limitations of existing solutions;
- Minimizing the implementation costs due to its compatibility with IEEE 802.3 standard and the configuration effort through its auto-configuration mechanisms.

In the next chapter, we investigate the real-time performance of AeroRing to prove its predictability.

Chapter 4

Performance Evaluation of AeroRing

Contents

In this chapter, we start by detailing the main system assumptions and model. Then, we present the main iterative conventional Network Calculus approaches to compute the end-to-end delay bounds for networks with cyclic dependencies, and we show through a test case their limitations, in terms of network scalability (number of interconnected nodes) and resource efficiency (network utilization rate). Afterwards, we introduce a new global analysis approach, Pay Multiplexing Only at Convergence points (PMOC), to enable the computation of tighter end-to-end delay bounds. Extensive analyses of the proposed approach are conducted, regarding the delay bound tightness and its impact on the system performance, in comparison to conventional methods and an achievable worst-case delay lower bound. Finally, we extend this new approach to the multiple-ring case. A background on the Network Calculus is given in Appendix B.

4.1 System Model

We are interested in computing an upper bound on Worst-Case Delay for a flow of interest *f.o.i* in ring networks with cyclic dependencies. To conduct such a timing analysis, we consider the following assumptions and notations, using upper indices to indicate nodes or a set of nodes, and lower indices to indicate flows:

Table 4.1: Notations

M	Number of nodes in the network
I	Set of flows served within the network
$i \oplus k$	k^{th} node downstream from node i
$i \ominus k$	k^{th} node upstream from node i
$i \ni k$	Flow i crossing the node k
$\mathbb{P}_i(n)$	Subpath of flow i from its source through n hops, $n \leq h_i$
$conv(i,f,n)$	the convergence points of the *f.o.i* f with the interfering flow i along its subpath of length n
h_i	Number of hops within $\mathbb{P}_i$
$\mathbb{K}_f(n)$	Set of interfering flows with flow f along $\mathbb{P}_f(n)$
$\overline{\mathbb{K}_f(n)}$	Transformed $\mathbb{K}_f(n)$ when cutting virtually the cycles
$Mft(i,f,n)$	First multiplexing node label of flows i and f along $\mathbb{P}_f(n)$
$Mlt(i,f,n)$	Last multiplexing node label of flows i and f along $\mathbb{P}_f(n)$
$\beta^k(t)$	Service curve guaranteed within node k
$\alpha_i^0(t)$	Input arrival curve of flow i at its initial source
$\alpha_i^{k\ominus 1}(t)$	Input arrival curve of flow i at node k along its path
A_i^k	Cumulative Arrival Function (CAF) for the flow i at the node k
D_i^k	Cumulative Departure Function (CDF) for the flow i at the node k
R^k	Service rate of node k
T^k	Service latency of node k
D^j	is the delay within the node j
$\sigma_i^{k\ominus 1}$	Maximum input burst of flow i at node k
ρ_i	Maximum rate of flow i

- We consider a unidirectional ring topology, as shown in Fig. 4.1, connecting M nodes, labelled from 1 to M, and serving a fixed set of flows I. The unidirectional topology is not restrictive, since a full-duplex ring can be considered as two independent unidirectional rings that can be analyzed separately;

- Each flow $i \in I$ follows a fixed path from its initial source until the final sink, defined as $\mathbb{P}_i = (0, i.ft, i.ft \oplus 1, ..., i.ft \oplus (h_i - 1))$, where 0 is a virtual node representing the source, $i.ft$ the first hop and h_i the number of hops of flow i with $h_i \leq M$ and the notations $l \oplus k$ and $l \ominus k$ designate the $k-th$ node downstream and upstream from node l, respectively, where the first downstream node for node M is node 1 and the first upstream node for node 1 is node M. For a flow i, the specific case $i.ft \ominus 1$ is the virtual node 0. Moreover, we define its subpath through $n \in [1, h_i]$ hops as $\mathbb{P}_i(n) = (0, i.ft, ..., i.ft \oplus (n-1))$, i.e., $\mathbb{P}_i = \mathbb{P}_i(h_i)$. It is worth noting that we consider only the output port of crossed nodes within the subpath $\mathbb{P}_i(n)$. Moreover, we assume that no two flows have the same path, since we can aggregate such flows (if any) and thus consider the aggregate flow;

- Within the network, flows are treated according to an aggregate scheduling, i.e. flows are classified within aggregates according to a common parameter, such as priority. Within an aggregate, flows are served under arbitrary multiplexing in each crossed node;

- We denote $i \ni k$ the set of flows crossing the node k, i.e., $i \ni k = \{i \in I \mid k \in \mathbb{P}_i\}$;

- Consider $\mathbb{K}_f(n)$ the set of interfering flows with a *f.o.i.* f along its subpath $\mathbb{P}_f(n)$; so that $\mathbb{K}_f(n) = \{i \neq f / \exists k \in \mathbb{P}_f(n) / i \ni k\}$. Moreover, for any flow $i \in \mathbb{K}_f(n)$, consider its first (last) multiplexing node label with flow f along the subpath $\mathbb{P}_f(n)$ as $Mft(i, f, n)$ ($Mlt(i, f, n)$);

- Each flow $i \in I$ has the CAF A_i^k and the CDF D_i^k at the node k;

- Each flow $i \in I$ is constrained by one leaky bucket of rate ρ_i and an initial burst σ_i^0 at its input source 0, thus admits an initial input arrival curve $\alpha_i^0(t) = \sigma_i^0 + \rho_i.t$. Moreover, we define its input arrival curve at each crossed node k along its path $\mathbb{P}_i$, as $\alpha_i^{k \ominus 1}(t) = \sigma_i^{k \ominus 1} + \rho_i t$;

- Each node k serves the traffic of an aggregate according to a strict service curve having a rate-latency form, with a rate R^k and a latency T^k, $\beta^k(t) = [R^k(t - T^k)]^+$;

- We consider the case of networks where the following condition is satisfied: for any node $k \in [1, M]$, $\frac{\sum_{i \ni k} \rho_i}{R^k} \leq 1$. This condition is necessary to guarantee finite delay bounds within each crossed node.

4.2 Conventional Analysis Methods and Limitations

One of the major challenges in applying Network Calculus is improving accuracy of performance bounds to avoid over-dimensioning of network resources; thus increasing the integration costs. In the research community, there has been a growing interest in the subject and

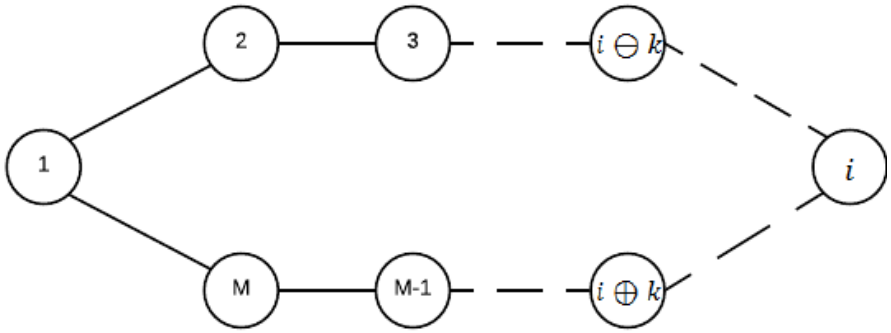


Figure 4.1: Ring-based Network Example

several approaches have been proposed to deal with the delay bounds tightness in networks with acyclic graph, also known as feedforward networks. An interesting overview of the most relevant approaches in this area is detailed in [64]. However, only few approaches related to computing end-to-end delay bounds in non-feedforward networks are reported in the literature, and none of these are dealing with the tightness issue.

A first class of interesting approaches has been proposed to break the potential cycles through prohibiting the use of some links or sub-paths to ensure the feed-forward property [65] [66]. Although these approaches simplify the timing analysis of non-feedforward networks, they imply at the same time a reliability level deterioration, since the use of some links is forbidden, e.g., a ring topology is transformed into line.

The second class of approaches introduces computation methods to support cycles using an iterative approach by successively analyzing the delay bound in each crossed node in the network, resulting in end-to-end delay bounds computation. The most relevant approaches are focusing on, either each crossed node delay bound, e.g., [22] [23] [24], or each crossed node backlog bound, e.g., [25] [20]. For the particular case of ring-based network, two interesting approaches have been proposed: the ***Time Stopping Method*** [22] and the ***Backlog-based Method*** [20].

In this section, we detail the two main conventional iterative analyses of delay bounds, based on Network Calculus. Then, we point out the limitations of each approach through an illustrative example.

4.2.1 Time Stopping Method

This approach has been proposed in [22] and consists of two steps. First, a finite burstiness bound for transmitted flows is assumed to obtain a set of equations to compute the delay bounds. Then, the feasibility conditions to solve these equations are defined. Therefore, we will first express all the equations to compute the upper bounds on bursts and delays in each crossed node. Then, we deduce the feasibility condition.

In [22], the burst propagation formula of a flow i at the output of node j is given by:

$$\sigma_i^j = \sigma_i^{j \ominus 1} + \rho_i * D^j$$

Hence, at the output of node j, flow i has already crossed $(j-i) \mod M$ nodes since node i. The output burst of flow i at the node j is given as follows:

$$\sigma_i^j = \sigma_i^0 + \rho_i * \sum_{k=0}^{(j-i) \mod M} D^{i \oplus k} \tag{4.1}$$

On the other hand, the delay D^k of the node k to process the crossing traffic is equal to the sum of its latency T^k and the processing time of all the crossing bursts:

$$D^k = \frac{\sum_{j \ni k} \sigma_j^{k \ominus 1}}{R^k} + T^k \tag{4.2}$$

Equations (4.1) and (4.2) can be represented by the following matrix system:

$$\begin{cases} D = A_1 * B + C_1 \\ B = A_2 * D + C_2 \end{cases} \tag{4.3}$$

where D is the vector of delays, B is the vector of propagated bursts, and $C1$ and $C2$ are the constant vectors.

Thus, by propagating these constraints, we obtain:

$$D = [I - A_1 * A_2]^{-1} * C_3 \tag{4.4}$$

where $C_3 = A_1 * C_2 + C_1$ and I is the identity matrix.

The system admits a solution if the $[I - A_1 * A_2]$ matrix is invertible, i.e., its determinant is not null. If this condition is verified, the upper bounds on delays can be computed.

The end-to-end delay communication bound of a given flow i with a path $\mathbb{P}_i$ is defined as follows:

$$EED_i = \sum_{k \in \mathbb{P}_i} (D^k + \delta) \tag{4.5}$$

where δ is the propagation delay.

4.2.2 Backlog-based Method

This method has been initially proposed in [25] and more recently generalized in [20]. The authors provide the maximum backlog bound when considering non work-conserving nodes, which is a maximum bound on the total amount of data present in the network at any time. This maximum backlog bound within node k is as follows:

$$Backlog^k = M\frac{\mu}{\eta}(M\sigma^{\max} + B) + \sigma + B \tag{4.6}$$

where:

- $\sigma = \sum_i \sigma_i^0$ is the sum of all flows bursts, and $\sigma^{\max} = \max_k \sum_{j \ni i} \sigma_j^{k \ominus 1}$ is the maximal sum of bursts that pass through any node;

- $\mu = \max_i[\sum_{j \ni i} \rho_j]$;
- $\eta_p = \min_i(R^i - \sum_{j \ni i} \rho_j)$;
- $B = \sum_i R^i.T^i$

The maximum bound on the delay within each node i is the processing time of the maximum backlogged traffic $Backlog$ in Eq. (4.6) served with a transmission capacity R^i, and it is as follows:

$$D^i = \frac{Backlog}{R^i} \tag{4.7}$$

The end-to-end delay communication bound still is computed using Eq. (4.5).

4.2.3 Discussion

In this section, we detail some numerical results of the delay upper bounds of AeroRing based on both conventional methods to point out their limitations. We consider the case study with the following assumptions:

- The topology is a unidirectional ring topology, connecting M nodes;
- The links speed is $R = 1Gbit/s$;
- All equipments are similar and the technological latency within each node is $600ns$;
- Each equipment generates a broadcast traffic with an arrival curve $\alpha \sim (128bytes, 128Kbps)$, with a deadline of $1ms$.

Scenarios are generated varying the flow and network parameters, as follows:

- Network size is varying from 10 to 100 nodes with a step of 10 nodes, i.e., $M \in [10, 100]$;
- Considering the maximum utilization rate $U_{max} \in [10\%, 100\%[$ with a step of 10%, we vary the flow rate according to the following condition: $M.\rho_{\max}/R \leq U_{max}$.

Fig. 4.2 shows a comparison of both approaches when enlarging the network size. Obviously, the delay bounds increase with the network size, since the number of transmitted messages and crossed nodes increases. As we can notice, for a large-scale network, e.g., 100 nodes, both approaches do not guarantee the flows deadline (1ms) and guarantee pessimistic delays, e.g., 33.8ms and 1.6s for Time-Stopping and Backlog-Based methods, respectively. Hence, the maximum network size respecting the flow deadline is about 20 and 27 nodes for the Backlog-Based and Time-Stopping methods, respectively.

Fig. 4.3 illustrates the impact of increasing the congestion on the different methods. Obviously, the delay increases with the network load, since the amount of transmitted data increases, which increases the interferences. As we can see, the Time Stopping method offers tighter bounds until it reaches its limit, i.e., it diverges for $U_{max} = 22.22\%$; whereas the Backlog-Based can achieve a full utilization, even if the delay bounds are overly pessimistic,

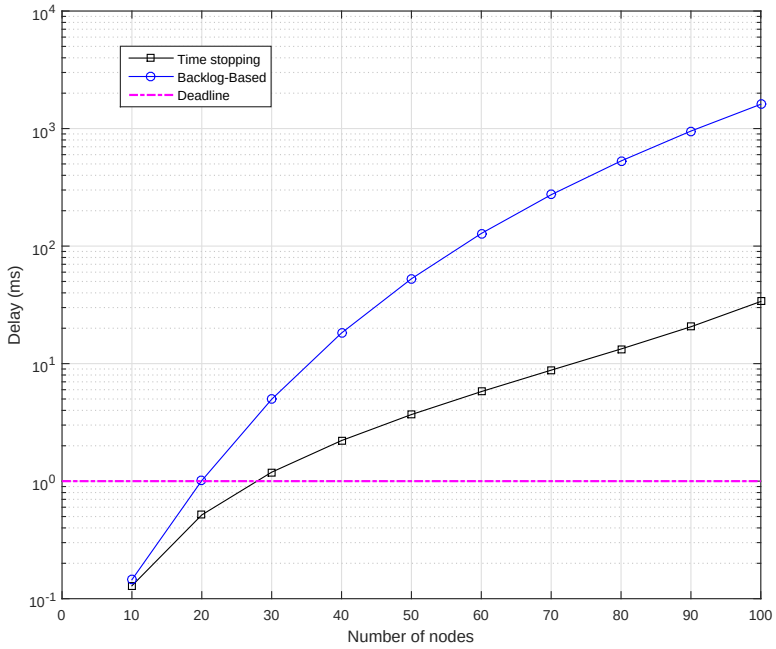


Figure 4.2: End-to-end delay bounds vs number of nodes.

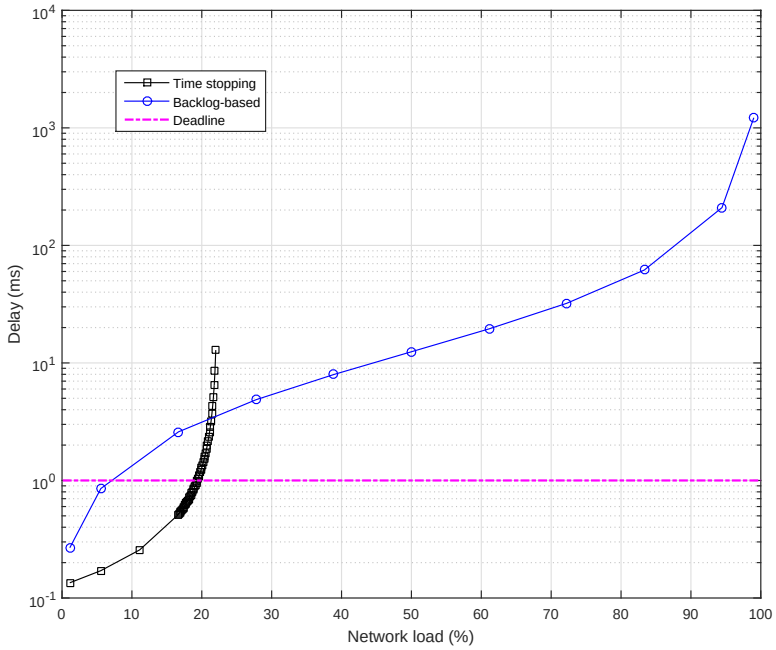


Figure 4.3: End-to-end delay bounds vs network utilization rate.

e.g., 1,22s for $U_{max} = 99\%$. Moreover, the maximum network utilization rate respecting the flows deadline is only about 7.1% and 19.36% with the backlog-based and time-stopping methods, respectively.

These results have the following theoretical explanations. For Time Stopping method, the matrix $[I - A_1 * A_2]$ is as follows:

$$-\frac{1}{C} \times \begin{pmatrix} -C & \rho & 2\rho & \cdots & M\rho \\ M\rho & -C & \rho & \cdots & (M-1)\rho \\ (M-1)\rho & M\rho & -C & \cdots & (M-2)\rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & 2\rho & 3\rho & \cdots & -C \end{pmatrix} \tag{4.8}$$

The system admits a solution if the matrix determinant is not null. In this particular case, the feasibility condition is $\rho < \frac{2*C}{M(M-1)}$. Therefore, the method allows computing bounds when the maximum utilization rate of the network is less than $\frac{2}{(M-1)}$. As we can see in Fig. 4.4, the maximum utilization rate for the Time Stopping method tends to 0, when $M \to \infty$, e.g., less than 0.1 for 20 nodes. This implies that the network has to be under utilized to satisfy the network stability condition, which limits the network resource-efficiency.

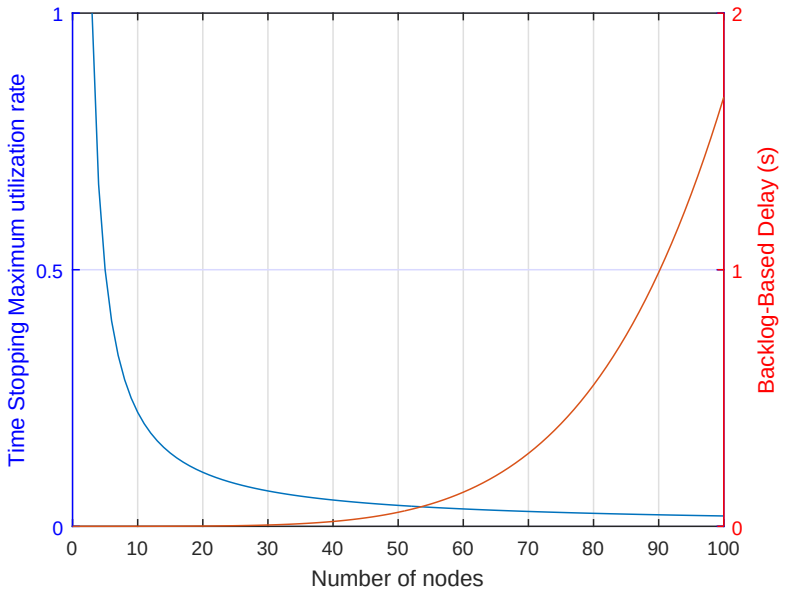


Figure 4.4: The maximum utilization rate for the Time Stopping Method and upper bound on delays for Backlog-Based Method vs number of nodes.

On the other hand, with the Backlog-Based approach the backlog and the end-to-end delay become polynomial functions of the variable M (number of nodes) of degree 3, and 4,

respectively:

$$Backlog = M\frac{\tau}{1-\tau} \cdot (M^2 \times \sigma + M \times L) + M(\sigma + L)$$

$$EDD = M(\frac{Backlog}{C} + \delta)$$

where $\tau = \frac{M \times \rho}{C}$.

This fact implies an end-to-end delay bound growing as $\theta(M^4)$, as shown in Fig 4.4. Hence, as we can notice, the Time Stopping approach offers tighter delay bounds than the Backlog-Based approach when the network is stable, i.e., $U_{max} < \frac{2}{M-1}$. However, the Backlog-Based approach can guarantee a full utilization rate, even if the delay increases dramatically.

The Time Stopping method actually limits the network performance in terms of resource efficiency, i.e., the utilization rate decreases dramatically when the network size increases; whereas the Backlog-based method limits the system scalability, i.e., the nodes number is hardly constrained to guarantee the temporal deadlines.

To overcome these limitations, we introduce in the next section an enhanced worst-case timing analysis of ring-based networks with cyclic dependencies, accounting the flow serialization phenomena along the flows paths.

4.3 Pay Multiplexing Only at Convergence Points

This approach consists in considering the flow serialization phenomena along the path of a *f.o.i*, by paying the bursts of interfering flows only at the convergence points[1]. Similar concepts have been developed in the literature for feedforward networks, i.e., with no cyclic dependencies, such as the Pay Bursts Only Once (PBOO) in [20] and the Pay Multiplexing Only Once (PMOO) in [67] [68]. However, tightening the delay bounds of non-feedforward networks still is an open problem in the literature, and such an approach does not exist yet for non-feedforward networks. The main idea of this method is to handle such an issue for ring-based and general networks.

4.3.1 Illustrative Example

We illustrate herein the cyclic dependency problem and the main idea of PMOC principle through the example of Fig. 4.5.

Consider as a *f.o.i* f_1 with the path $\mathbb{P}_{f_1} = (0,1,2,3)$. To compute the end-to-end delay bound of f_1, we need to integrate the impact of all the interfering flows along its path, $\mathbb{K}_{f_1}(3) = \{f_2, f_3, f_4\}$. Hence, at the input of node 1, we need to quantify the arriving bursts of flows f_3 and f_4. Moreover, the burst of f_4 at the input of node 1 depends on the burst of f_3 at the input of node 4, which in its turn depends on the burst of the *f.o.i* f_1 at the input of node 3. As we can notice, to analyse the impact of interfering flows on the *f.o.i* f_1, we need to quantify its impact on these interfering flows; thus the cyclic dependency. There is actually no start point, where all the flows bursts are known, to launch the delay computation.

[1] In ring-based networks, two flows paths may join at a node, called the convergence point, then disjoin after having a common subpath to maybe join again at another convergence point.

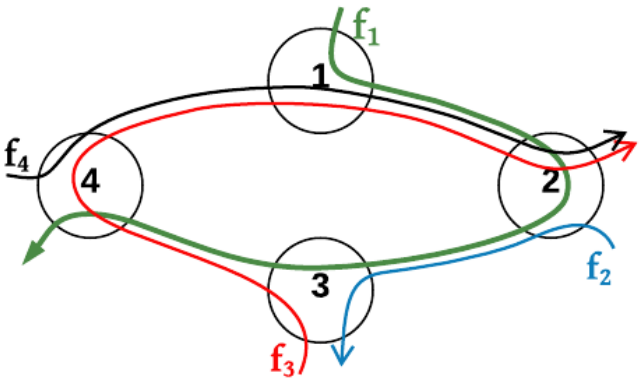


Figure 4.5: A Ring network with cyclic dependency.

To overcome such a difficulty, the main idea of PMOC approach is to compute the tightest possible upper bound on these unknown bursts, when considering the flow serialization phenomena, along the path of the *f.o.i*, and integrating the impact of interfering flows only at the convergence points. As illustrated in Fig. 4.5, because of the ring topology, there are only two possible convergence points with a *f.o.i*:

- If the convergence point is the interfering flow source, then the burst impacting the *f.o.i* is known, e.g., f_2 burst in node 2;
- If the convergence point is the source of the *f.o.i*, then the burst impacting the *f.o.i* is unknown, e.g., f_3 and f_4 bursts in node 1.

Let's consider the example of computing the unknown burst of f_4 at the input of node 1. To compute such a propagated burst, we need to quantify the minimum guaranteed service of f_4 until reaching the input of its convergence point with the *f.o.i* f_1, i.e., the service along $\mathbb{P}_{f_4}(1) = (0,4)$. However, this service depends on the burst of f_3 at the input of node 4, which depends in its turn on the minimum guaranteed service of f_3 until reaching the input of node 4, i.e., the service along $\mathbb{P}_{f_3}(1) = (0,3)$. Detailing such dependencies for all the flows crossing the network reveals actually the need to quantify the service curve guaranteed to each flow f along each of its subpaths, i.e., the service along $\mathbb{P}_f(n)$ for $\forall n \leq h$.

Expressing the service curves and the propagated bursts, for any flow along any of its subpaths, will define a system of linear equations. The latter can be solved using matrices, when a necessary and sufficient condition on the flows rates is verified. These different steps of our proposed PMOC approach, to compute the delay upper bounds, will be detailed in Sections 4.3.2 and 4.3.3, and illustrated for a special case of ring networks in Section 4.3.4.

4.3.2 Service Curve for a Flow of Interest

We focus herein on the first step of the PMOC approach, which consists in defining the guaranteed service curve for a *f.o.i* along any of its subpaths in a ring network. We first present such a curve under arbitrary multiplexing within the crossed nodes in Th. 1. Afterwards, we extend this result to Fixed Priority (FP) multiplexing in Corollary 1.

Theorem 1. *(Service Curve in Ring Networks under Arbitrary Multiplexing) The service curve offered to a flow f along its subpath, $\mathbb{P}_f(n)$, in a ring network under arbitrary multiplexing with strict service curve nodes of the rate-latency form $\beta_{R,T}$ and leaky bucket constrained arrival curves $\alpha_{\sigma,\rho}$, is a rate-latency curve, with a rate $R^{\mathbb{P}_f(n)}$ and a latency $T^{\mathbb{P}_f(n)}$, defined as follows:*

$$R^{\mathbb{P}_f(n)} = \min_{k \in \mathbb{P}_f(n)} [R^k - \sum_{i \ni k, i \neq f} \rho_i] \tag{4.9a}$$

$$T^{\mathbb{P}_f(n)} = \sum_{k \in \mathbb{P}_f(n)} T^k + \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^0 . 1_{\{f \ni i . f t\}} + \rho_i \cdot \sum_{k \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^k}{R^{\mathbb{P}_f(n)}}$$
$$+ \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^{f.ft \ominus 1} . 1_{\{i \ni f.ft / i.ft \neq f.ft\}}}{R^{\mathbb{P}_f(n)}} \tag{4.9b}$$

where $1_{\{cdt\}}$ is equal to 1 if cdt is true and zero otherwise.

The proof of Th. 1 is provided in Appendix C.1. As shown in Eq. (4.9b), some flow bursts are payed twice. These particular flows have actually two convergence points with the *f.o.i*: their own source and the *f.o.i* source; thus respecting the principle of the PMOC approach introduced in Section 4.3.1.

Let us detail the end-to-end service curve of the f.o.i f_1 in the example of Fig. 4.5, when the assumptions of the system model detailed in Section 4.1 are fulfilled, and all the crossed nodes offer the same service curve $\beta_{R,T}$. According to Th. 1, this service curve is a rate-latency curve, with a rate $R^{\mathbb{P}_{f_1}(3)} = \min[R - \rho_3 - \rho_4, R - \rho_2, R - \rho_3]$ and a latency $T^{\mathbb{P}_{f_1}(3)} = 3.T + \frac{1}{R^{\mathbb{P}_{f_1}(3)}}.(\sigma_2^0 + \rho_2.T + \sigma_3^0 + \rho_3.(2.T) + \rho_4.T) + \frac{1}{R^{\mathbb{P}_{f_1}(3)}}.(\sigma_3^4 + \sigma_4^4)$.

To extend such a result to the case of FP multiplexing, we need to introduce the following terms:

- $PL(i)$ for the priority level of flow i, where each crossed node has at maximum NP priority levels and 0 denotes the highest one;
- $L_{max}(i)$ for the maximum packet length of flow i, accounting the communication protocol overhead;
- $hp_f^k = \{i \neq f / i \ni k, PL(i) \leq PL(f)\}$ for the set of flows crossing the node k excluding the *f.o.i* f, with priority higher or equal to the f one;
- $lp_f^k = \{i \ni k, PL(i) \geq PL(f)\}$ for the set of flows crossing the node k with priority lower or equal to the f one;
- $\mathbb{K}_{\leq f}(n) = \{i \neq f / \exists k \in \mathbb{P}_f(n) / i \ni k, PL(i) \leq PL(f)\}$ for the set of flows interfering with the *f.o.i* f along its subpath, $\mathbb{P}_f(n)$, with a priority higher or equal to f one.

It is worth noting that the worst-case behavior under FP multiplexing is covered under Arbitrary multiplexing, but the latter may infer overly-pessimistic bounds since it does not

take into account the priority impact, i.e., any flow may be delayed by all the other flows independently from their priorities. Hence, to overcome such limitations, we define the guaranteed service curve for a *f.o.i* in ring a network, under FP multiplexing, in Corollary 1.

Corollary 1. *(Service Curve in Ring Networks under FP Multiplexing) The service curve offered to a flow of interest f along its subpath, $\mathbb{P}_f(n)$, in a ring network under FP multiplexing with strict service curve nodes of the rate-latency type $\beta_{R,T}$ and leaky bucket constrained arrival curves $\alpha_{\sigma,\rho}$, is a rate-latency curve, with a rate $R^{\mathbb{P}_f(n)}$ and a latency $T^{\mathbb{P}_f(n)}$, defined as follows:*

$$\begin{aligned} R^{\mathbb{P}_f(n)} &= \min_{k\in\mathbb{P}_f(n)}[R^k - \sum_{i\ni hp_f^k}\rho_i] \\ T^{\mathbb{P}_f(n)} &= \sum_{k\in\mathbb{P}_f(n)}(T^k + \frac{\max_{i\in lp_f^k} L_{max}(i)}{R^k}) \\ &+ \sum_{i\in\mathbb{K}_{\leq f}(n)} \frac{\sigma_i^0.1_{\{f\ni i.ft\}} + \rho_i.\sum_{k\in\mathbb{P}_f(n)\cap\mathbb{P}_i}(T^k + \frac{\max_{j\in lp_f^k} L_{max}(j)}{R^k})}{R^{\mathbb{P}_f(n)}} \\ &+ \sum_{i\in\mathbb{K}_{\leq f}(n)} \frac{\sigma_i^{f.ft\ominus 1}.1_{\{i\ni f.ft/i.ft\neq f.ft\}}}{R^{\mathbb{P}_f(n)}} \end{aligned} \tag{4.10}$$

Proof. The proof is straightforward following the Theorem 1. Under FP multiplexing, within each crossed node, a *f.o.i* f is selected for transmission only if all flows with higher or equal priority are already transmitted. Furthermore, since the transmission is non-preemptive, f may be blocked at the worst-case during the transmission time of one maximum packet length with a lower priority level.

Hence, we start by accounting only the impact of lower priority flows on the *f.o.i*, due to the non-preemptive transmission. The left-over service curve of each crossed node under FP is computed in this case through the application of Cor. 6. The obtained service curve is a strict service curve and still has a rate-latency form, with a rate R^k and a latency $\frac{\max_{j\in lp_f^k} L_{max}(j)}{R^k} + T^k$ for each crossed node k. Afterwards, we need to consider only the impact of higher or equal priority flows in $\mathbb{K}_{\leq f}(n)$ when applying Th. 1, to infer the guaranteed service curve of the *f.o.i* f. □

Let us detail the end-to-end service curve of the f.o.i f_1 in the example of Fig. 4.5. Consider that all the crossed nodes implement FP multiplexing with two priority levels and offer the same service curve $\beta_{R,T}$. Moreover, the flows f_1 and f_3 have the highest priority, whereas f_2 and f_4 have the lowest one. According to Cor. 1, this service curve is a rate-latency curve, with a rate $R^{\mathbb{P}_{f_1}(3)} = \min[R-\rho_3, R, R-\rho_3]$ and a latency $T^{\mathbb{P}_{f_1}(3)} = 3.T + L_{max}(4)/R + L_{max}(2)/R + \frac{1}{R^{\mathbb{P}_{f_1}(3)}}.(\sigma_3^0 + \rho_3.(2.T + L_{max}(4)/R)) + \frac{1}{R^{\mathbb{P}_{f_1}(3)}}.\sigma_3^4$.

4.3.3 Computation of the Delay Upper Bound

Now that we have expressed the service curve guarantees for each *f.o.i* along any of its subpaths, we can move to the second step of the PMOC approach, which consists in computing the delay bounds. We put down all the system constraints in a ring network under arbitrary multiplexing, which depend on some variables, i.e., propagated bursts and the offered services:

- **Service Curve Constraint**
  $\forall f \in I$, $\forall n \leq h$, for any $]s,t]$, according to Th. 1,

$$D_f^{f.ft\oplus(n-1)}(t) - A_f^{f.ft}(s) \leq \beta_{R^{\mathbb{P}_f(n)},T^{\mathbb{P}_f(n)}}(t-s)$$

- **Output Arrival Curve Constraint**
  $\forall f \in I$, $\forall n \leq h$, according to Th. 3, in Appendix B

$$\alpha_f^{f.ft\oplus(n-1)}(t) = \alpha^0 \oslash \beta_{R^{\mathbb{P}_f(n)},T^{\mathbb{P}_f(n)}}(t)$$

- **Delay bound**
  $\forall f \in I$, $\forall n \leq h$, according to Th. 3, in Appendix B

$$EED_f^{\mathbb{P}_f(n)} = h(\alpha^0, \beta_{R^{\mathbb{P}_f(n)},T^{\mathbb{P}_f(n)}})$$

In the case of rate-latency service curves and leaky-bucket arrival curves, these system constraints are linear and can be replaced with the following set (*):

- **Service Curve Constraint**
  $\forall f \in I$, $\forall n \leq h$, for any $]s,t]$,

$$R^{\mathbb{P}_f(n)} = \min_{k\in\mathbb{P}_f(n)}[R^k - \sum_{i\ni k, i\neq f}\rho_i]$$

$$T^{\mathbb{P}_f(n)} = \sum_{k\in\mathbb{P}_f(n)} T^k + \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^0.1_{\{f\ni i.ft\}} + \rho_i \cdot \sum_{k\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^k}{R^{\mathbb{P}_f(n)}} + \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^{f.ft\ominus 1}.1_{\{i\ni f.ft/i.ft\neq f.ft\}}}{R^{\mathbb{P}_f(n)}}$$

- **Output Arrival Curve Constraint**
  $\forall f \in I$, $\forall n \leq h$,

$$\sigma_f^{f.ft\oplus(n-1)} = \sigma_f^0 + \rho_f \times T^{\mathbb{P}_f(n)}$$

- **Delay bound**
  $\forall f \in I$, $\forall n \leq h$,

$$EED_f^{\mathbb{P}_f(n)} = \frac{\sigma_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

Hence, the set (*) can be written in a matrix form as follows:

Service Curve Constraint

$$\overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T} = \overbrace{\begin{bmatrix} c1_{f1} \\ \vdots \\ c1_{fh_f} \\ \vdots \end{bmatrix}}^{C1} + \overbrace{\begin{bmatrix} a1_{f,1} & \cdots & a1_{f,h_f} & \cdots \\ \vdots & \ddots & \ddots & \\ a1_{fh_f,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}}^{A1} \times \overbrace{\begin{bmatrix} \sigma_f^{f.ft\oplus 1} \\ \vdots \\ \sigma_f^{f.ft\oplus(h_f-1)} \\ \vdots \end{bmatrix}}^{\sigma}$$

where T is the vector that holds the latencies of the offered service (Eq. (4.9b)), $A1$ is the matrix of the coefficients of unknown propagated bursts and $C1$ is the vector of constants, i.e, the latencies T^i and initial bursts transmission times, appearing in the service curve constraints of (*).

Output Arrival Curve Constraint

$$\overbrace{\begin{bmatrix} \sigma_f^{f.ft\oplus 1} \\ \vdots \\ \sigma_f^{f.ft\oplus(h_f-1)} \\ \vdots \end{bmatrix}}^{\sigma} = \overbrace{\begin{bmatrix} c2_{f1} \\ \vdots \\ c2_{fh_f} \\ \vdots \end{bmatrix}}^{C2} + \overbrace{\begin{bmatrix} a2_{f,1} & \cdots & a2_{f,h_f} & \cdots \\ \vdots & \ddots & \ddots & \\ a2_{fh_f,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}}^{A2} \times \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

where σ is the vector of the unknown propagated bursts, $A2$ is the matrix of the coefficients of the corresponding unknown offered service latencies, i.e., the flow rate, and $C2$ is the vector of constants, i.e., the initial bursts σ_f^0, appearing in the output arrival curve constraints of (*).

Delay bound

$$\overbrace{\begin{bmatrix} EED^{\mathbb{P}_f(2)} \\ \vdots \\ EED^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{EED} = \overbrace{\begin{bmatrix} c3_{f1} \\ \vdots \\ c3_{fh_f} \\ \vdots \end{bmatrix}}^{C3} + \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

where $C3$ is the vector of constants, i.e., the initial bursts transmission times, appearing in the delay bound constraints of (*).

When propagating the different constraint, this matrix form is transformed to the following ($\mathbb{M}^*$):

$$\begin{cases} (Id - A1 \times A2) \times T = C1 + A1 \times C2 \\ \qquad\qquad EED = C3 + T \end{cases} \tag{4.11}$$

Based on the matrix form $\mathbb{M}^*$, we deduce in the following corollary a necessary and sufficient condition on the existence of delay upper bounds for each *f.o.i* along any of its subpaths,

in the general case of ring networks under arbitrary multiplexing. This condition will be detailed in the next section for a special case of ring networks.

Corollary 2. *(Delay Bound under Arbitrary Multiplexing) In a ring network under arbitrary multiplexing, the delay upper bound of each f.o.i f along its subpath $\mathbb{P}_f(n)$ exists and is at most equal to*

$$EED_f^{\mathbb{P}_f(n)} = \frac{\sigma_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

if and only if the matrix $(Id - A1 \times A2)$ in $\mathbb{M}^*$ is invertible, i.e., its determinant is not zero.

Proof. Based on known results in linear algebra, we can see from $\mathbb{M}^*$ that the vector of latencies T exists and is unique, if and only if the square matrix $(Id - A1 \times A2)$ is invertible. Under this necessary and sufficient condition, we have $T = (Id - A_1 \times A_2)^{-1} \times (C1 + A1 \times A2)$. Consequently, $EED = C3 + (Id - A_1 \times A_2)^{-1} \times (C1 + A1 \times A2)$ exists and is unique. This finishes the proof of Corollary 2. □

Such a result can be easily extended under FP multiplexing. We need to order the delay bound calculus according to the decreasing order of priority levels, i.e., computing the delay bounds of the highest priority first. We distinguish the following main steps:

1. For each priority level $p \in [0, NP-1]$, we define the corresponding matrix form $\mathbb{M}^*$, when including only the constraints related to the flows with higher or equal priority, i.e., $\forall f \in I$ with $PL(f) \leq p$. It is worth noting that the impact of lower priority flows is already integrated within the service curve formula, defined in Cor. 1;

2. If the necessary and sufficient condition of Cor. 2 is satisfied, then we compute the delay bounds of all the flows of priority level p along their subpaths;

3. The unknown parameters in $\mathbb{M}^*$ defined for the priority level p, i.e., propagated bursts and service latencies, are updated with the computed values in step 2;

4. If $p < NP - 1$, then back to the step 1 when focusing on the priority level $p \leftarrow p + 1$.

Hence, we have the following corollary concerning the computed delay bounds for each *f.o.i* of priority level p along any of its subpaths, in ring networks under FP multiplexing:

Corollary 3. *(Delay Bound under FP Multiplexing) In a ring network under FP multiplexing, the delay upper bound of each f.o.i f of priority level p along its subpath $\mathbb{P}_f(n)$ exists and is at most equal to*

$$EED_f^{\mathbb{P}_f(n)} = \frac{\sigma_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

if and only if for each priority level pp higher than p, the matrix $(Id - A1 \times A2)$ in $\mathbb{M}^*$ associated to the priority level pp is invertible, i.e., its determinant is not zero.

Proof. The proof is straightforward following the Corollary 2. Moreover, following the main steps of the delay calculus under FP multiplexing, detailed above, we have to verify in step 2 the necessary and sufficient condition of Corollary 2 for each priority level higher than the *f.o.i* priority level. □

4.3.4 Special Case: Regular Ring Networks

We introduce herein a particular case of ring networks, called regular ring networks, for which we deduce a specific necessary and sufficient condition for the existence of delay upper bounds, in comparison to the general one in Cor. 2.

Definition 1. *(Regular Ring Network) A ring network connecting M nodes is a regular ring network with a degree h, where $2 \leq h \leq M$, when it satisfies the following assumptions: (i) all the nodes guarantee the same rate-latency service curve, $\beta_{R,T}$; (ii) each node $l \in [1, M]$ is generating a (σ, ρ)-constrained flow, destined to all its k-th downstream nodes from l, $\forall k \leq h$.*

It is worth noting that a ring network with a broadcast communication pattern is a regular ring network with a degree $h = M$.

We have the following conjecture on the delay bounds in regular ring networks, based on a more specific necessary and sufficient condition than the one in Cor. 2:

Conjecture 1. *(Delay Bound in Regular Ring Networks) In a regular ring network under arbitrary multiplexing and with a degree h, the delay upper bound of each f.o.i f along its subpath $\mathbb{P}_f(n)$ exists and is at most equal to*

$$EED_f^{\mathbb{P}_f(n)} = \frac{\sigma_f^0}{R^{\mathbb{P}_f(n)}} + T^{\mathbb{P}_f(n)}$$

if and only if the following equivalent conditions are verified:
(i) (Flow rate Cdt.) The maximum rate of each generated (σ, ρ)-constrained flow is as follows: $\rho < \frac{R}{2 \cdot (h-1)}$;
(ii) (Utilization rate Cdt.) The maximum utilization rate of the network, $U_{max} = h \cdot \rho / R$, is as follows: $U_{max} < \frac{h}{2 \cdot (h-1)}$. Thus, as $h \to \infty$, the maximum utilization rate tends to 50%.

This conjecture is based on the observation of the behavior of the maximum utilization rate (resp. maximum flow rate), satisfying the necessary and sufficient condition of Cor. 2, for regular ring networks when varying the degree $h \in [2, M]$, as illustrated in Fig. 4.6. We actually have built the associated matrix form $\mathbb{M}^*$ for $h \in [2, 100]$ and $R = 1Gb/s$. Then, based on a symbolic computation tool, we have computed the maximum utilization rate of the network (resp. maximum flow rate), for which the determinant of the matrix $(Id - A1 \times A2)$ in $\mathbb{M}^*$ vanishes. As we can see, The maximum network utilization rate decreases from 100% for $h = 2$ to 50.5% for $h = 100$, while the maximum flow rate decreases from $\frac{R}{2}$ for $h = 2$ to $\frac{R}{198}$ for $h = 100$. These values are coherent with the upper bounds defined in the Conjecture 1, which are $\frac{h}{2 \cdot (h-1)}$ for the maximum network utilization rate and $\frac{R}{2 \cdot (h-1)}$ for the maximum flow rate. It is worth noting that the maximum utilization rate in Conjecture 1 is more restrictive than the one in Section 4.1, i.e., $h \cdot \rho / R \leq 1$.

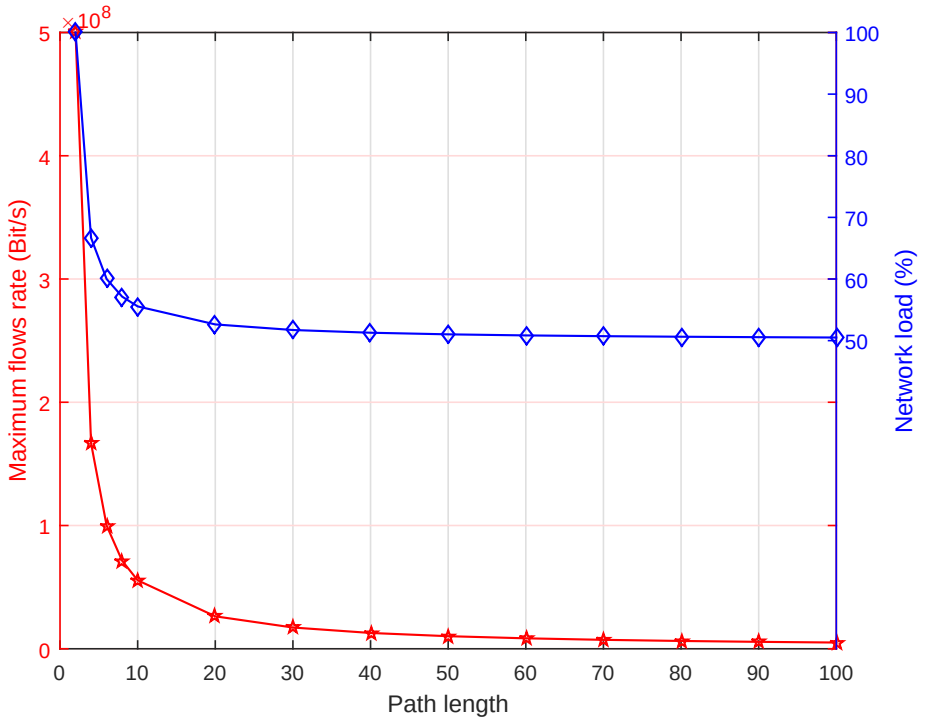


Figure 4.6: Maximum network utilization and flow rate vs network degree, i.e., flow path length, for which the determinant of the matrix $(Id - A1 \times A2)$ in $\mathbb{M}^*$ vanishes

Example

We now explicit the matrix form $\mathbb{M}^*$ and the necessary and sufficient condition on the existence of delay bounds for a simple example. Consider a regular ring network with 3 nodes, labeled from 1 to 3, and a degree $h = 2$ under arbitrary multiplexing, as illustrated in Fig. 4.7. Each node i sends a (σ^0, ρ)-constrained flow f_i and guarantees a service curve $\beta_{R,0}$. The aim is to compute the end-to-end delay bound of the *f.o.i* f_1.

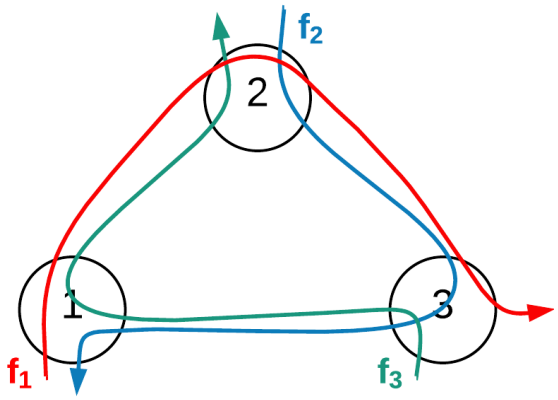


Figure 4.7: Example of a regular ring network with $M = 3$ and $h = 2$

First, we explicit the different parameters of the matrix form $\mathbb{M}^*$ of such a network as follows:

$$T^T = (T^{\mathbb{P}_{f_1(1)}}, T^{\mathbb{P}_{f_1(2)}}, T^{\mathbb{P}_{f_2(1)}}, T^{\mathbb{P}_{f_2(2)}}, T^{\mathbb{P}_{f_3(1)}}, T^{\mathbb{P}_{f_3(2)}})$$

$$C1^T = \frac{\sigma^0}{R-\rho} \cdot (0,1,0,1,0,1)$$

$$A_1 = \frac{1}{R-\rho} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\sigma^T = (\sigma^1_{f_1}, \sigma^2_{f_1}, \sigma^2_{f_2}, \sigma^3_{f_2}, \sigma^3_{f_3}, \sigma^1_{f_3})$$

$C2^T = \sigma^0 \cdot (1,1,1,1,1,1,)$ and $A2 = \rho \cdot I_{(h \times M)}$

Then, to verify the necessary and sufficient condition defined in Cor. 2, we express the determinant of the matrix $(Id - A_1 \times A_2)$, which is as follows:

$$(\rho - \frac{R}{2}) \cdot (-2\rho^2 + 2R\rho - 2R^2)/(R-\rho)^3$$

This function vanishes for the maximum flow rate $\rho = \frac{R}{2}$. This value is coherent with the Conjecture 1, where the upper bound of the maximum flow rate is $< R/2 \cdot (h-1)$, i.e., $R/2$ for $h = 2$. Hence, if the flow rate condition is verified, i.e., $\rho < R/2$, then the end-to-end delay upper bound of the *f.o.i* f_1, $EED_{f_1}^{\mathbb{P}_{f_1}(2)}$, exists and is at most equal to $\frac{\sigma^0}{R^{\mathbb{P}_{f_1}(2)}} + T^{\mathbb{P}_{f_1}(2)}$, where $R^{\mathbb{P}_{f_1}(2)} = R - \rho$ and $T^{\mathbb{P}_{f_1}(2)} = \frac{2\sigma^0}{R-\rho} + \frac{\sigma^0\rho(\rho^2 - R\rho + R^2)}{(R-\rho)(R^3 - 3R^2\rho - 2\rho^3)}$

4.3.5 Performance Evaluation

In this section, we detail some numerical results of the delay upper bounds of a *f.o.i.* in a ring network with cyclic dependencies, under different scenarios, when applying our approach PMOC. First, we describe the considered case study and scenarios. Then, we report the sensitivity analysis of such computed upper bounds with respect to flows burst, rate and path length, for various values of network size M. Finally, we assess their tightness through a lower bound on WCD (Worst-Case Delay) in several scenarios.

Case study and scenarios

We consider the case study with the following assumptions:

- The topology is a unidirectional ring topology, connecting M nodes;
- All nodes guarantee a rate-latency service curve $\beta_{R,T}$ with $R = 1Gbps$ and $T = 600ns$;
- Each node generates one leaky-bucket constrained flow with a burst σ and a rate ρ;
- The considered network is a regular ring network with a degree h.

To analyse the sensitivity of the computed delay bounds and to assess their tightness, we consider various network configurations, where each network configuration is defined with the tuple (σ, ρ, h, M). The main idea is to vary only one parameter of this tuple at a time, to highlight its impact on the computed delay bounds.

Sensitivity analysis

We discuss herein the impact of each network configuration parameter on the delay bounds, computed with the PMOC approach. The numerical results for different scenarios are reported in Figs. 4.8, 4.9 and 4.10.

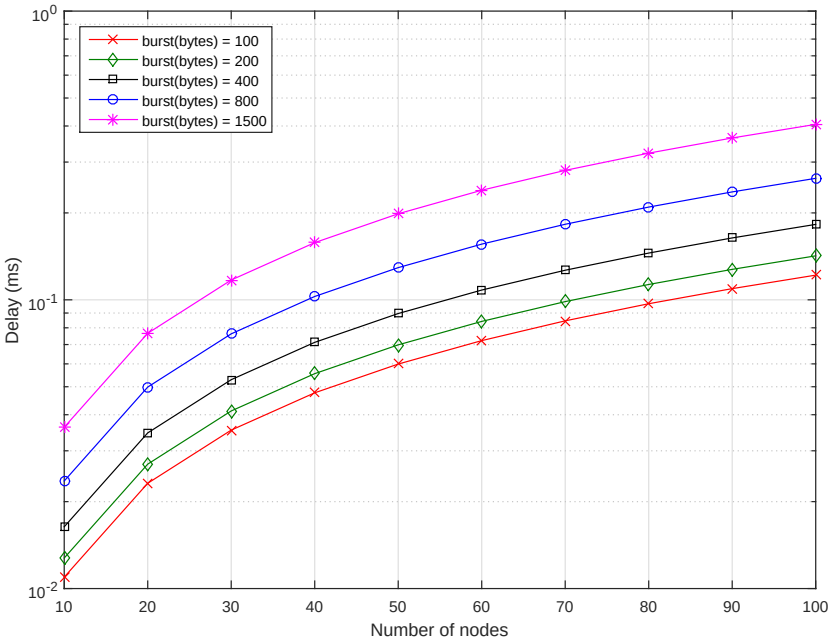


Figure 4.8: The impact of the flow burst on the delay bounds vs network size for ($\sigma \in [100-1500]bytes, \rho = 128Kbps, h = M, M \in [10-100]$).

Fig. 4.8 shows the impact of the burst size on the delay bounds. Obviously, for a fixed network size, the delay increases when increasing the flow burst, since the multiplexing time increases within each crossed node. Moreover, for a fixed flow burst, the delay increases with the network size. There are two main observations to note from this analysis scenario: (i) the delay bound grows logarithmically in terms of flow burst, e.g., for $M = 100$, when the flow burst increases from $100bytes$ to $1500bytes$, i.e., $\times 15$, the delay goes only from $0.12ms$ to $0.4ms$, i.e., $\times 3.3$; (ii) the delay bound for a fixed flow burst increases in a more noticeable way with the network size but still grows linearly, e.g., for $\sigma = 100$bytes, the delay goes from 10^{-2}ms for $M = 10$ nodes to almost 10^{-1}ms for $M = 100$ nodes, i.e., $\times 10$, which is equivalent to the scaling factor of the network. These results infer that the interfering flow bursts have higher impact on the delay bound of a *f.o.i* than its own burst. This fact is very coherent with the delay bound expression, defined in Section 4.3.3.

Fig. 4.9 shows the impact of the flow rate on the delay bounds. As we can notice, there are two distinguishable behaviors of the delay bounds: (i) when the flow rate condition in Conjecture 1 is verified, the delay bounds grow logarithmically in terms of the flow rate, e.g., for $M = 40$, when the rate increases from 1 Mb/s to 9Mb/s, i.e., $\times 9$, the delay bound grows from almost 10^{-2}ms to 3.10^{-2}ms, i.e., $\times 3$; (ii) when this condition is violated, the delay bound tends to infinity, e.g., for $\rho = 8Mb/s$, the delay bound diverges for a network size higher than $M = 63$, which corresponds to the condition $\rho < \frac{R}{2(M-1)} \Leftrightarrow M < \frac{R}{2\rho} + 1 = 63.5$. This fact infers an exponential growth of the delay bounds with the network size, when the flow rate condition

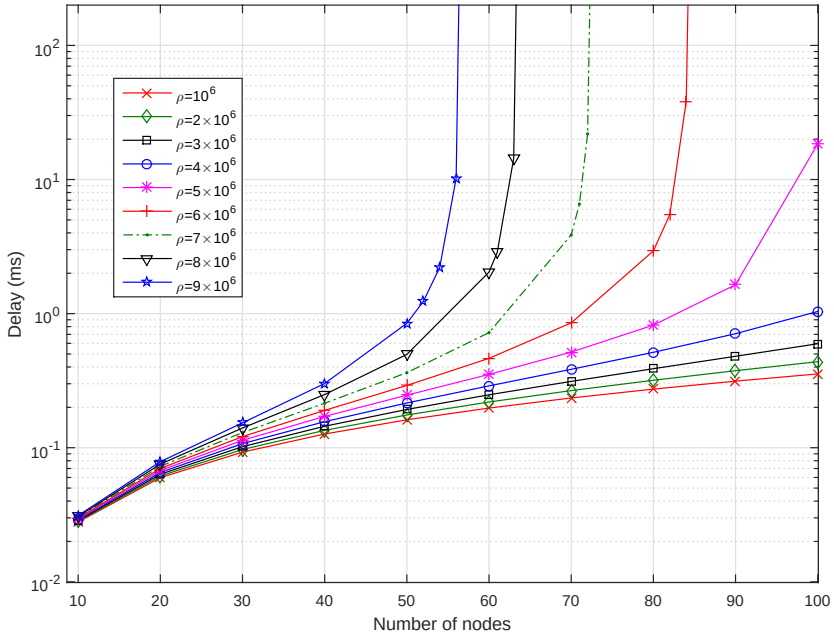


Figure 4.9: The impact of flow rate on the delay bound vs network size for ($\sigma = 128bytes, \rho = [1-9]Mbps, h = M, M \in [10-100]$).

achieves its limit. These results show the inherent impact of the flow rate on the delay bounds with the PMOC approach, which is relevant with our conjecture on the network stability condition of regular ring networks in Section 4.3.4.

Fig. 4.10, shows the impact of the flow path length on the delay bounds. As it is shown, the delay bound has similar behavior in terms of flow path length than its rate, i.e., grows logarithmically when the flow rate condition is verified. Increasing the flow path length induces a higher number of interfering flows along the path; thus a higher service latency and lower service rate according to the PMOC approach. Moreover, it is worth noting that the delay bounds for regular ring networks depend only on the network degree h, i.e., flow path length. For instance, the delay bound is $0.79ms$ for $h = 20$ independently from the network size. This result is coherent with Conjecture 1.

These results show that the delay bounds computed with the PMOC approach are particularly sensitive to the flow rate and path length. This fact is mainly due to the conditions defined in Conjecture 1, which depend on both parameters and infer an exponential behavior of the delay bounds when the conditions achieve their limit.

Tightness analysis

To investigate the tightness of our approach, we compare the delay bounds obtained with our proposed method to an achievable worst-case delay, denoted as WCD lower bound. The latter is computed when considering an intuitive worst-case scenario, which consists in integrating

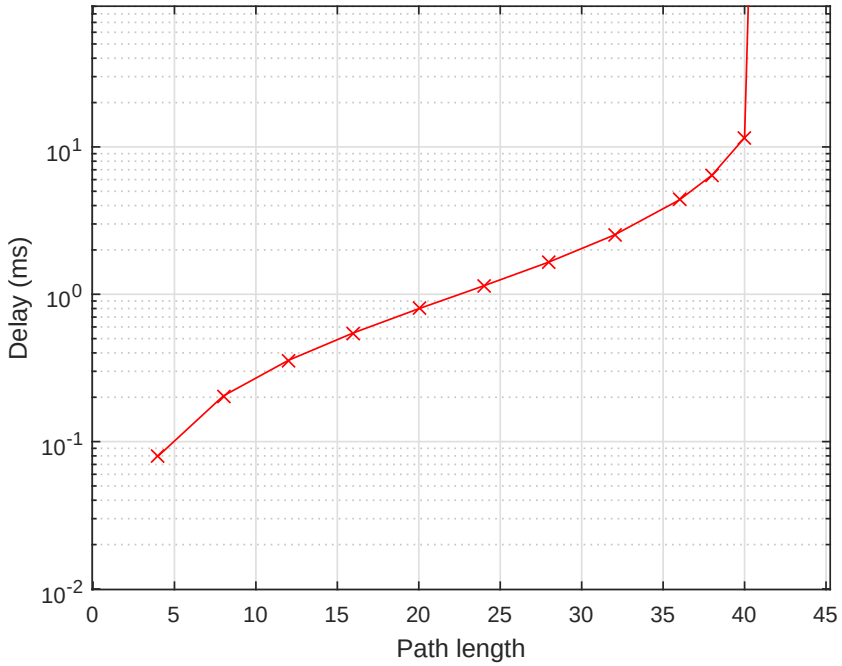


Figure 4.10: The impact of the flow path on delay bound for ($\sigma = 1500bytes, \rho = 12Mbps, h \in [4-45], \forall M > h$).

for each flow of interest only the impact of downstream flows interferences within each crossed node, and ignoring the impact of the upstream flows at its source node, i.e., this is the unknown variable due to cyclic dependency and it is considered as null for this intuitive WCD. The size of the interval between the computed upper delay bounds and WCD lower bounds will give us an idea about the delay bound tightness, i.e., this interval includes the exact worst-case delay; thus if this interval duration is small, then the upper bound delay is tight.

Figs. 4.11, 4.12 and 4.13 report the numerical results of different analysis scenarios, conducted to assess the delay bounds tightness. As we can notice, the gap between the delay bound computed with the PMOC approach and the WCD lower bound still is bounded and relatively small, e.g., up to 0.5ms, when varying the flow burst (Fig. 4.11), and also in terms of network utilization rate and flow path length, when the network stability condition is verified, i.e., $U_{max} < \frac{h}{2(h-1)} = 52.6\%$ in this scenario. For instance, the network utilization rate condition is still verified when the network utilization rate is up to 50% and the network size is up to 80 nodes, as illustrated in Figs. 4.12 and 4.13.

However, when the network utilization rate condition is violated, we can not conclude on the delay bound tightness since it tends to infinity.

These results show that: if the network utilization rate condition is verified, then the delay bounds computed with the PMOC approach are noticeably accurate, i.e., less than 0.5***ms of pessimism, when varying different network and flow parameters.***

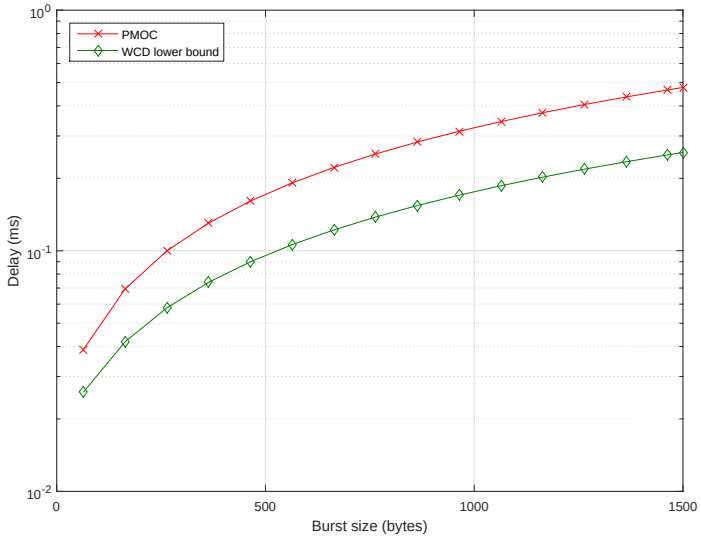


Figure 4.11: Impact of the burst on delay bound tightness for ($\sigma = [64-1500]bytes, \rho = 128Kbps, h = M, M = 20$).

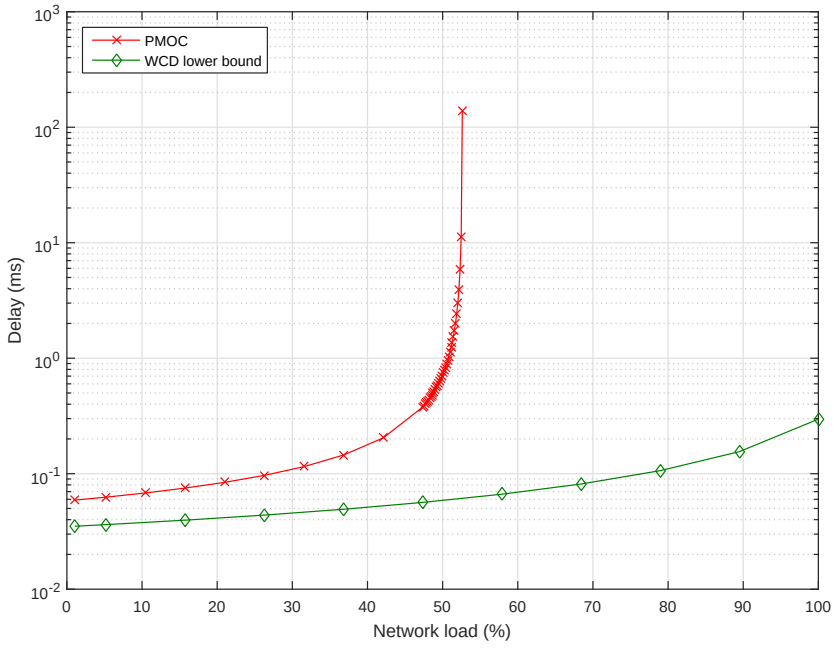


Figure 4.12: Impact of the maximum network utilization rate on delay bound tightness for ($\sigma = 128bytes, \rho = [0.5-50]Mbps, h = M, M = 20$).

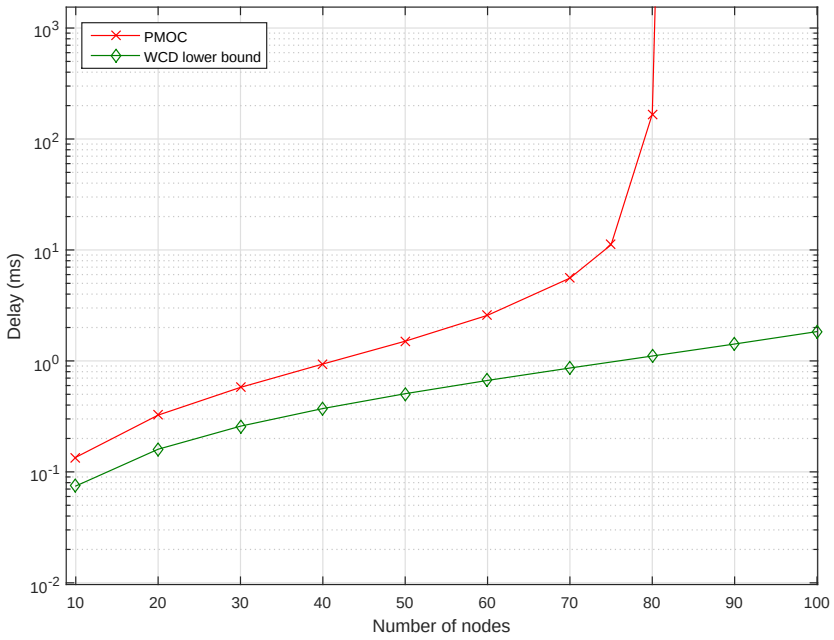


Figure 4.13: Impact of network size on delay bound tightness for ($\sigma = 787bytes, \rho = 6.3Mbps, h = M, M \in [10-100]$).

Comparison with the Related Work

In order to benchmark the delay bounds obtained with the PMOC approach against the existing ones, i.e., Time Stopping and Backlog-based, we consider the same case of study and scenario detailed in Section 4.2.3.

Fig. 4.14 shows a comparison of the different approaches when enlarging the network size. As we can notice, the PMOC approach offers tighter delay bounds for large-scale networks, while guaranteeing the flows deadline, in comparison with the conventional methods, e.g., the PMOC delay is 0.3ms compared to 33.8ms and 1.6s for Time-Stopping and Backlog-Based methods for a network of 100 nodes. Hence, the maximum network size respecting the flow deadline is about 20 and 27 nodes with the Backlog-Based and Time Stopping methods, respectively, whereas it achieves 100 nodes with PMOC approach.

Fig. 4.15 illustrates the impact of increasing the congestion on the different methods. As we can see, the Time Stopping method diverges for a global utilization rate around 22.22%, which corresponds to $\frac{2}{M-1}$; whereas it achieves 55.55% with our proposed approach, which corresponds to $\frac{M}{2(M-1)}$. However, a full utilization rate is still achievable under the Backlog-Based method, even if the delay bounds are overly pessimistic, e.g., 1,22s for $U_{max} = 99\%$. Furthermore, the maximum network utilization rate respecting the flows deadline is only about 7.1% and 19.36% with the Backlog-Based and Time Stopping methods, respectively, compared to 54.6% with PMOC.

This comparative analysis shows that using PMOC approach yields enhanced network

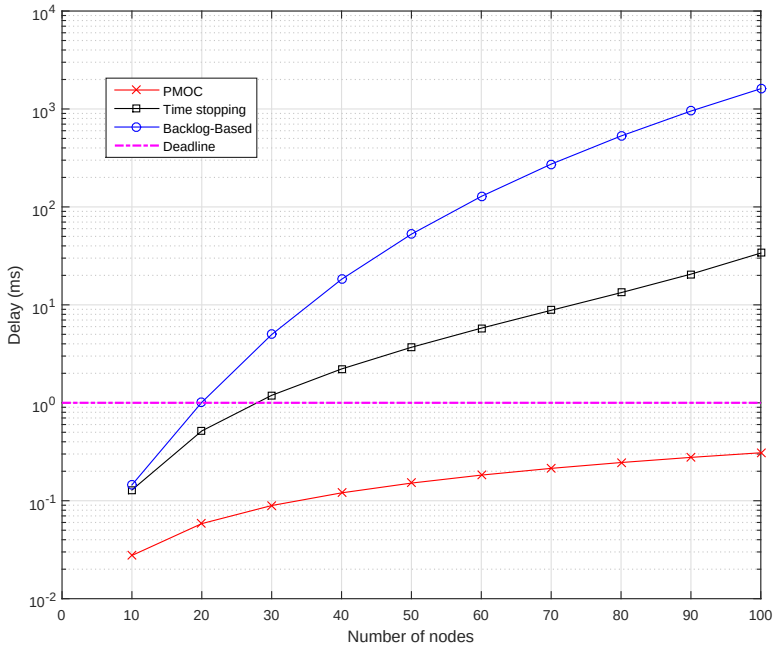


Figure 4.14: End-to-end delay bounds vs number of nodes for ($\sigma = 128bytes, \rho = 128Kbps, h = M, M \in [10-100]$).

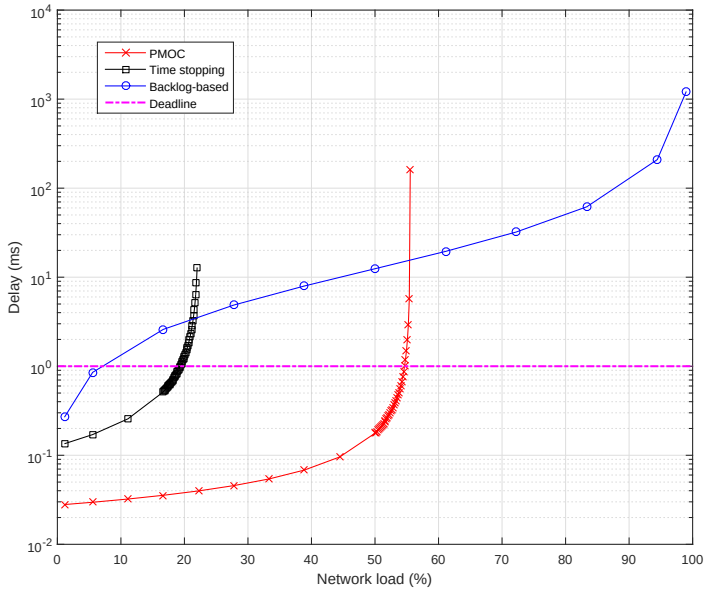


Figure 4.15: End-to-end delay bounds vs network utilization rate for ($\sigma = 128bytes, \rho \in [1-100]Mbps, h = M, M = 10$).

performance, in terms of resource efficiency and network scalability, in comparison with the conventional timing analyses.

4.4 Generalization of PMOC for Multiple Ring Networks

We detail in this section the generalization of the PMOC approach to be applied for the multiple ring networks. First, we define the guaranteed service curve for a *f.o.i* along any of its subpaths under arbitrary multiplexing in Cor. 4 for such a topology. Then, we extend this result to Fixed Priority (FP) multiplexing in Cor. 5. Afterwards, we analyse the sensitivity of the derived delay bounds with respect to several network and flows parameters, in comparison with the mono-ring topology.

4.4.1 Service Curve for a Flow of Interest

Corollary 4. *(Service Curve under Arbitrary Multiplexing) The service curve offered to a flow f along its subpath, $\mathbb{P}_f(n)$, in a multiple ring network under arbitrary multiplexing with strict service curve nodes of the rate-latency form $\beta_{R,T}$ and leaky bucket constrained arrival curves $\alpha_{\sigma,\rho}$, is a rate-latency curve, with a rate $R^{\mathbb{P}_f(n)}$ and a latency $T^{\mathbb{P}_f(n)}$, defined as follows:*

$$R^{\mathbb{P}_f(n)} = \min_{k \in \mathbb{P}_f(n)} [R^k - \sum_{i \ni k, i \neq f} \rho_i] \tag{4.12a}$$

$$T^{\mathbb{P}_f(n)} = \sum_{k \in \mathbb{P}_f(n)} T^k + \sum_{i \in \mathbb{K}_f(n)} \frac{\sum_{k \in conv(i,f,n)} \sigma_i^{k \ominus 1} + \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j}{R^{\mathbb{P}_f(n)}} \tag{4.12b}$$

The proof of Cor. 4 is provided in Appendix C.2. Afterwards, we extend such a result to the FP multiplexing case, based on the same notations presented in 4.3.2.

Corollary 5. *(Service Curve under FP Multiplexing) The service curve offered to a flow of interest f along its subpath, $\mathbb{P}_f(n)$, in a multiple ring network under FP multiplexing with strict service curve nodes of the rate-latency type $\beta_{R,T}$ and leaky bucket constrained arrival curves $\alpha_{\sigma,\rho}$, is a rate-latency curve, with a rate $R^{\mathbb{P}_f(n)}$ and a latency $T^{\mathbb{P}_f(n)}$, defined as follows:*

$$\begin{aligned} R^{\mathbb{P}_f(n)} &= \min_{k \in \mathbb{P}_f(n)} [R^k - \sum_{i \ni hp_f^k} \rho_i] \\ T^{\mathbb{P}_f(n)} &= \sum_{k \in \mathbb{P}_f(n)} (T^k + \frac{\max_{i \in lp_f^k} L_{max}(i)}{R^k}) \\ &+ \sum_{i \in \mathbb{K}_{\leq f}(n)} \frac{\sum_{k \in conv(i,f,n)} \sigma_i^{k \ominus 1} + \rho_i \cdot \sum_{k \in \mathbb{P}_f(n) \cap \mathbb{P}_i} (T^k + \frac{\max_{j \in lp_f^k} L_{max}(j)}{R^k})}{R^{\mathbb{P}_f(n)}} \end{aligned} \tag{4.13}$$

Proof. The proof of Cor. 5 is based on the same idea than Cor. 1. □

It is worth noting that the second step of the PMOC approach, which consists in computing the delay bounds, remains the same as explained in Section 4.3.3.

4.4.2 Performance Evaluation

In this section, we investigate the offered timing performance of a multiple ring topology compared to the mono-ring one, with respect to the inter-ring communication load *interNet* and the number of rings *nbR*, to show their impact on the performances.

Hence, we study the offered end-to-end delay bounds under different configurations according to the set of parameters $(interNet, M, nbR, \sigma, \rho)$.

We consider the case study with the following assumptions:

- The network is based on a mono or multiple ring topology with nbR rings, connecting M nodes, i.e., each ring connects $\frac{M}{nbR}$ nodes;
- The links speed is $R = 1Gbps$;
- Technological latency within each node is $600ns$;
- Each node generates one leaky-bucket constrained flow with a burst σ and a rate ρ.
- Communications within the same ring are broadcast.

Fig. 4.16 shows the impact of the inter-ring communication load and the number of rings on the end-to-end delays. As we can see, the multiple-ring network is more sensitive to the inter-ring communication load when the number of rings increases, e.g., the 12-rings network offers the best delay bounds for an inter-ring communication load less than 34.8%, whereas, it guarantees the highest delay bounds for a load higher than 59%. This phenomena is observed under two conditions:

1. First, it is worth noting that the number of convergence points increases with the number of rings. Hence, the more this parameter increases, the more the delay bounds may increase;
2. Second, increasing the inter-ring communication load leads to a higher impact of interfering flow at each convergence point.

As we can notice, the 12-rings topology satisfies both conditions for an inter-ring communication load higher than 59%.

Fig. 4.17 shows the impact of the network size and the number of rings on the end-to-end delay bounds. As it is shown, the delay bounds are generally decreasing when increasing the number of rings. This is mainly due to the decreasing flow path length. In the worst case for the multiple-ring case, a flow needs to cross the source local ring, the backbone ring and the destination local ring to reach its destination. Hence, the path length is equal to $M_{cross} = \frac{2M}{nbR} + nbR + 1 < M$ when $M > 12$ and $nbR > 2$. Moreover, the delay bounds for the 12-rings topology increases dramatically for a network size higher than 60 nodes. This is mainly related to the increasing inter-ring communication load, due to the increasing network size, which leads to a higher impact of interfering flows at each convergence point, as illustrated in Fig. 4.16.

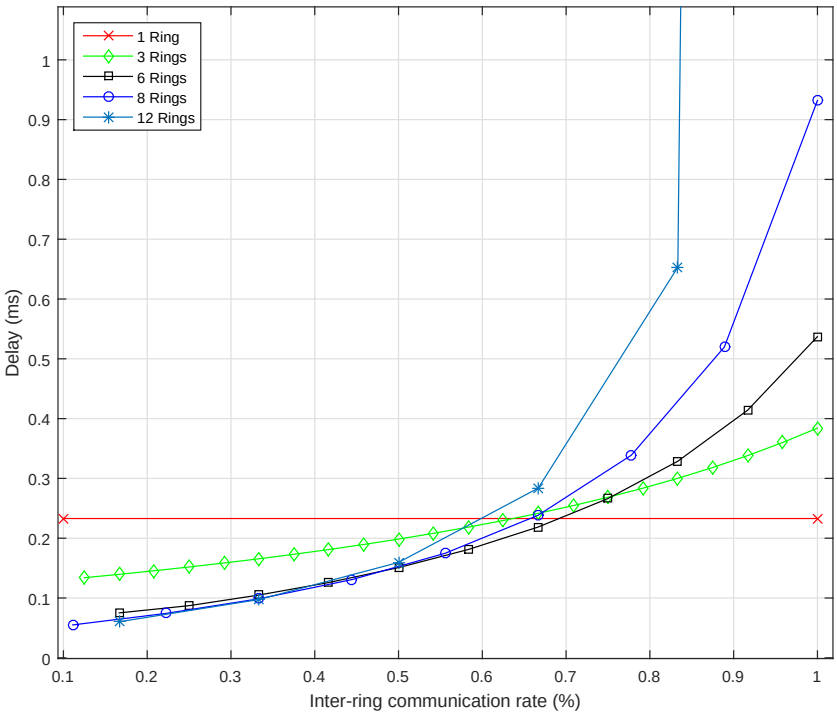


Figure 4.16: The impact of number of rings on delay bounds vs inter-ring communication load for ($interNet \in [0.2-1], M = 72, \sigma = 128bytes, \rho = 5 \cdot 10^5 bit/s$).

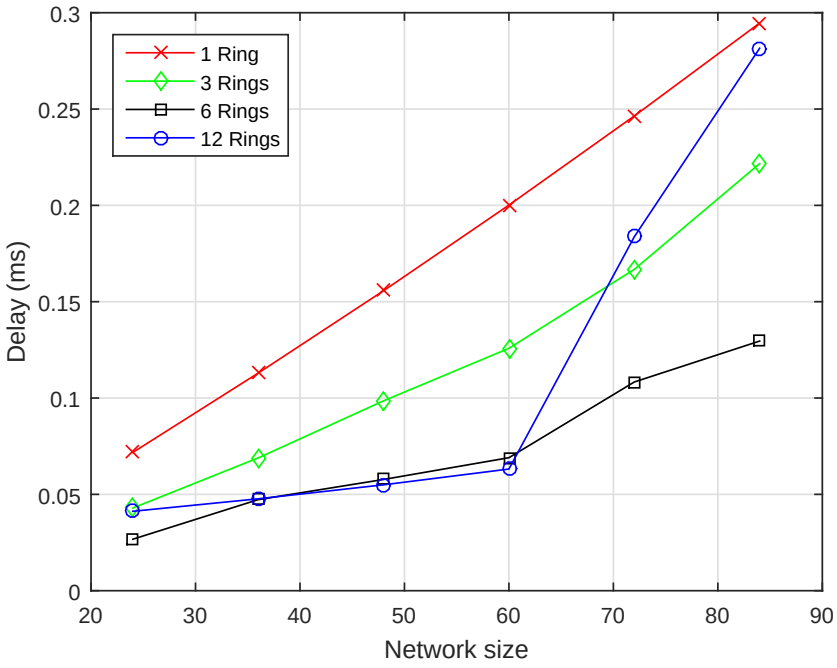


Figure 4.17: The impact of number of rings on delay bounds vs the network size, ($interNet = 0.2, M = [24-84], \sigma = 128bytes, \rho = 10^6 bit/s$).

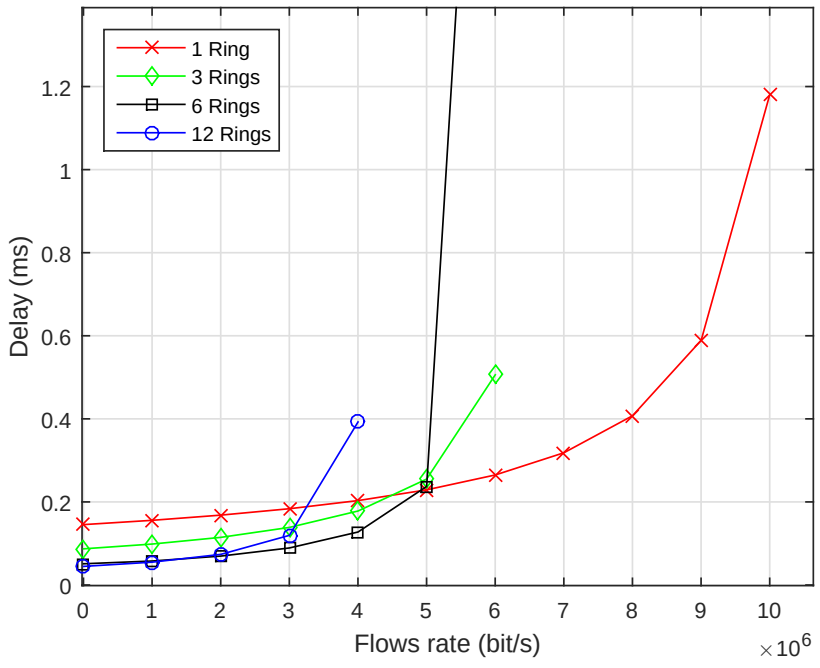


Figure 4.18: The impact of number of rings on delay bounds vs the flows rate, ($interNet = 0.2, M = 48, \sigma = 128bytes, \rho = [10^3 - 10^7] bit/s$).

Fig. 4.18 shows the impact of the flows rate on the end-to-end delay bounds. We observe that the multiple-ring network is more sensitive to the flows rate when the number of rings increases, i.e., the 12-rings network offers the lowest delay bounds for a rate up to 10^3bit/s, however it is the first to lead to the delay bound divergence for a rate bigger than 4×10^6. This fact is mainly due to the violation of the network utilization rate condition.

On the other hand,, we can also observe from Fig. 4.19 that multiple-ring topology is less sensitive to the flows burst than flow rate, i.e., the more the number of rings increases, the more the delay bounds decreases.

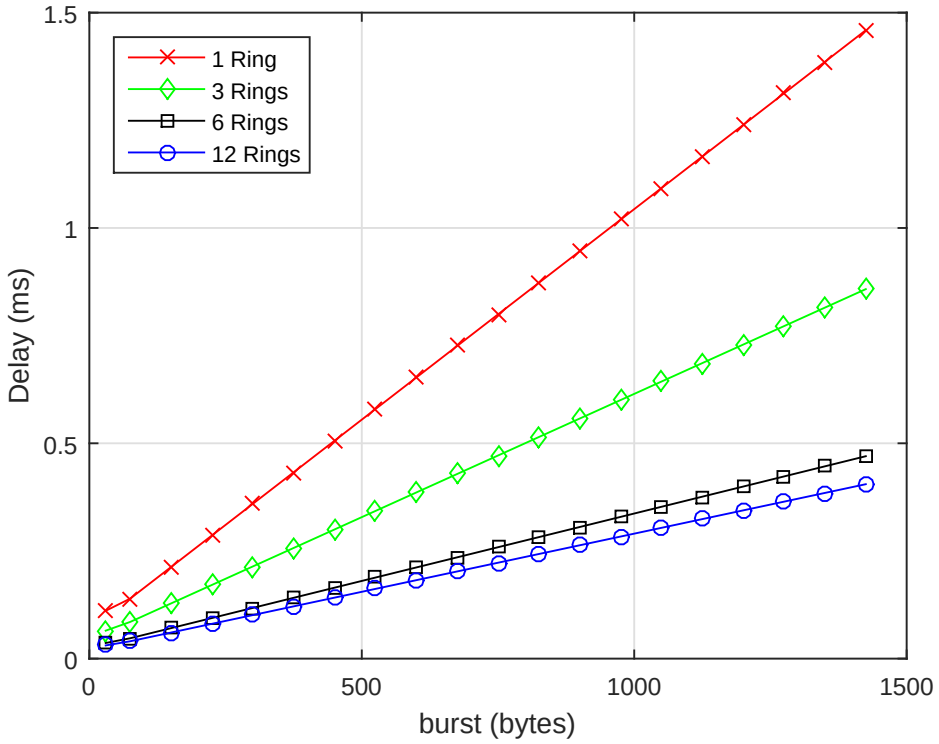


Figure 4.19: The impact of number of rings on delay bounds vs the flows burst, ($interNet = 0.2, M = 60, \sigma = [30 - 1500] bytes, \rho = 5 \times 10^5 bit/s$).

These results have shown that the end-to-end delay bounds of the multiple-ring topology are particularly sensitive to the inter-ring communication load and the flows rate. For a low inter-ring communication load, dividing the network into several rings may improve the end-to-end delay bounds, since it reduces the impact of interfering flows and the path length. However, for a high inter-ring communication load, the impact of convergence points increases with the number of rings, which leads to increasing the delay bounds.

4.5 Conclusion

In this chapter, we have analysed the timing performance of ring-based networks. The results have shown that our introduced approach, *PMOC*, which takes into account the flows serial-

ization phenomena, has improved in a noticeable manner the tightness of the delay bounds, in comparison with the conventional methods.

This approach has been generalized to the multiple-ring topology, and conducted analysis has shown that the multiple-ring topology may offer better performance than the mono-ring, when the inter-ring communication load is limited.

In the next chapter, we will detail the dependability analysis of AeroRing, and more particularly, the reliability level.

Chapter 5

Dependability Analysis of AeroRing

Contents

In this chapter, we are interested in the reliability of AeroRing, which is the ability of a system to continuously deliver its intended services throughout a given interval of time. The reliability requirements for avionic flight control systems are quite high, with a failure rate 10^{-9} failures per hour.

Hence, the guaranteed reliability level of AeroRing will be analytically quantified depending on several aspects, such as the network size, the equipment reliability and the mission time. To achieve this aim, we first present a short background of dependability analysis. Afterwards, we detail our system assumptions as well as the adopted failure model, and the AeroRing model. Finally, we present and discuss some numerical results of the reliability level of AeroRing under different scenarios.

5.1 Background

Dependability is defined as the trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers [69]. Additionally, it is defined as the ability to perform as and when required. We are interested in one of its main attributes, the ***Reliability***, which is the ability of a system to continuously deliver its intended service throughout a given interval of time [70].

The dependability is affected by several impairments: *faults, errors and failures* [70]. These impairments have a cause-effect relationship. A fault is a defect in the behavior of a system or in the way the system is designed or built, which can lead to errors. An error is an incorrect result delivered by the system, which can lead to failures. A failure is when the behaviour of the system deviates from the expected service. Two types of failures exist: i) permanent failures which do not recover; ii) and transient failures which can appear and disappear in some intervals of times.

Saying a fault or a failure depends on whether we are considering a subsystem or the whole system. Fig. 5.1 shows a simplified example of a T-AeroRing subsystem, consisting of the core, the power-supply, the PHY interfaces and the links. As shown, a failure at the link subsystem can lead to a fault at the T-AeroRing system.

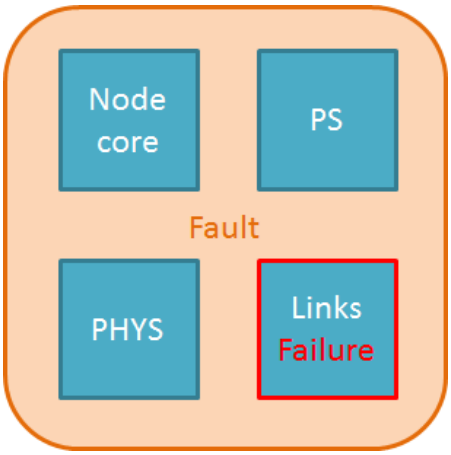


Figure 5.1: A simplified example of a T-AeroRing subsystem

Failures can manifest in different ways, called *Failure Modes*, according to their behaviour. These failure modes have been classified in [71] according to a hierarchy classification:

- Byzantine or arbitrary failures;
- Authentification detectable Byzantine failures;
- Incorrect computation failures;
- Performance failures;
- Omission failures;
- Crash failures;
- Stopping (Fail-stop) failures.

There are two types of system reliability evaluation: qualitative evaluation and quantitative evaluation [70]. The first one checks the ability of the system to deal with all the faults included in its fault model; whereas the second one aims to numerically verify that the system fulfills its dependability requirements, e.g., its intended reliability.

A well-known qualitative evaluation tool is the *Model checking* technique [72]. It allows formally verifying system properties. The first step consists in building a model of the system. Then, the user asks, by means of queries, whether or not the modelled system fulfills certain properties. Finally, a software tool, called model checker, exhaustively analyses all the possible states of the model and determines whether or not each property/query holds.

Quantitative evaluation techniques allow to analytically quantify the desired metric, as it is the case for the required reliability level for avionics system, e.g., the Design Assurance Level (DAL) of the avionics standards DO-254 [73] and DO-178 [74], which is represented by a failure rate. They are generally based on the specification of a model of the system. *Markov Chains* [75] and *Petri Nets* [76] are two of the most used formalisms for this type of evaluation.

These models depend on a certain amount of parameters, which have an impact on the final results, such as the equipment failure rate and the system fault-tolerance mechanisms coverage. The latter is an abstraction of the probability of success of the different processes, which constitute the fault-tolerance mechanisms [70]. This coverage has a strong influence on the dependability of a system, as it has been demonstrated and highlighted in [77, 78, 79].

In our case, we have selected Petri Nets, and more particularly Stochastic Activity Networks (SANs) [80, 81, 82, 83, 75]. SANs have been introduced in 1984 and are a stochastic extension of Petri nets (PNs). SANs are more powerful and flexible than most other stochastic extensions of Petri Nets, including Stochastic Petri Nets (SPNs) and Generalized Stochastic Petri Nets (GSPNs) [84]. SANs offer primitives, which allow to build up models easy to understand [85].

There are mainly five primitives in SANs:

- *Place*, which has a certain number of tokens to determine the state of the modelled system;

- *Activities* which, are connected to one or more source places and have one or more cases, where each one is connected to one or more destination places. Each activity can be enabled or disabled. When enabled, it fires, i.e., is launched to change the marking of places; thus, evolving the system state: i) immediately if it is an *instantaneous* one; ii) according to a statistical distribution, if it is a *timing* one, ;

- An *input gate*, which defines a condition for an activity to fire and how to change the marking of the source places;

- An *output gate*, which is connected to a given activity and specifies the marking changes to be performed depending on some conditions.

The SANs formalism also provides two additional primitives to build a model as a hierarchical composition of submodels:

- *Join* primitive, which allows interconnecting different submodels by sharing places;

- *Rep* primitive, which can be used to replicate a given submodel in order to model different instances of the same submodel.

Furthermore, the SANs formalism offers the possibility of specifying a reward model. A given reward model is associated to specific states of the model and aims to calculate a specific metrics or attributes, such as reliability, availability and throughput [70].

To conduct such analyses for AeroRing, we have used the Moëbius software [86] to build and solve analytically our SANs models. This tool is very used for model-based performance and dependability evaluation of real-time distributed systems.

5.2 System Assumptions and Failure Model

We present in this section the different system components and entities, as well as the adopted failure model.

5.2.1 AeroRing Components and Entities

An important step to model the reliability of the system is to identify the different components, which constitute the system (the T-AeroRings, the links and connectors), since the reliability of the system depends directly on the reliability of its components. The more a system has components, the more the probability of faults occurrence increases.

However, in practice, it is difficult to model a system according to each individual component. Instead, we look for a compromise, where we apply a certain level of abstraction and we gather the components of the system into different entities. These entities correspond to the

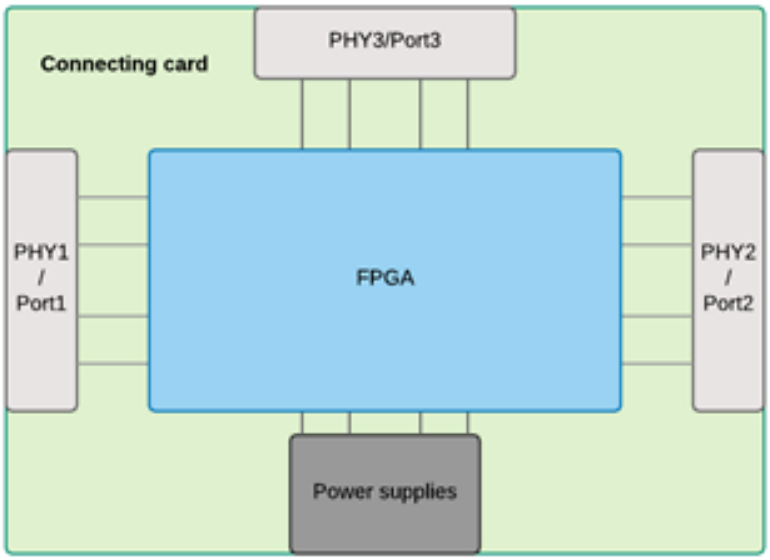


Figure 5.2: An architecture of a T-AeroRing with the different blocks

different blocks that constitute the T-AeroRing. Fig. 5.2 shows the architecture of a T-AeroRing with these different blocks and Fig. 5.3 shows a picture of a T-AeroRing prototype.

A T-AeroRing is consists of:

- The *Node Core*, which represents the intelligence of the T-AeroRing. It is an FPGA (Field-Programmable Gate Array) with an internal memory responsible for the data computation and processing;

- Three *PHY interfaces* entities, which correspond to the three ports of AeroRing;

- The *power supply* entity responsible for providing the T-AeroRing by the needed power;

- A *connecting card*, which connects all the different entities and handles the intra-communications.

For the communication medium, we consider the *Attachment* entities, which represent the connectors and connected cables.

5.2.2 Failure Model

A realistic assumption for modelling the time to failure of a given component is to assume it exponentially distributed [87]. This fact simplifies the way in which the models are mathematically treated to obtain a numerical solution. Hence, the corresponding SANs model will be

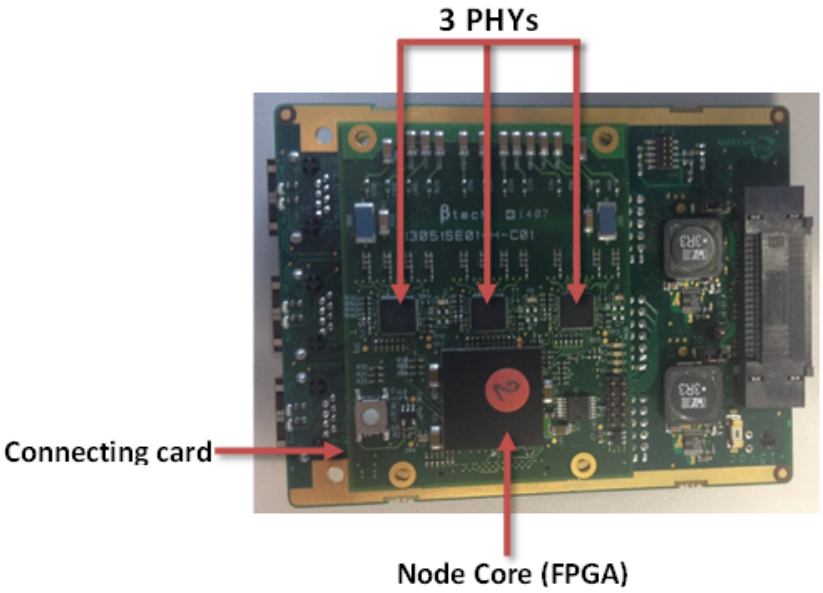


Figure 5.3: A real T-AeroRing prototype with the different components

characterized by means of CTMC (Continuous Time Markov Chain), which can be analytically solved [86]. It is worth saying that we are interested only in the permanent failures to conduct our analyses.

Since components can fail in different ways, we affect a probability of occurrence (or a weight) to each possible type of failures of the component. The different identified failures are *the crash, the syntax errors, the byzantine error and the timing and babbling errors*, which correspond to *a crash failure, an incorrect computation failure and a performance failure*, respectively.

In addition, we consider the reconfiguration errors, which are related to the efficiency of the fault detection and redundancy mechanisms, i.e., the coverage of the fault-tolerance mechanisms.

To prevent from the incorrect computation, byzantine and performance failures, we consider that the responsible entity, i.e., *core node*, is redundant within the T-AeroRing. All the operations are performed in parallel by both core nodes, and the final results are compared. In the nominal case, the results are similar; otherwise, the equipment is considered as faulty and turned off by the non faulty core node to not propagate its fault within the network.

In the rest of the chapter, we consider the notations and default values of the AeroRing model parameters described in Table 5.1.

5.3 AeroRing Model

In this section, we describe the followed strategy and the reliability models of our system for the different topologies.

Table 5.1: AeroRing model parameters

Parameter	Default value	Meaning
CC		connecting card
PS		power supply
NC		node core
failure rate (FR)	10^{-9}	failure rate
Core_failure_rate	10^{-9}	The core node (FPGA) failure rate
Card_failure_rate	10^{-9}	The connecting card failure rate
PS_failure_rate	10^{-9}	The power supply failure rate
PHY_failure_rate	10^{-9}	The PHY interfaces failure rate
link_failure_rate	10^{-9}	The links failure rate
p_crash_core	0.5	FPGA crash probability
p_syntax_core	0.2	FPGA syntax error probability
p_tb_core	0.15	FPGA timing or babbling errors probability
p_reconf_core	0.15	FPGA reconfiguration errors probability
p_crash_card	0.75	Connecting card crash probability
p_syntax_card	0.25	Connecting card syntax error probability
nodeNumber	[10 ~ 100]	Size of the network
toleratedFault (TF)	0 or 1	Number of tolerated faults
nbrSystems	[1 ~ 3]	Number of duplicated networks
Dup_cov	0.999999999	Duplication error-containment coverage
FD_cov	0.999999999	Fault detection mechanism coverage
AC_cov	0.999999999	Auto-configuration mechanism coverage

5.3.1 Modelling Strategy

In order to model our system, we have divided our system model into SANs submodels classified in 4 categories:

- ***Cat1.*** models the fault occurrence of the different entities. The basic structure of this submodel is depicted in Fig. 5.4. It consists of a place, called *nodes*, to represent the network nodes; an activity *TA_Failure* to model the different failure occurrences. Then for each failure mode, there is a set of places to represent how the fault manifests, and a set of places to determine which entity fails within the T-AeroRing;

- ***Cat2.*** models the impact of the failure on the network. When a failure mode and the faulty entity are selected, a token is written at the place *FailedEntityx*, which represents this case. Then, a Cat2. submodel will model its impact on the network, e.g., the ring becomes a line or increasing the number of faulty nodes;

- ***Cat3.*** according to the failure mode, a Cat3. submodel determines whether the fault is successfully handled or not, e.g., the duplication of the *core node* has succeeded in case

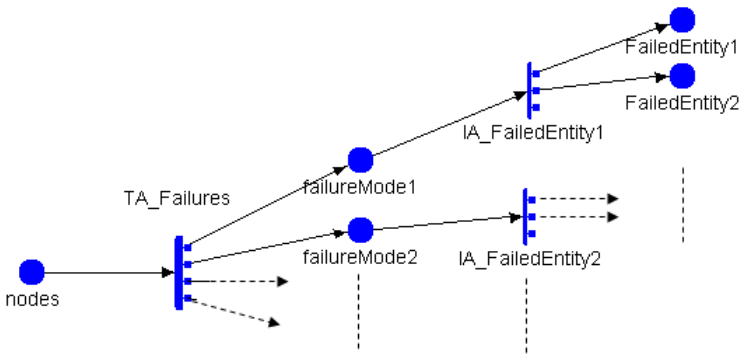


Figure 5.4: Basic structure of Cat1. submodel

of a performance failure or the recovery mechanism has succeeded to reconfigure the ring into a line;

- ***Cat4.*** This submodel is used for a system fault evaluation and it takes into account the state of the network, e.g., the occurred faults, whether or not each one of them has been successfully isolated, to decide if the system is failed or not. This decision will depend on several parameters, such as the number of tolerated faulty nodes and the system redundancy. For instance, the *SFeval* in Fig. 5.9 is a Cat4. submodel.

5.3.2 AeroRing Submodels for Simple Mono-Ring Topology

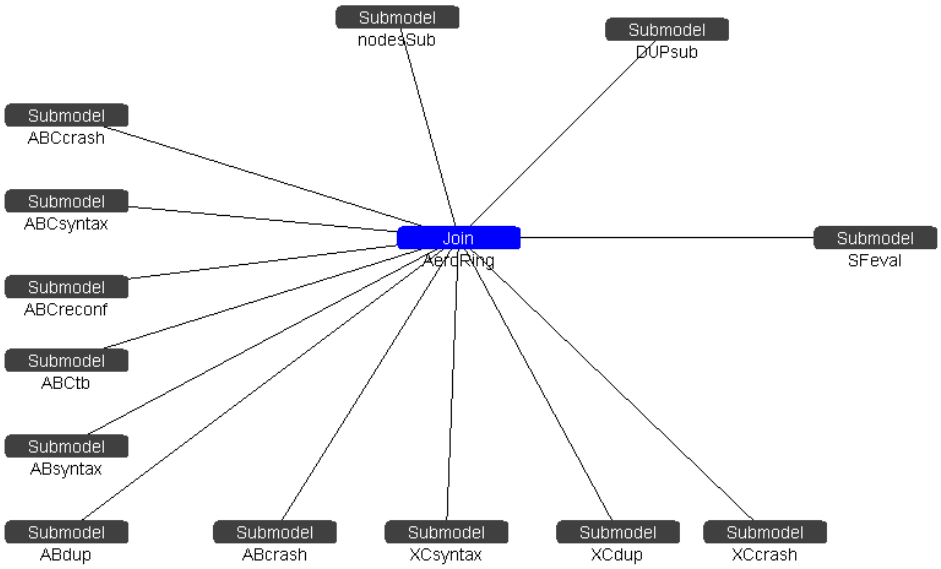


Figure 5.5: AeroRing composed model for simple mono-ring topology

Fig. 5.5 shows the AeroRing composed model for simple mono-ring topology, which consists of submodels interconnected by means of the Join primitive. We can notice that it includes the Cat1. submodel, *nodesSub*, which models the fault occurrence of the different entities; the Cat2. submodels at the left and the bottom side of the figure; the Cat3. submodel, *Dupsub*; and the Cat4. submodel, *SFeval*.

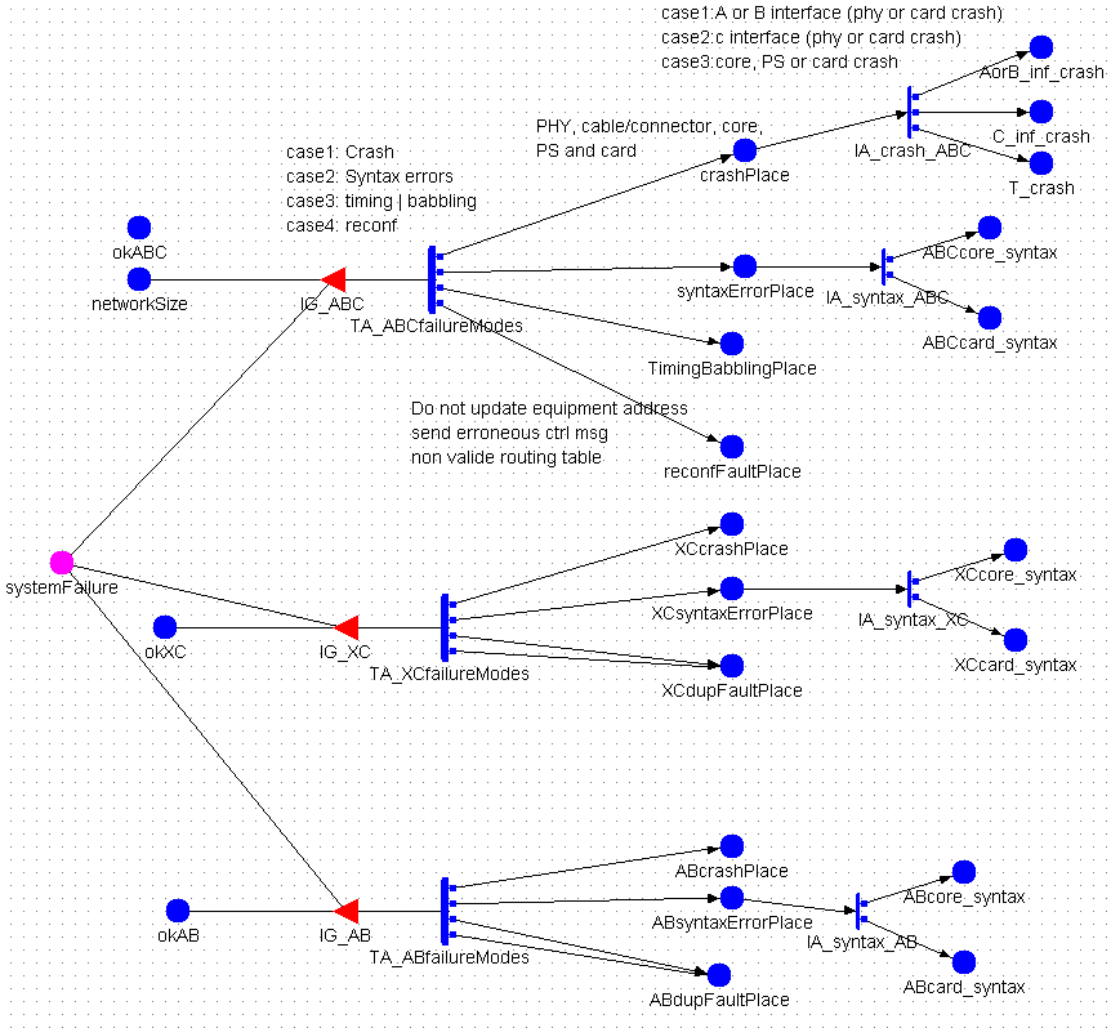


Figure 5.6: Faults occurrence on *nodesSub*

Fig. 5.6 shows our Cat1. submodel *nodesSub* that models the faults occurrence on the network nodes, i.e., T-AeroRings. Note that we have 3 different types of places for the T-AeroRings:

- A place *okABC* containing tokens representing the number of T-AeroRings, which have two neighbours and an equipment;
- A place *okAB* with a number of tokens representing the number of T-AeroRings, which lost the connection with the connected equipment;

- A place *okXC* whose number of tokens represents the number of T-AeroRings which lost a connection with one of their neighbours.

These places are connected to activities, which model the failure rates of any of their entities, and have a set of cases representing different failure modes. Each mode is connected to a place, which represents the failure mode. If different entities of a node can be affected by a failure mode, the corresponding place is connected to an instantaneous activity that have a set of cases representing the different entities, e.g., a crash in an okABC node can be due to the failure of one of the PHYs, the core node or the power supply.

The activity *TA_ABCfailureModes* models the time that elapses until any non-faulty node fails. This time is exponentially distributed taking into account the failure rate of all non-faulty nodes:

$$okNodes->mark() \times node\_failure\_rate$$

As explained in Section 5.2.1 and shown in Fig.5.2, the failure rate of a T-AeroRing is the sum of failure rates of its entities, i.e., *the Node Core, the three interfaces, the power supply and the connecting card.*

$$\begin{aligned} node\_failure\_rate = & 3.0 \times (PHY\_failure\_rate + link\_connector\_failure\_rate) \\ & + Core\_Failure\_Rate + PS\_failure\_rate + Card\_failure\_rate \end{aligned}$$

When the activity *TA_ABCfailureModes* fires, a token is erased from *okNodes* and added to one of its four cases, which corresponds to the chosen failure modes, i.e. *the crash, the syntax errors and the timing and babbling errors.* These failure modes are selected according to their probability of occurrence. For instance, for the first case corresponding to a crash, all the entities that constitute the T-AeroRing can crash. The probability of crash for the PHYs and PS entities when a failure happens is one, since it is their only failure mode. The probability of crash for the node core and the CC entities are respectively *p_crash_core* and *p_crash_card*. According to this, the probability of the crash case is:

$$\begin{aligned} case1 = & \frac{3.0 \times (PHY\_failure\_rate + link\_connector\_failure\_rate) + PS\_failure\_rate}{node\_failure\_rate} \\ & + \frac{p\_crash\_core \times Core\_Failure\_Rate + p\_crash\_card \times Card\_failure\_rate}{node\_failure\_rate} \end{aligned}$$

The second case corresponds to a syntax error, which can appear at the level of the NC of the CC. The probability of syntax error for the NC and the CC entities are *p_syntax_core* and *p_syntaxh_card*, respectively. According to this, the probability of the syntax error case is:

$$case2 = \frac{p\_syntax\_core \times Core\_Failure\_Rate + p\_syntax\_card \times Card\_failure\_rate}{node\_failure\_rate}$$

The third case corresponds to the timing and babbling errors, which can happen at the level of the NC. The probability of failure for the NC is *p_tb_core*. According to this, the probability of this case is:

$$case3 = \frac{p\_tb\_core \times Core\_Failure\_Rate}{node\_failure\_rate}$$

The last case corresponds to the the reconfiguration errors due to the NC with a probability p_reconf_core:

$$case4 = \frac{p\_reconf\_core \times Core\_Failure\_Rate}{node\_failure\_rate}$$

Once the failure mode is selected, an instantaneous activity selects the responsible entity among the entities, where the failure mode can manifest, e.g., for the *crashPlace*, the activity *IA_crash_ABC* determines whether it is a crash of a ring PHY entity, the equipment PHY entity or a T-AeroRing crash related to the crash of the NC or CC. The probability of the three cases is:

$$case1 = \frac{2.0 \times (PHY\_failure\_rate + link\_connector\_failure\_rate)}{node\_failure\_rate}$$
$$case2 = \frac{(PHY\_failure\_rate + link\_connector\_failure\_rate)}{node\_failure\_rate}$$
$$case3 = \frac{(Core\_Failure\_Rate + Card\_failure\_rate)}{node\_failure\_rate}$$

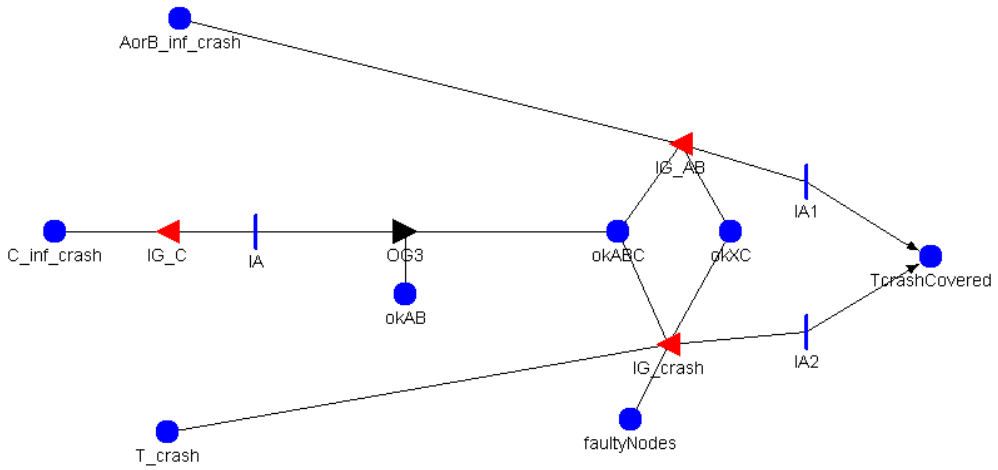


Figure 5.7: ABCcrash submodel

Once the failed entity selected, e.g., *AorB_inf_crash, T_inf_crash or reconfFaultPlace*, the corresponding Cat2. submodel models its impact on the network. For instance, for a *T_crash*, the Cat2. submodel is *ABCcrash* as shown in Fig. 5.7. The place *T_crash* is connected to an input gate enabled if its marking is greater than one; and connected to an instantaneous activity to mark the place *TcrashCovered*, which launches the coverage process modelled in the Cat3. submodel *DUPsub*, illustrated in Fig. 5.8.

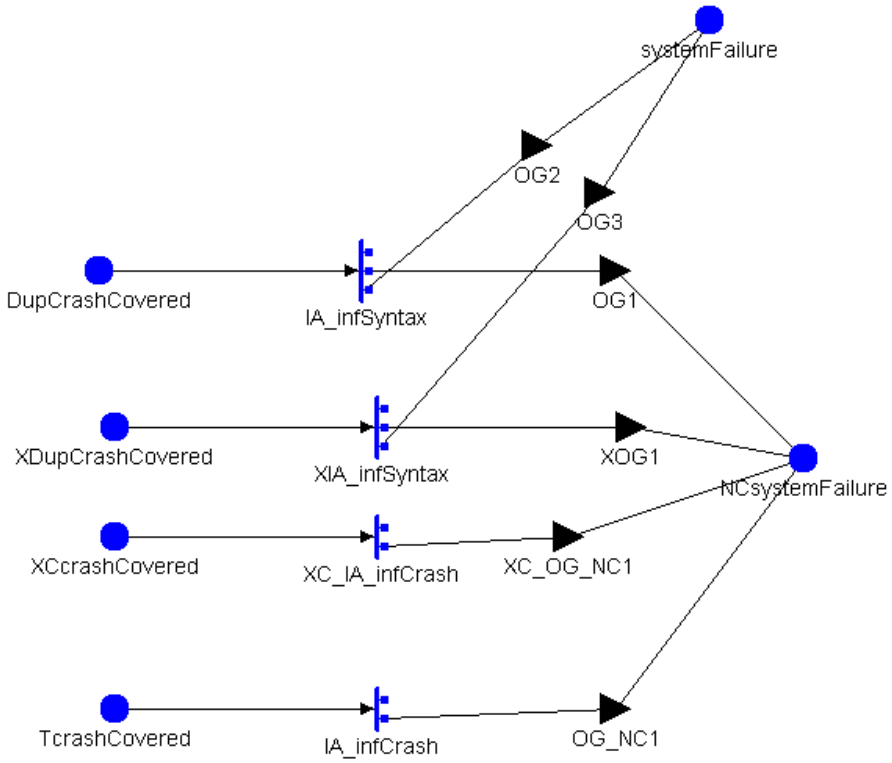


Figure 5.8: DUPsub submodel

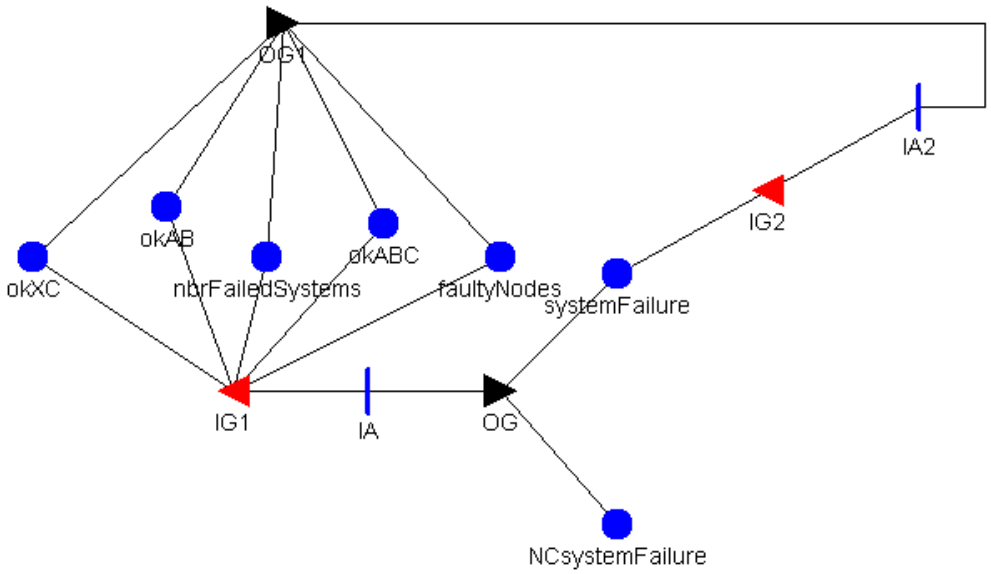


Figure 5.9: SFeval submodel

DUPsub computes the probability whether the crash or the duplication coverage succeeds. When a T-AeroRing crashes, two neighbours should detect the failure and reconfigure the network. The coverage probability is:

$$FD\_cov^2 \times AC\_cov$$

In case of performance or computation failures, the duplication mechanism should succeed in addition to the fault detection and reconfiguration mechanisms. The coverage probability is:

$$Dup\_cov \times FD\_cov^2 \times AC\_cov$$

The SFeval submodel follows the evolution of the network state and determines whether the network is faulty or not, according to the T-AeroRing states and the number of tolerated faulty nodes. Fig. 5.9 shows the SFeval submodel. The input gate *IG1* enables to fire the instantaneous activity *IA* when the system is down. The input prediction of *IG1* is $okXC->Mark() > 2$, which means that the ring is divided into lines (more than one); or $okAB->Mark() + faultyNodes->Mark() > toleratedFault$, which means that the number of lost equipment is more than the tolerated faults. Once the *IA* fires, the output gate *OG* puts a token in the *systemFailure* and *NCsystemFailure* places to indicate that the system is failed. The input gate *IG2* allows to enable the instantaneous activity *IA1* that allows to reset all the places to zero and stop the evolution of the system. It allows also to increment the marking of the place *nbrFailedSystems*, which counts the number of faulty systems in case of redundancy, as it will be explained in the next section.

5.3.3 AeroRing Model for Duplicated Mono-Ring Topology

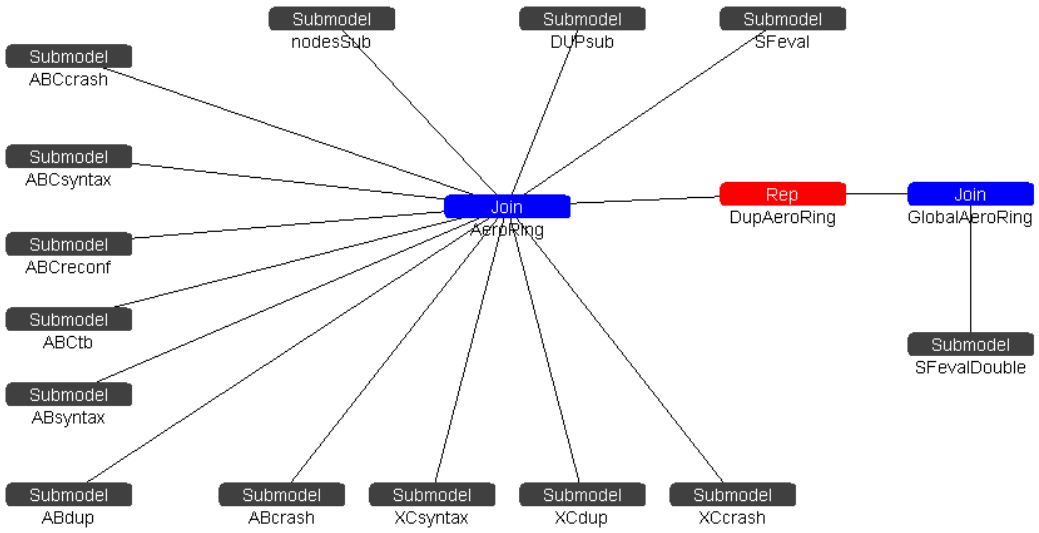


Figure 5.10: AeroRing composed model for duplicated mono-ring topology

Fig. 5.10 shows the AeroRing model for the duplicated moro-ring topology. We use a Rep primitive *DupAeroRing* to duplicate the described AeroRing model shown in Fig. 5.5. The *SFeval* subsystem of each ring increments the marking of a shared place, *nbrFailedSystems*, when the associated ring is down. This replicated model is joined with a *SFevalDouble* subsystem that observes the state of the global system, illustrated in Fig. 5.11. The global system is down if all the replicated systems are down, i.e., $nbrFailedSystems->Mark() == nbrSystems$. In that case the output gate *OG* marks the *GlobalSystem* place, which is enabled when the *IA* fires with the prediction of *IG* of $nbrFailedSystems->Mark() == nbrSystems$.

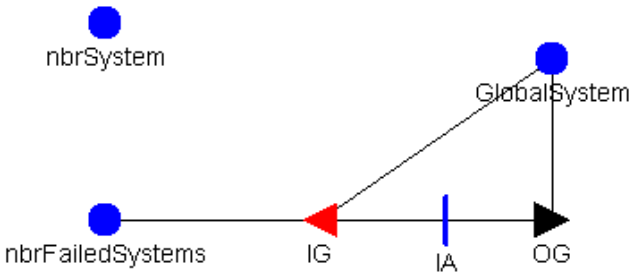


Figure 5.11: SFevalDouble submodel

5.3.4 AeroRing Model for Multiple-Ring Topology

Fig. 5.12 shows the AeroRing composed model for multiple-ring topology. In order to model the Multiple-ring topology, we have modelled the peripheral rings by replicating the AeroRing model shown in Fig. 5.5, using the rep primitive *PeripheralRep*; and the backbone network by a separate ring, joined by the primitive *backbone*, where the size of the backbone is equal to the the number of replicated peripheral ring.

The *SFevalMulti* submodel is responsible to evaluate the state of the network, as shown in Fig. 5.13. The network is down if one of the rings is down, i.e., peripheral or backbone.

It is worth noting that the gateways model is the same as the T-AeroRing model.

5.4 Numerical Results

In this section, we detail some numerical results of the reliability level of AeroRing under different scenarios. First, we describe the considered case study with the different parameters and scenarios. Then, we conduct a sensitivity analysis of the reliability of AeroRing according to several parameters, such as the network size, the failure error and the network topology.

5.4.1 Case Study

We consider the case study with the default values of the model parameters described in Tab. 5.1.

Scenarios are generated varying the network size, the failure rates of the T-AeroRing entities, the number of tolerated faults, the number of duplicated systems and the network topology as follows:

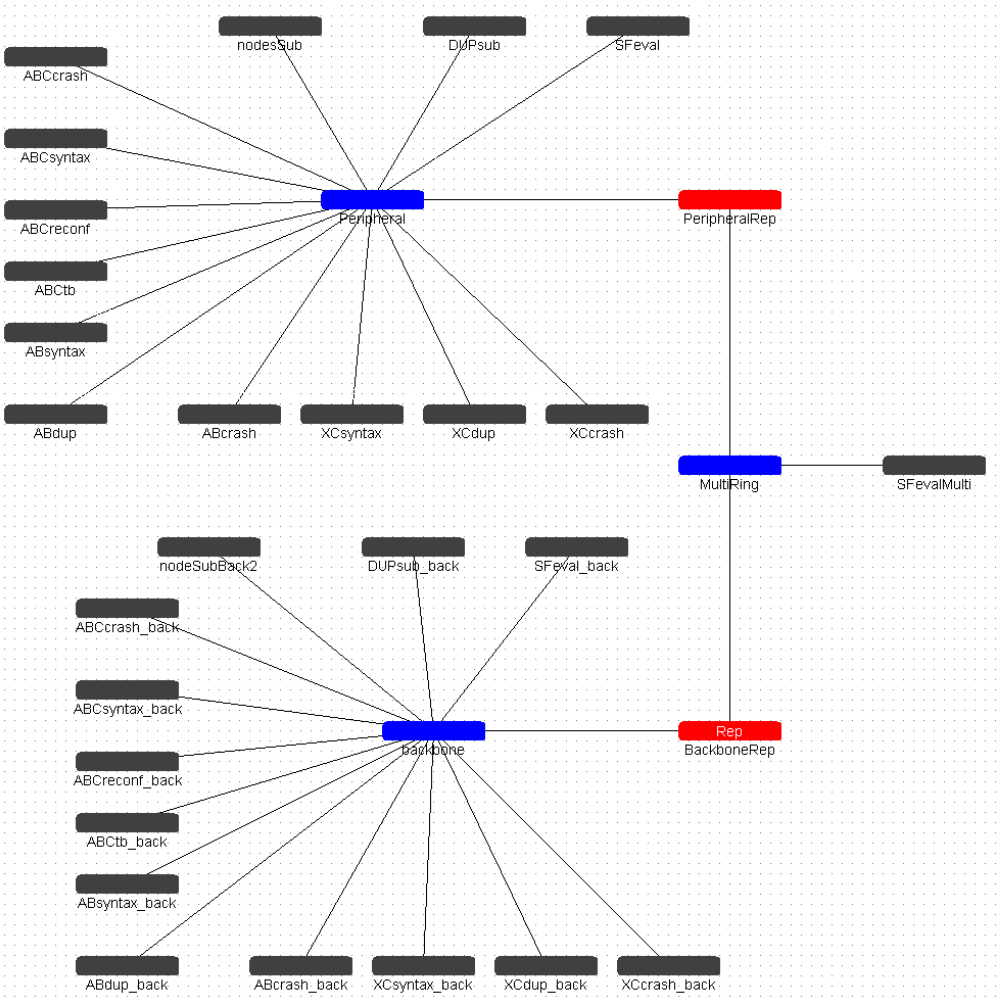


Figure 5.12: AeroRing composed model for the multiple-ring topology

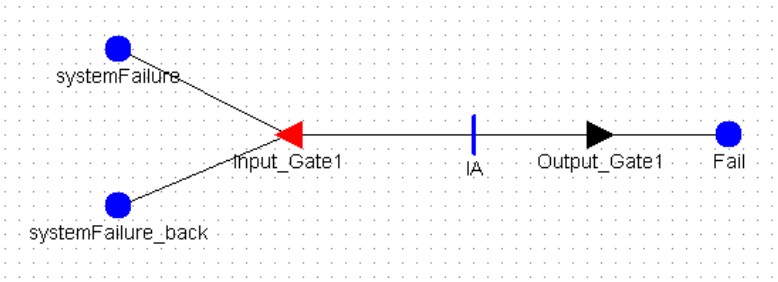


Figure 5.13: SFevalMulti submodel

- The number of nodes varying from 10 to 100 with a step of 10, i.e. *nodeNumber* $\in$ [10, 100];
- The failure rate of the components varying from 10^{-6} to 10^{-10} failure/hour;
- The system tolerates zero or a single failure (*TF=0, TF=1*);
- The system can be replicated up to 3 times;
- The mission time;
- We consider both Mono-ring and Multiple-ring topologies.

5.4.2 Sensitivity Analysis

We discuss herein the impact of the different model parameters on the system reliability and the mission time.

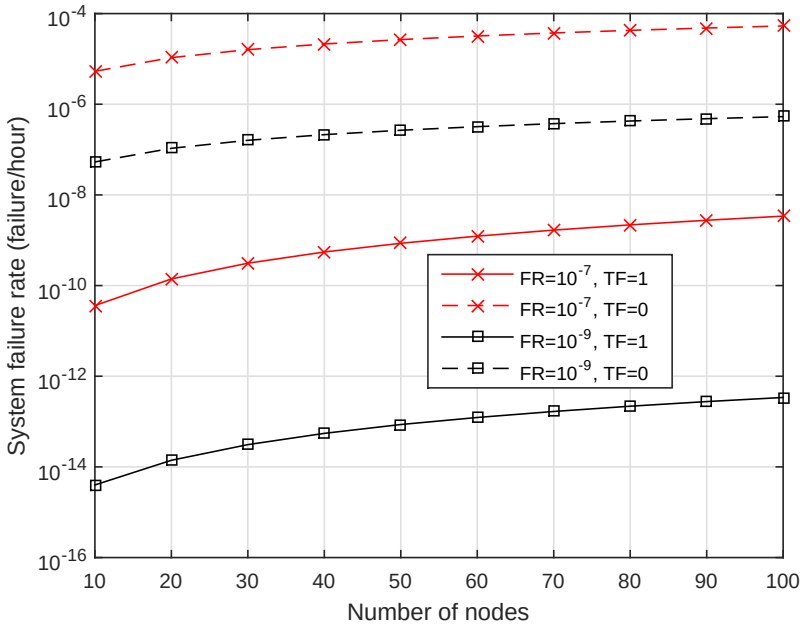


Figure 5.14: System failure rate of the mono-ring vs size of the network when varying the failure rate of components and number of tolerated failures

In Fig. 5.14, we plot the System failure rate of the mono-ring topology according to the network size with an equipment failure rate $FR = 10^{-7}$ and 10^{-9} and zero or one fault tolerance ($FT = 0$ or 1). Clearly, the system failure rate is reduced in a significant way when the system tolerates a failure, since the system is considered down in case of two failures. This scenario corresponds to the case when the critical nodes are duplicated within the network. The system failure rate depends also on the network size and the equipments *FR*. increasing the number of network equipment increases the probability of fault occurrence at the system level, which in its turn decreases the reliability of the system. For instance, when increasing the network

size from 10 nodes to 100 nodes with zero tolerance and an equipment $FR = 10^{-9}$, the system failure rate is increased by 900%. Furthermore, decreasing the equipment reliability decreases the system reliability.

Fig. 5.15 shows the system failure rate when varying an entity FR (others entities FR are fixed to 10^{-9}). As we can see, the system failure rate increases with the entity FR, since the faults occurrence increases. We can notice that the results of the different entities are almost similar, which means that the reliability of different entities impacts the system almost in the same way.

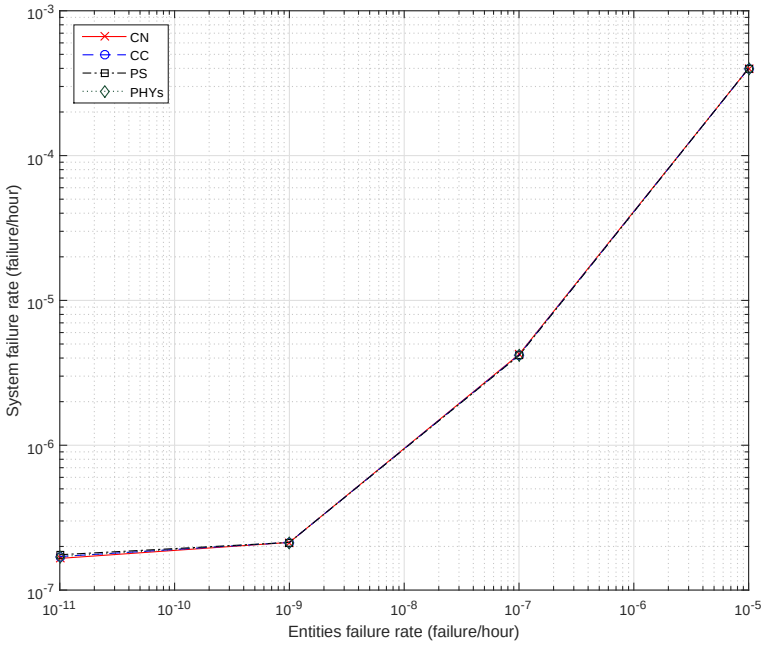


Figure 5.15: Mono-ring system failure rate vs entities failure rate for a network size of 40 nodes

Fig. 5.16 shows how the system duplication improves the global system reliability. In this scenario, a ring does not tolerate any fault and the equipment FR is 10^{-7}. As we can see, the system failure rate is noticeably improved when the system is duplicated for the different network sizes. For a network of 50 nodes, the failure rate is 2.29×10^{-5} and 7.27×10^{-10} for the single and duplicated system, respectively. In addition, the failure rate is almost null when adding a third replica. These results show that a fully redundant network achieves higher reliability level than its components.

Fig. 5.17 shows the evolution of the failure rate according to the mission time and the number of system replicas. In this scenario, the system does not tolerate any fault and the equipment FR is 10^{-9}. As we can see, the failure probability increases with the mission time. It goes from 2.13×10^{-7} for a mission time of one hour to 5.12×10^{-6} for a mission of 24 hours in case of a single ring; and from 0 to 1.33×10^{-10} for the duplicated ring. The latter shows the high reliability level of AeroRing based on duplicated topology, i.e., it meets the DAL-A

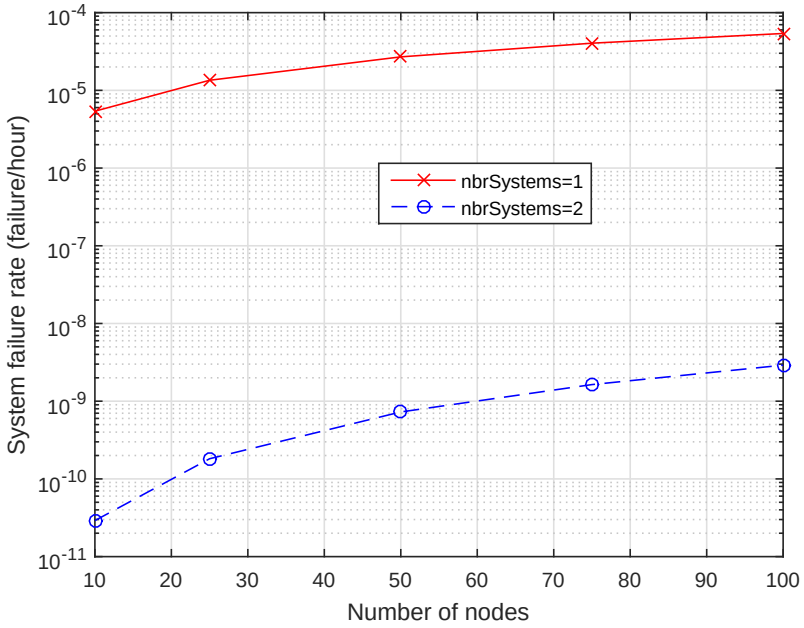


Figure 5.16: Mono-ring system failure rate vs size of the network when varying the number of system replicas

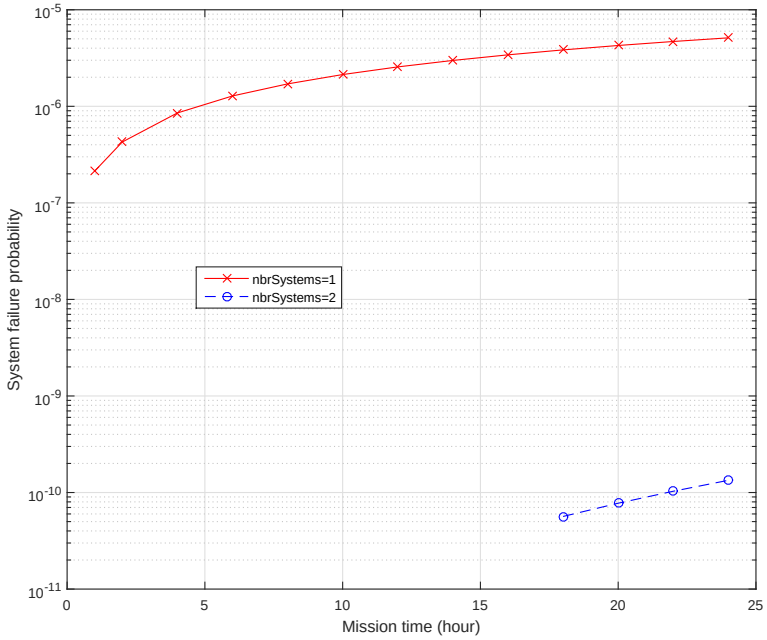


Figure 5.17: System failure rate vs mission time when vaying the number of system replicas

requirements for a mission time of 24 hours ($< 10^{-9}$).

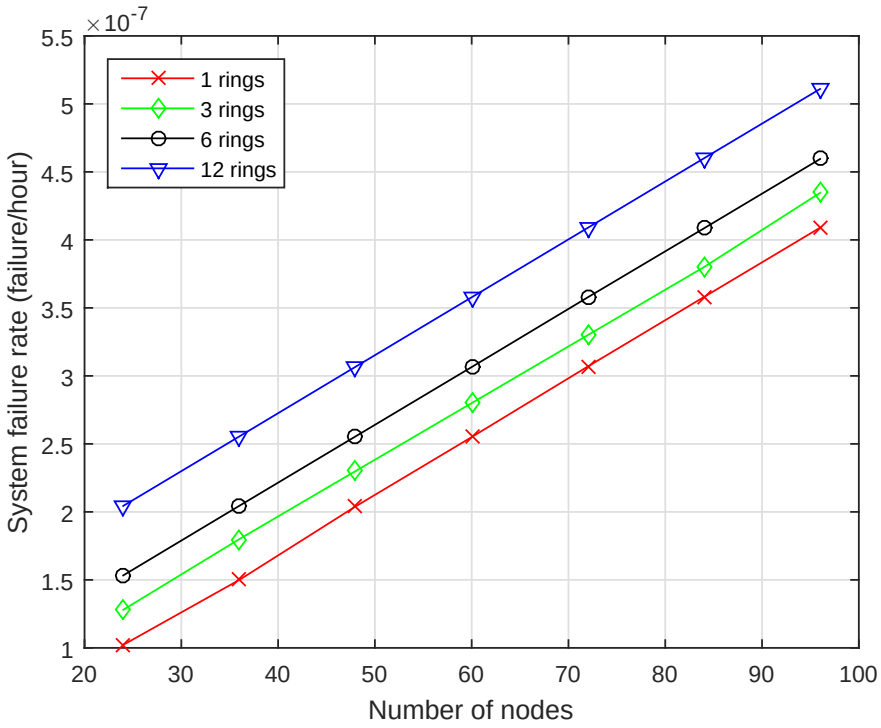


Figure 5.18: System failure rate for different AeroRing topologies vs network size

Fig. 5.18 shows the impact of the network size and the number of rings on the system reliability. As we can see, the failure rate increases when increasing the rings number. This result is coherent with the results of Fig. 5.14 and 5.16, since the multiple-ring topology introduces additional gateways, i.e., the number of gateways is equal to $2 + (number\_of\_rings) \times 2$, which increases the system failure rate.

5.5 Conclusions

In this Chapter, we have analysed the achievable reliability of AeroRing using SANs and shown how the different parameters of the system affect such metric.

Results show that AeroRing reliability level meets the avionics constraints, i.e., DAL-A, when using duplicated mono-ring topology. We have seen that all the T-AeroRing entities, i.e., CN, CC, PS, PHYs and links, have almost the same impact on the reliability of the system, as illustrated in Fig. 5.14 and 5.15. Furthermore, results show that the multiple-ring topology has comparable reliability level than the mono-ring topology.

Chapter 6

Validation on an Avionics Case Study

Contents

In this chapter, the validation of AeroRing performance, i.e., predictability and reliability levels, is conducted through a realistic avionics case study. First, AeroRing performance is compared with the current avionics network of an A380 based on the AFDX standard, and the most relevant RTE solutions. Then, we conduct a new comparison, when considering a generic configuration, through varying the network and flows parameters.

Firstly, we describe the avionics case study and the considered scenarios. Then, we report the timing performance and availability level, i.e., the maximum end-to-end delay bounds and the maximum recovery time, of AeroRing in comparison with the AFDX and RTE solutions under different network topologies. Afterwards, we report the reliability level of AeroRing for the different considered topologies to see whether or not it meets the avionics requirements. Secondly, we describe the generic case study and discuss the obtained results of the comparison of AeroRing with the most relevant RTE solutions.

6.1 Avionics Case Study

The considered case study is a representative avionics backbone network of an A380. As shown in Fig. 6.1, it consists of 8 AFDX switches connecting 54 end-systems, where there are 6 switches connecting between 6 and 13 end-systems each, and two additional switches connecting the others switches to reduce the number of hops of exchanged data between any two end-systems. Table 6.1 summarizes the described configuration in terms of number of end-systems and VLs within each switch.

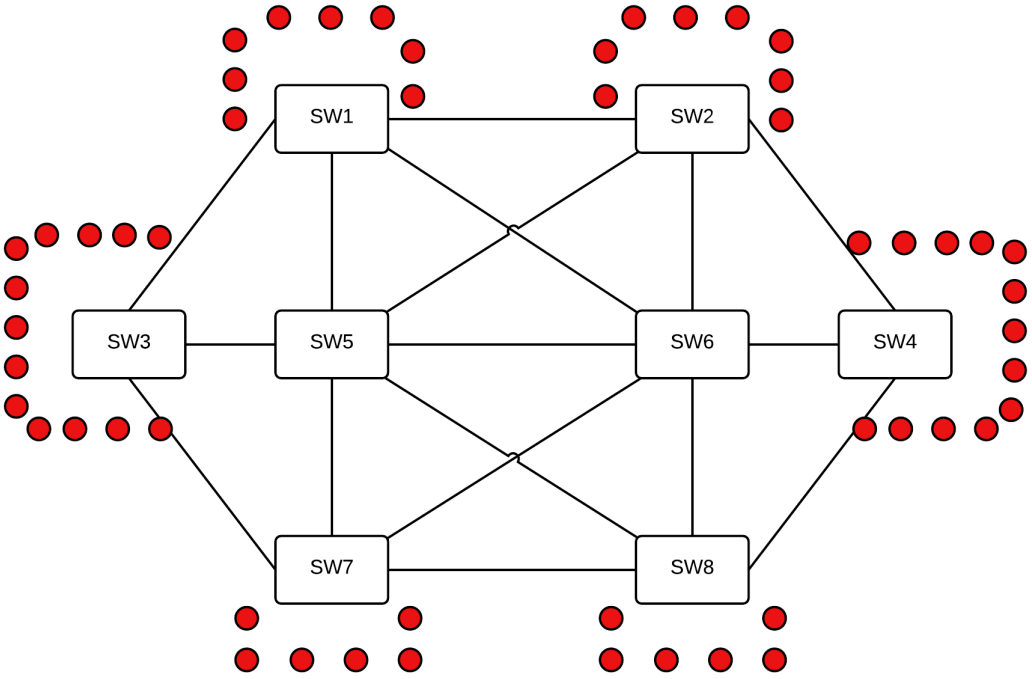


Figure 6.1: A representative A380 AFDX network

Table 6.1: Description of the AFDX configuration

Switch ID	# end-systems	# VLs
SW1	8	64
SW2	8	64
SW3	13	104
SW4	13	104
SW5	0	0
SW6	0	0
SW7	6	48
SW8	6	48

As shown in Table 6.2, the AFDX configuration is based on 432 different VLs distributed within three different traffic classes (TCs): the first class has a BAG value of 4 ms and MFS value of 480 bytes; the second class has a BAG value of 8 ms and MFS value of 16 bytes; and the third class has a BAG value of 32 ms and MFS value of 480 bytes. Moreover, each end-system generates 8 VLs, as illustrated in Table 6.2.

Table 6.2: Traffic Classes

TC	Period (ms)	Payload size (byte)	Rate (bit/s)	# VLs (Flows) per end-system
1	4	480	1024×10^3	1
2	8	16	72×10^3	1
3	32	480	128×10^3	6

To investigate the AeroRing performance, we replace the current AFDX backbone network described in Fig. 6.1 and Table 6.1 by an AeroRing network. Then, we compare the obtained results with reference to the AFDX network and the most relevant RTE solutions, described in Chapter 2. It is worth noting that the current implementations of the AFDX and the considered RTE solutions have a speed of 100 Mbps. However, to conduct our comparison, we enlarge their speed to 1Gbps. The considered topologies are described in Table 6.3.

Table 6.3: Multiple-ring configurations

Peripheral ring id	6 rings	4 rings	3 rings	1 ring
R1	SW1	SW1+SW7	SW1+SW2	SW1+SW2+SW3+SW4+SW7+SW8
R2	SW2	SW3	SW4+SW8	-
R3	SW3	SW2+SW8	SW3+SW7	-
R4	SW4	SW4	-	-
R5	SW7	-	-	-
R6	SW8	-	-	-

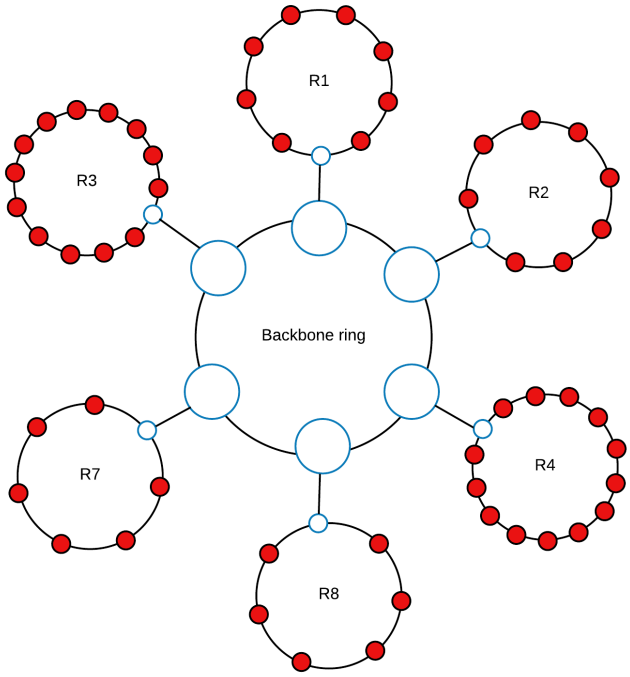


Figure 6.2: 6 rings AeroRing network

Furthermore, we consider the three following scenarios.

Scenario 1 in this scenario, we asses the temporal performance and the availability level of AeroRing, i.e., the maximum end-to-end communication delay bounds and the redundancy recovery time, in comparison with the AFDX and RTE solutions. Hence, we consider the AeroRing topology described in Fig. 6.2 and Table 6.3, where we have replaced the AFDX network by a multiple-ring topology, i.e., each switch with its connected end-systems is replaced by a peripheral ring and all the peripheral rings are connected to a backbone ring. Moreover, two service policies are analysed:

1. we consider only one priority level with FIFO policy;

2. we consider a Static Priority (SP) policy, where we affect to each traffic class a priority level, i.e., TC1 has the highest level, whereas TC2 and TC3 the medium and lowest levels.

Scenario 2 in this scenario, we asses the impact of the multiple-ring topology on the system performances. For this, we compute the maximum end-to-end communication delay bounds and the redundancy recovery time, under several multiple-ring topologies. The considered topologies are:

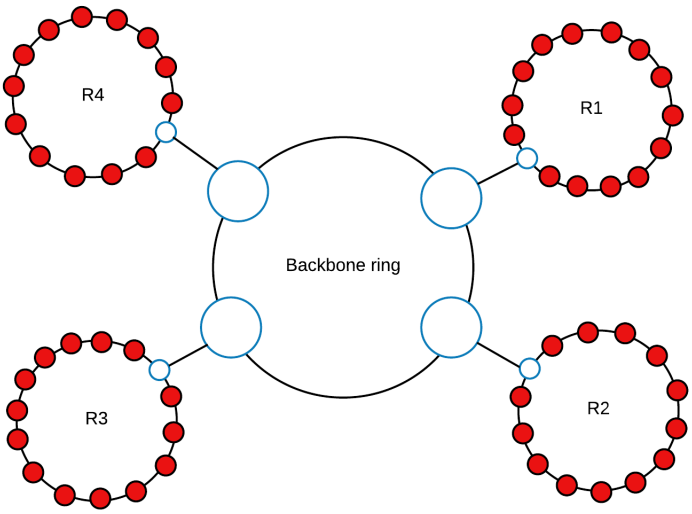


Figure 6.3: 4 rings AeroRing network

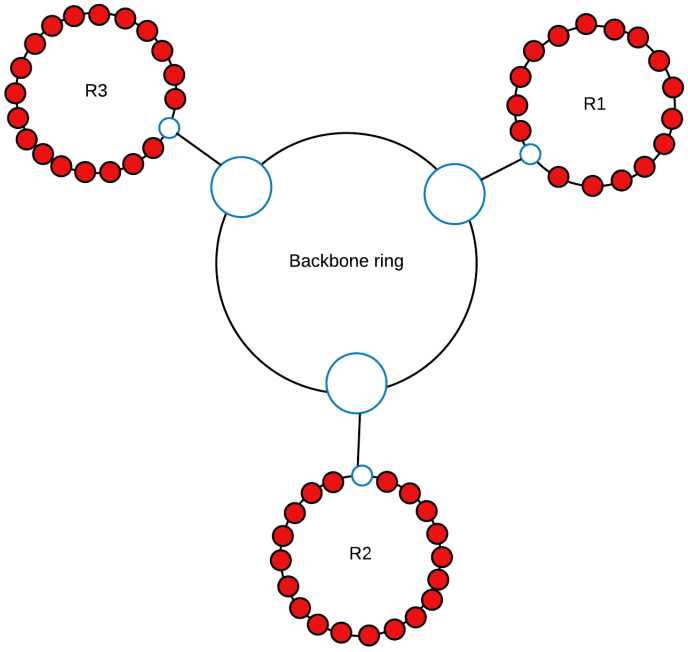


Figure 6.4: 3 rings AeroRing network

- The 6-rings topology, described in Fig. 6.2 and Table 6.3, where we replace each AFDX switch by a peripheral ring;
- The 4-rings topology as described in Fig. 6.3 and Table 6.3, where switches SW3 and SW4 are replaced each by a peripheral ring, whereas switches SW1 and SW7 (resp. SW2 and SW8) are grouped within the same peripheral ring;
- The 3-rings topology as described in Fig. 6.4 and Table 6.3, where each couple of switches among (SW1, SW2), (SW3, SW7) and (SW4, SW8) is replaced by one peripheral ring;
- The mono-ring topology, where all the end-systems of the AFDX switches are gathered in the same ring.

In addition, the considered service policy for all configurations is SP.

Scenario 3 in this scenario, we compute the reliability level offered by AeroRing for the considered case of study. For this, we consider the different AeroRing topologies described in scenario 2. In addition, we integrate the impact of the physical redundancy, i.e., the network is fully redundant.

6.1.1 AeroRing vs AFDX and RTE Solutions

In this section, we report the results of the comparison between AeroRing, AFDX and the RTE solutions when considering scenario 1.

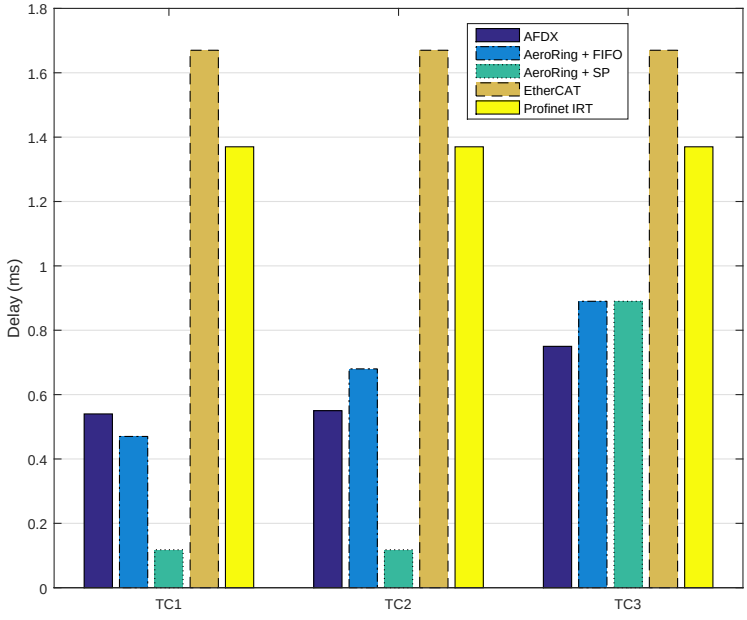


Figure 6.5: Maximum end-to-end delay bounds per traffic class

Fig. 6.5 shows the maximum end-to-end delay bounds of the different traffic classes for the current AFDX network, AeroRing with both service policies, EtherCAT and Profinet IRT. As we can notice, all the solutions respect the temporal constraints of the different traffic classes and AeroRing outperforms the RTE solutions, i.e., EtherCAT and Profinet IRT, with both service policies. In addition, AeroRing with SP service policy outperforms AFDX for TC1 and TC2, and offers a slightly higher delay bound for TC3, in comparison with the AFDX. For instance, AeroRing with SP policy offers a delay bound for TC1 4.57, 14 and 11.6 times less than the AFDX, the EtherCAT and the Profinet IRT, respectively. These results show the high timing performance of AeroRing with reference to AFDX and the most relevant RTE solutions.

Using the SP policy allows to serve the higher priority levels before the lower ones, which allows to reduce the interference, thus reduces the delays, in comparison with the FIFO case. The maximum delay bound for TC1 (resp. TC2) is 4 (resp. 5.7) times less with the SP policy than FIFO.

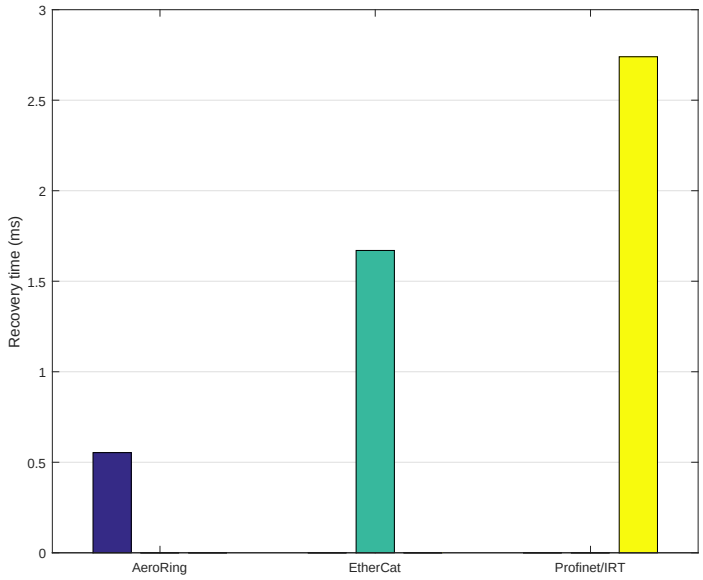


Figure 6.6: Maximum recovery time

In addition, Fig. 6.6 shows a comparison of the maximum recovery time under AeroRing and the different RTE solutions. As we can see, AeroRing offers the lower recovery time, e.g., the recovery time for AeroRing, EtherCAT and Profinet IRT is 0.55ms, 1.67ms and 2.74ms, respectively. This result shows the high availability of AeroRing, which is mainly due to the dynamic redundancy mechanism ARRP. First, the local fault detection mechanism implemented within AeroRing ensures a faster fault detection time than centralized or global fault detection mechanisms, where control messages need to cross all the network, as implemented within EtherCAT and Profinet IRT. Moreover, using a single control message, with the highest priority and sent by the node detecting the failure, allows to reduce the recovery time under AeroRing.

6.1.2 Mono-ring vs Multiple-ring Topologies

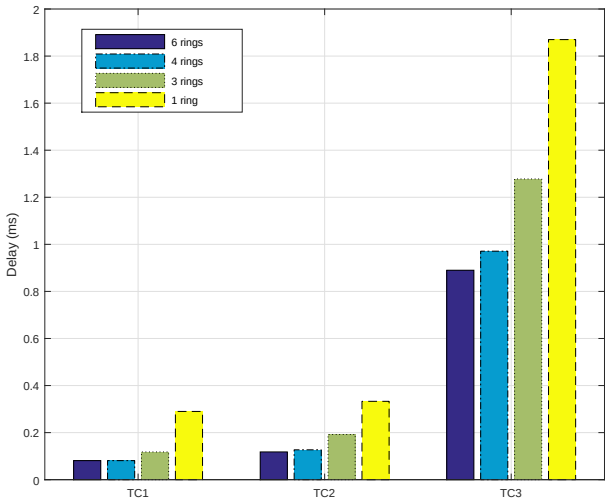


Figure 6.7: Maximum end-to-end delay bounds per TC for different AeroRing topologies

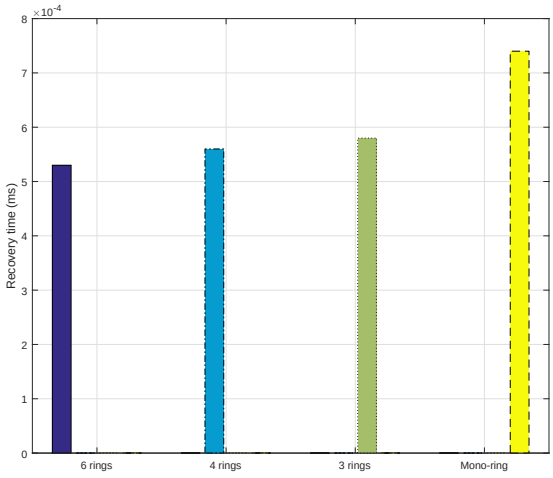


Figure 6.8: Maximum recovery time for different AeroRing topologies

We consider herein the results of scenario 2. Fig. D.15 and 6.8 show the maximum end-to-end delay bounds of the different traffic classes and the recovery time for different AeroRing topologies, respectively. As we can see in Fig D.15, all the configurations respect the flows deadlines. Moreover, we notice that the end-to-end delay bounds and recovery time increase when reducing the number of peripheral rings, i.e., increasing the peripheral rings size. This fact is due to the increasing number of crossed nodes when the peripheral ring size increases, which increases the interferences.

6.1.3 AeroRing Reliability

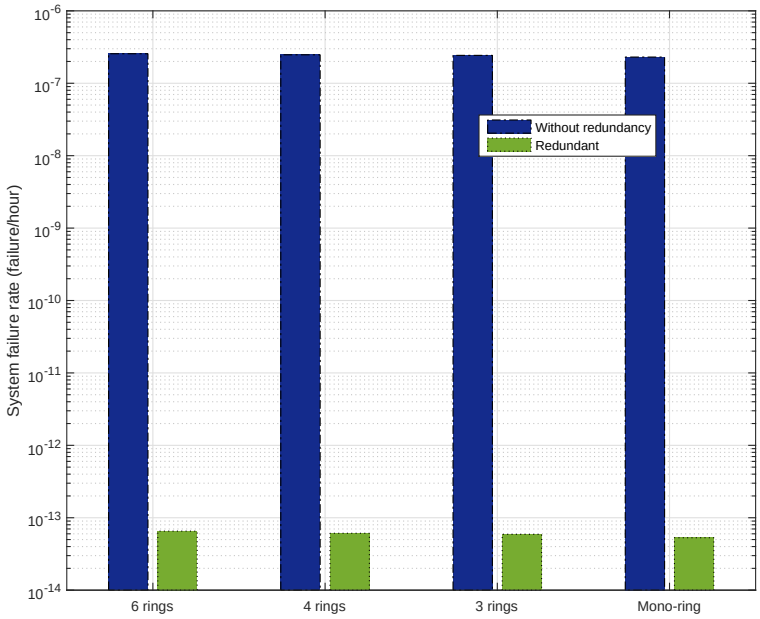


Figure 6.9: AeroRing reliability for the different topologies

Fig. 6.9 shows the system failure rate of an AeroRing network with or without redundancy, according to the network topology when considering scenario 3. As we can see, a redundant AeroRing network offers a high reliability level with a failure rate less than 10^{-13} under various topologies, which satisfies the required avionics DAL-A level, i.e., failure rate less than 10^{-9}. Moreover, we can notice that the different network topologies have similar reliability levels.

6.2 Generic Case Study

In this section, we use the same representative case study described in Section 2.3.2, which has been used to benchmark the main RTE solutions. This analysis will allow us to compare AeroRing performance with those of RTE solutions. It is worth noting that the considered AeroRing topology in this case study is the mono-ring one.

Fig. 6.10 illustrates the maximum end-to-end delay bounds with the different RTE solutions. We observe that Ethernet/IP has the highest delivery time, in comparison to the rest of the solutions, which have quite similar performance for I/O data. However, AeroRing is more scalable since it allows to connect more end-systems, while respecting the most constrained deadline, i.e., I/O deadline of 2ms. The maximum number of RTE end-systems respecting the I/O deadline is about 8, 70, 76 and 81 for Ethernet/IP, EtherCAT, Profinet/IRT and AeroRing, respectively. This result shows the high scalability of AeroRing.

Concerning resource efficiency, we can observe Figures 6.11 and 6.12 illustrating the RTE throughput for the different types of traffic and the Non-RTE bandwidth, respectively.

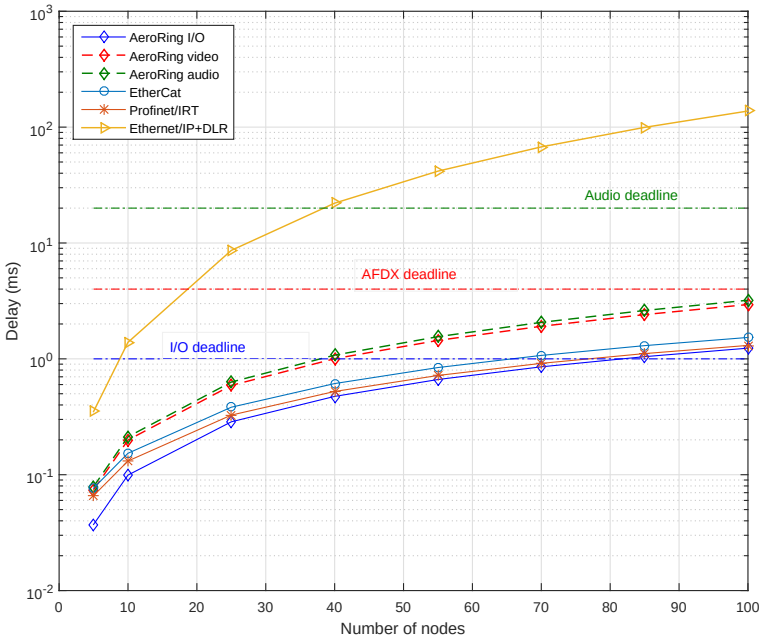
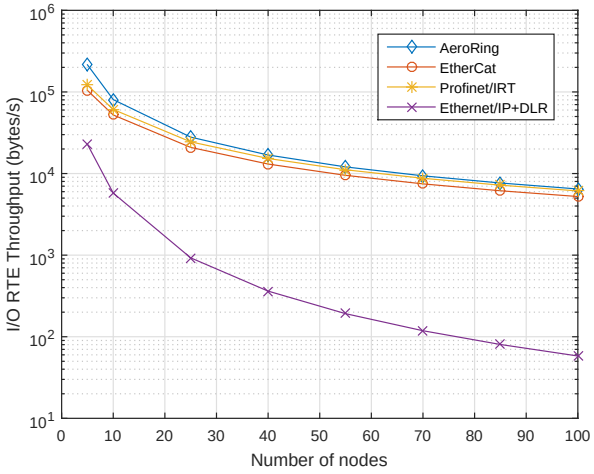


Figure 6.10: Maximum Delivery Time of Ring-based RTE solutions

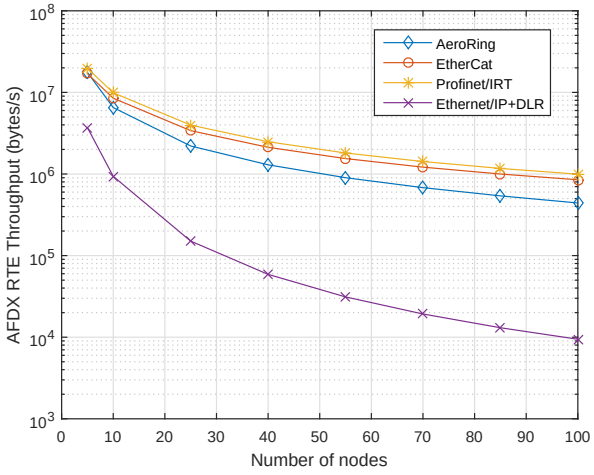
The obtained results show the high resource efficiency of AeroRing, in comparaison with the main RTE solutions, in terms of RTE throughput and non RTE bandwidth. This fact is mainly related to the QoS management within AeroRing, which enhances the highest priority delay bound; thus, a better NRT bandwidth.

Finally, to compare the availability level of the different RTE solutions with the AeroRing one, the maximum recovery time is shown in Fig. 6.13. As we can see, AeroRing outperforms the existing RTE solutions due to its efficient fault detection mechanisms.

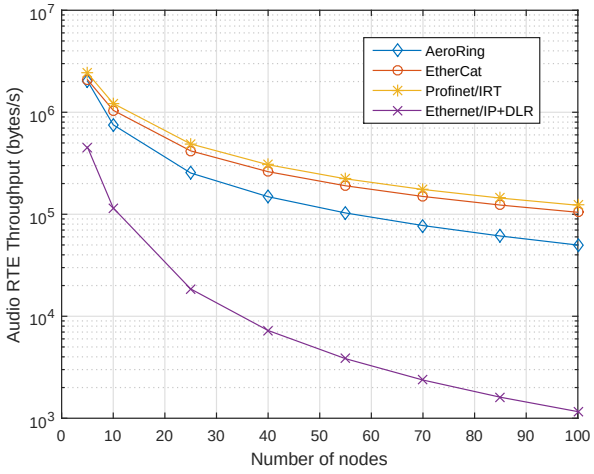
Based on these results, we conclude that AeroRing guarantees the best predictability and availability levels, with reference to the main existing RTE solutions, since it offers the lowest delay bounds for the most constrained traffic, i.e., I/O and recovery time.



(a)



(b)



(c)

Figure 6.11: RTE Throughput of Ring-based RTE solutions: (a) I/O traffic; (b) AFDX traffic; (c) Audio traffic

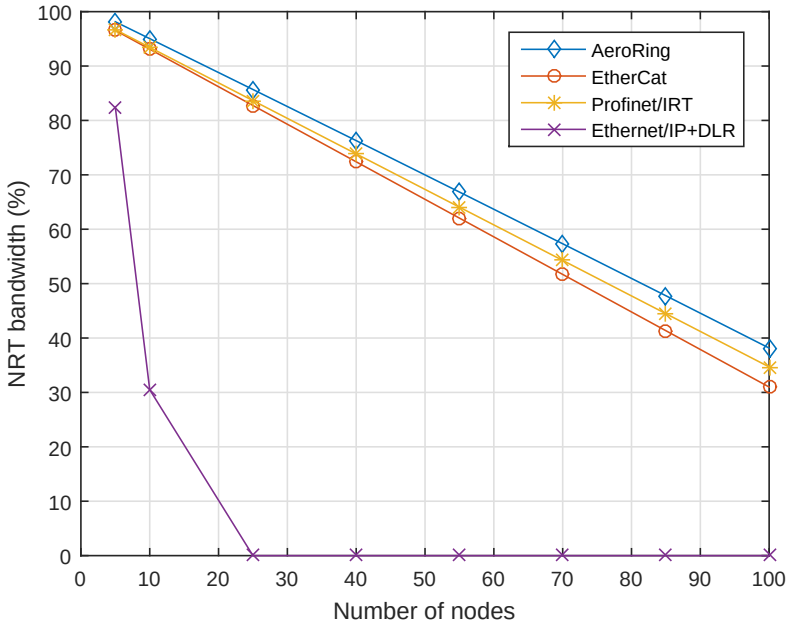


Figure 6.12: Non-RTE Bandwidth of Ring-based RTE solutions

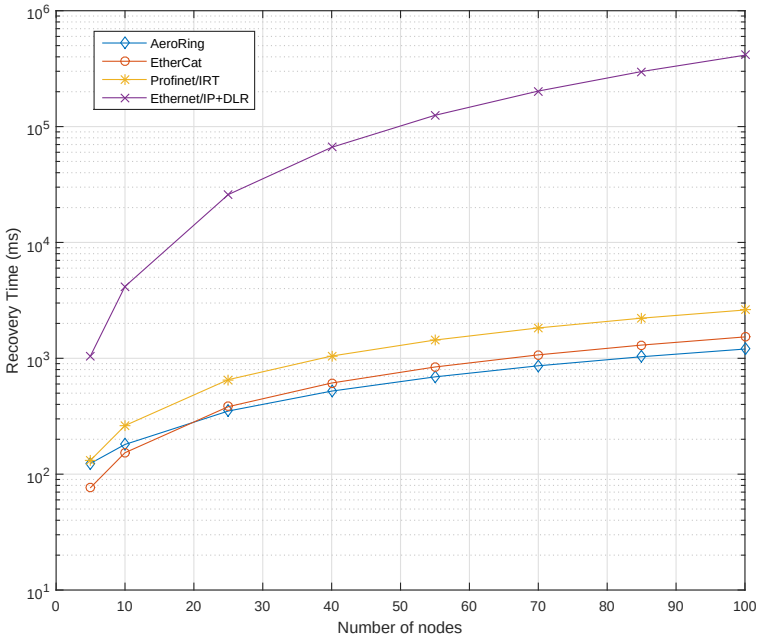


Figure 6.13: Redundancy Recovery Time of Ring-based RTE solutions

6.3 Conclusion

In this chapter, the validation of AeroRing has been conducted through a realistic avionics case study under several network configurations. AeroRing performances were compared with reference to the current AFDX network and the most relevant RTE solutions. Obtained results show the high predictability, reliability and availability levels of AeroRing. First, AeroRing offers the lowest delay bounds, in comparison with the AFDX and the main RTE solutions. Second, due to its efficient dynamic redundancy mechanisms (ARRP), it offers the highest availability level, since it offers the lowest recovery time. Finally, AeroRing offers a high reliability level, satisfying the required DAL-A level for avionics, under different network topologies.

Chapter 7

Conclusions and Perspectives

Contents

7.1 Conclusions

Although the current avionics communication architecture fulfills the main avionics requirements, it leads at the same time to a significant quantity of wires, which increases the system weight and costs. To handle these emerging needs, we proposed in this thesis the integration of a new avionics communication network, called AeroRing, based on the Gigabit Ethernet technology and a ring topology. The IEEE802.3 compliance of AeroRing guarantees its interoperability with the AFDX technology, which will facilitate its adoption in the market. Moreover, the ring-based topology decreases the cabling complexity, in comparaison with the switched topology, while allowing a high availability level due to the implicit redundant path. The integration of such a solution has many interesting benefits, in terms of reducing the weight and costs; however it infers at the same time many challenging issues to be adopted in avionics, mainly related to the predictability and reliability requirements. To achieve this aim, we pursued during this thesis a specific design methodology with the main following steps.

First, we ***designed AeroRing*** to integrate various features favouring the avionics requirements:

- To guarantee a high **modularity** level, we selected the event-triggered communication paradigm, similar to the AFDX one. This fact reduces the reconfiguration effort, but may introduce cyclic dependencies, which complexify the timing analysis. The latter was one of the main challenging issues during this thesis, that we handled based on an innovative analytical approach, as explained in Chapter 4;

- To favour the **predictability**, we integrated traffic shaping to prevent the network saturation and discard non-conformant data; and we selected the static priority policy to manage 3 data priority levels, i.e., HRT, SRT and NRT. Furthermore, we defined a QoS-aware routing algorithm, which infers the transmission of HRT data on both possible paths to enhance the reliability, and the transmission of SRT and NRT data only on the shortest path to enhance timing performance;

- To guarantee high **availability** level, we specified an innovative dynamic redundancy protocol, called ARRP, which:
  - unlike existing protocols takes advantages of the multi-path feature of ring topologies through implementing filtering functions within each T-AeroRing, instead of transforming the ring into line;
  - implements a local fault detection mechanisms based on control messages, enabling short recovery times; thus increasing the AeroRing availability level;
  - incorporates an auto-configuration mechanism to build dynamically the routing tables within the T-AeroRings, reducing the configuration overhead and effort;

- To guarantee high **reliability** level, AeroRing supports duplicated Multiple-ring topologies allowing to achieve the DAL A requirement, i.e., less than 10^{-9} F/h, as shown in Chapter 6.

Second, to ***prove the timing performance*** of AeroRing, we have proceeded as follows:

- We modelled our solution based on the Network Calculus formalism and we evaluated its timing performance based on the existing approaches in the literature, dealing with the cyclic dependencies problem. The first results showed the limitations of these methods, in terms of scalability, i.e., very limited number of nodes, and resource efficiency, i.e., low utilisation rate, to enable the temporal constraints guarantee;

- To handle the limitations of the existing approaches, we introduced a new approach based on Network Calculus, Pay Multiplexing Only at Convergence Points (PMOC), taking into account the flow serialisation phenomena along the shared paths and paying the bursts of interfering flows only at the convergence points. This method allowed the computation of tighter delay bounds; thus enhanced the scalability and resource-efficiency of AeroRing;

- We generalised the PMOC approach to the Multiple-Ring topologies and conducted sensitivity analysis to measure the impact of such topologies on AeroRing performances. The results showed that the Multiple-Ring topology can be highly efficient to improve timing performance, when the inter-ring communication load is limited.

Third, to ***assess the availability*** level of AeroRing:

- We defined the related Performance Indicators, i.e., fault detection time and redundancy recovery time, induced by the specified ARRP protocol;

- We computed these PIs for different AeroRing configurations regarding the network size and topology; and showed the high availability level of AeroRing, in comparison with the existing Real-Time Ethernet (RTE) solutions.

Fourth, to ***measure the reliability*** level of AeroRing, we conducted dependability analyses based on Stochastic Active Networks (SANs) and proceeded as follows:

- We modelled the different failure modes of AeroRing in the case of Mono-Ring topology, to compute the system failure rate based on the Mobius tool;

- We extended such a modelling to cover the various AeroRing topologies, i.e., simple and duplicated Mono and Multiple Ring topologies;

- We conducted a sensitivity analysis regarding various system parameters; and the results showed the high reliability level of AeroRing under specific conditions.

Finally, to have the proof of concept of AeroRing, we analysed the AeroRing performances, i.e., timing, availability and reliability levels, for a realistic avionics case. The results showed that:

- In terms of timing performance, AeroRing outperforms the main existing RTE solutions and the AFDX network;

- In terms of availability, AeroRing induces the shortest redundancy recovery time, in comparison to the main RTE solutions;

- In terms of reliability, the duplicated Multiple-Ring topology of AeroRing fulfills the DAL A requirements.

7.2 Perspectives

In the extent of our research work, we have identified some interesting topics at both practical and fundamental levels and at the short and average terms.

First, at the ***practical level***, we have distinguished the following emerging issues:

- At the **short term**, we are considering the standardisation of AeroRing as a new open real-time ethernet solution for safety-critical applications. The definition and evaluation of its main Performance Indicators, defined in the standards IEC 61784-[1-2] [88, 1], will definitely facilitate such a process;

- At the **average term**, we are planning to extend the application domains where AeroRing may be efficient, and more particularly smart factory. This extension will induce new challenges, in terms of scalability and adaptability features. Moreover, we believe that extending the AeroRing prototype to support Multiple-Ring topology will be of great interest, since conducting experiments on such topologies will deliver us more insights into AeroRing performances and will consolidate our theoretical results.

Second, at the ***fundamental level***, we have identified the following relevant challenges:

- At the **short term**, we are envisaging to extend the PMOC approach under First In First Out (FIFO) policy, to enhance the condition on the maximum utilisation rate and enable a full utilisation rate. Such an extension presents some hard points to handle, mainly related to the complexity of computing global residual service curves under FIFO, since only the residual curve within one node is available in the literature. Moreover, we believe that generalising the PMOC approach to support any Non-Feed Forward network, i.e., with cyclic dependencies, independently from the network topology will be of utmost importance to extend the applicability of the Network Calculus framework;

- At the **average term**, we are planning to conduct qualitative dependability analyses based on Model checking, to verify and prove formally the reliability and availability properties of AeroRing.

7.3 List of Publications

Proceeding of International Conferences

- **[ERTS2-16]** Amari, A., Mifdaoui, A., Frances, F., Lacan, J., Rambaud, D., and Urbain, L, "AeroRing: Avionics Full Duplex Ethernet Ring with High Availability and QoS Management", in European Congress on Embedded Real Time Software and systems 2016 (ERTS2'16).

- **[WFCS-16]** Amari, A., Mifdaoui, A., Frances, F., and Lacan, J. "Worst-case timing analysis of AeroRing - A Full Duplex Ethernet ring for safety-critical avionics", in IEEE International Workshop on Factory Communication Systems 2016 (WFCS'16).

- **[ETFA-17]** Mifdaoui, A. and Amari, A., "Real-Time Ethernet Solutions supporting Ring topology from an Avionics Perspective: a Short Survey", in IEEE international Conference on Emerging Technologies and Factory Automation 2017 (ETFA'17).

- **[RTCSA-17]** Amari, A. and Mifdaoui, A., "Worst-case Timing Analysis of Ring Networks with Cyclic Dependencies using Network Calculus", in IEEE International Conference on Embedded and Real-Time Computing Systems and Applications 2017 (RTCSA'17).

International Journals

- **[IES-18]** Amari, A., Mifdaoui, A., Frances, F., and Lacan, J., "AeroRing with Multiple Ring Topologies for New Generation Avionics Applications", in IEEE Transactions on industrial Informatics-Special issue on Embedded and Networked Systems for Intelligent Vehicles and Robots to appear in April 2018 (Under submission)

- **[RTS-17]** Amari, A., Mifdaoui, A., "Enhancing Performance Bounds of General Ring Networks with Cyclic Dependencies by Paying Multiplexing Only at Convergence Points", in Real Time Systems Journal 2017. (Under submission)

Appendix A

The 1000-BASE-T PHY sublayers and Degradation

Contents

A.1 The 1000-BASE-T PHY sublayers

In this section, the main features of the 1000BASE-T PHY layer are presented, as well as the different functions of its sublayers.

A.1.1 Physical Coding Sublayer (PCS)

The PCS sublayer receives the data from the upper layer, i.e., Data link layer, that converts them into symbols with the 4 D-PAM5 method. The symbols are sent continuously to the Physical Medium Attachment (PMA) to be sent out on the 4 pairs of the cable.

Code groups are used to control the flow of communication, report and correct errors. The different cases and options are detailed in [6]. The beginning and the end of a flow transmission is marked by some specific code groups. Between the transmission of two frames, each equipment sends to its neighbour information about the good functioning of its PHY layer. This phase is called the Idle mode. Moreover, the PCS Layer receives the symbols of the PMA sublayer and converts them into data bits to send them to the upper layer.

A.1.2 Physical Medium Attachment (PMA) Sublayer

The PMA sublayer allows to have a 125 M symbol/s full-duplex communication on each pair of the four pairs of the cable on a distance less than or equal to 100 meters.

The received symbols from the PCS sublayer are converted to analog signal to be sent on the medium using four transmitters, i.e., Analog Digital Converters (ADCs). Moreover, the PMA transforms the received analog signal into symbols using a receiver, i.e., Digital Analog Converter (DAC), and sends them to the PCS layer. The PMA layer allows also to control the link and to select an operating mode for the PHYs.

A.1.3 Auto-Negotiation (AUTONEG)

The 1000BASE-T PHY can operate in master or slave mode. During the auto negotiation phase, one of the two PHYs that shares the same medium goes into the master mode and the second into the slave mode. The master PHY uses its own clock for the transmission and synchronization, while the slave PHY regenerates the clock from the received signal.

The 1000BASE-T PHY communication passes through two modes:

- *Training Mode:* in this mode, the PCS sends only Idle codes until both PHYs are ready to operate in normal mode.
- *Normal Mode:* the PCS generates different types of symbols and codes (data symbols, Idle codes, flows markers ...etc).

A.2 Medium Degradation

Simultaneous transmission in full-duplex on four pairs of cable generates signal damage and interference. Fig. A.1 shows the main causes of interference.

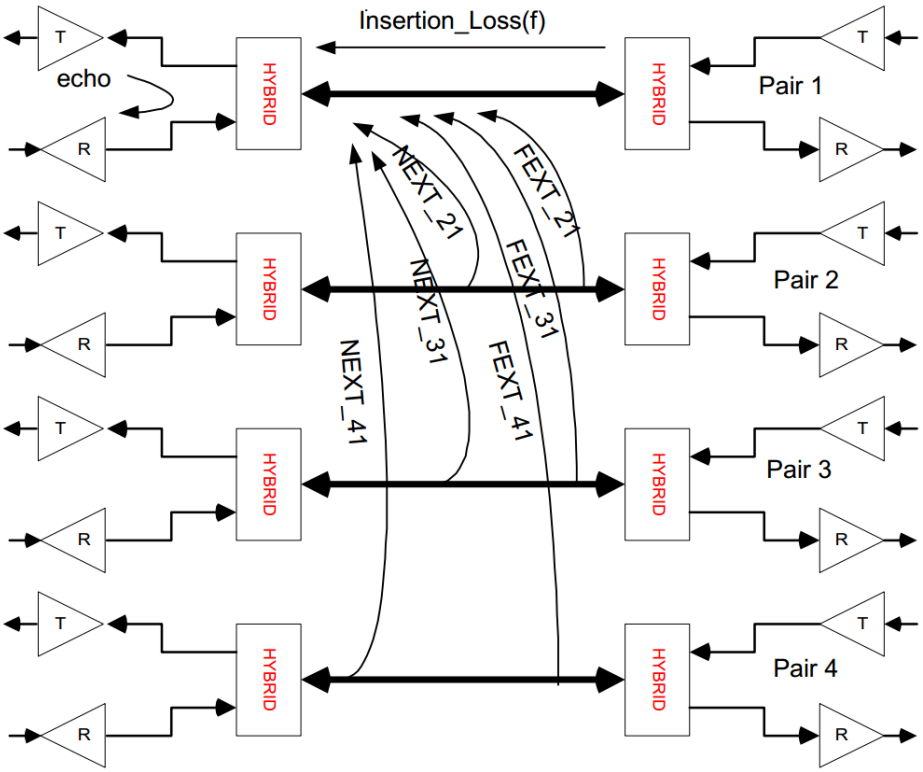


Figure A.1: Interference causes [7]

- Full-duplex mode: because of the simultaneous transmission in both directions on the pairs, the signals are mixed. A module called Hybrid allows to extract the received signal from the transmitted signal;
- Attenuation: it is the loss of signal in the cable between the transmitter and the receiver. This attenuation increases proportionally to the distance and frequency. The 4D-PAM5 modulation allows to reduce the frequency by a factor of two;
- Echo: The reflection of the transmitted signal to the transmitter;
- Crosstalk: An undesired signal is generated by the transmission of other adjacent pairs;
  - NEXT (Near-end crosstalk): interference between two pairs of a cable measured on the same side as the transmitter.
  - FEXT (Far - end crosstalk): interference between two pairs of a cable measured on the opposite side of the transmitter one.

Fig. A.2 shows a simplified diagram of cancellation methods of interfering signals. Transmissions on the pairs are in full-duplex. At the reception of a signal, the Hybrid module extracts the received signal from the transmitted one. Then, the signal is transformed into a digital signal using a ADC. This digital signal passes through several operations to delete the damage due to interfering signals (echo and crosstalk). These interfering signals are computed using mathematical equations from the original signals of the four pairs (more details in [6]). The data is then decoded by a Viterbi decoder and sent to the PCS sublayer.

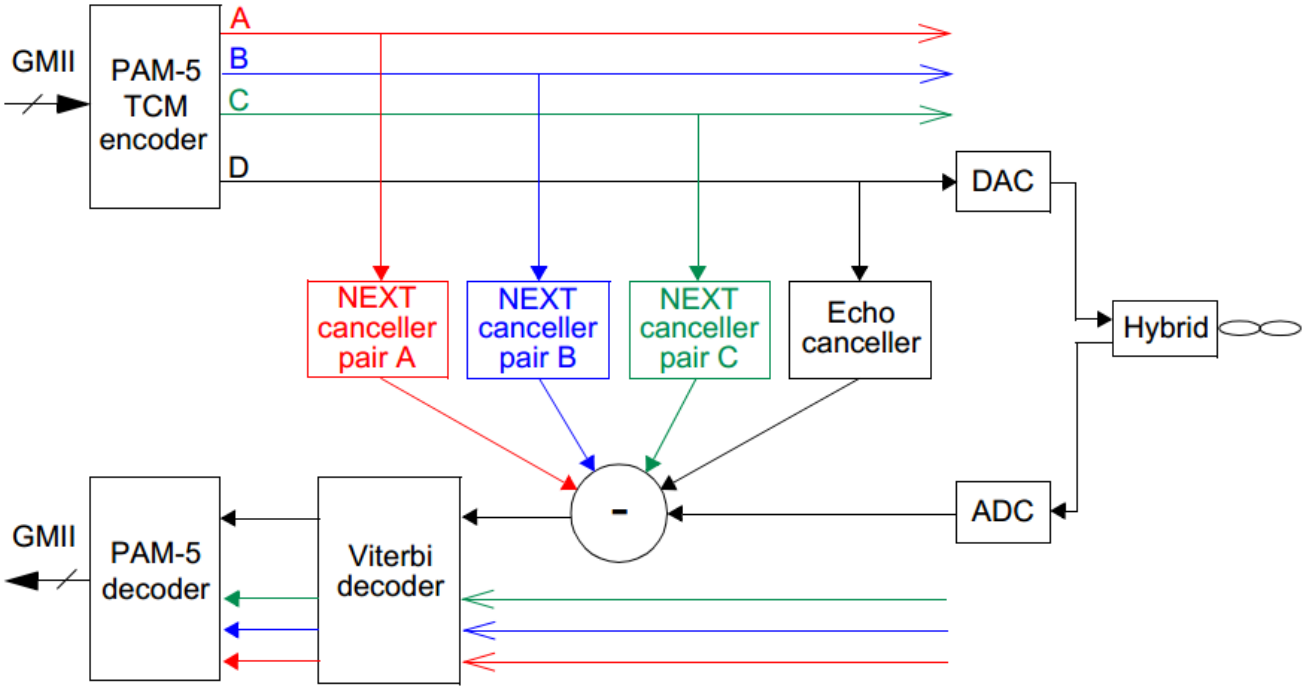


Figure A.2: A simplified diagram of interfering signals attenuation

Appendix B

Network Calculus Background

Contents

B.1 Network Calculus Background

An overview of the main principles of Network Calculus framework used in Chapter 4 are described herein. Further details on this framework can be found in two substantial books [20] and [89] and a noticeable survey [64]. The *Network Calculus* is a mathematical framework to derive maximum bounds on system performance, such as delays, backlogs or throughput. This framework has been founded by the seminal work of Cruz in [90, 22], and then extended with min-plus Algebra operations in [89] and [20]. The latter extension is based on the idea of modeling the communication nodes as in conventional system theory, with an input function, a transfer function and an output function, where addition and multiplication are replaced by minimum and addition, respectively.

In this appendix, we will try to answer some primordial questions when applying Network Calculus to conduct performance analysis of a realistic network. The first question concerns modeling the input traffic and is detailed in Section B.2. Then, the second is about modeling the node specifications to consider its impact on the system performance, which is presented in Section B.3. Finally, how to deal with a network of nodes to compute end-to-end performance, and the details are given in Section B.4.

B.2 Traffic Model

Network Calculus describes data flows by means of cumulative functions, defined as the number of transmitted bits during the time interval $[0, t]$. These functions are non negative and wide sense increasing:

$$\mathscr{F} = \{f : \mathbb{R}^+ \to \mathbb{R}^+ \mid f(0) = 0, \forall\, t \geq s : f(t) \geq f(s)\}$$

Consider a system S receiving input data flow with a Cumulative Arrival Function (CAF), $A(t)$, and putting out the same data flow with a Cumulative Departure Function (CDF), $D(t)$. Furthermore, S fulfills the causality condition, i.e., $\forall\, t \in \mathbb{R}^+$, $A(t) \geq D(t)$. These functions allow computing the main performance metrics, defined as:

Definition 2. *The flow backlog at time t is:*

$$q(t) = A(t) - D(t)$$

Definition 3. *The flow virtual delay at time t is:*

$$d(t) = \inf\{\tau \geq 0 : A(t) \leq D(t+\tau)\}$$

The backlog $q(t)$ and virtual delay $d(t)$ are simply the vertical and horizontal distances between the CAF and the CDF at instant t, respectively. To compute upper bounds on the worst case delay and backlog, we need to introduce one of the most fundamental concepts in Network Calculus, the maximum arrival curve. This curve provides an upper bound on the number of events, e.g., bits or packets, observed during any interval of time. This concept allows modeling a large panel of event arrival patterns, such as periodic, sporadic, with or without jitter or burst.

Definition 4. *(Arrival Curve)[20] A function α is an arrival curve for a data flow with the CAF A, iff:*

$$\forall t, s \geq 0, s \leq t, A(t) - A(s) \leq \alpha(t-s)$$

The arrival pattern necessary to define the maximum arrival curve can be obtained from traffic traces if any, or application specification. The latter is more common for real-time communication networks. The network designer generally specifies a traffic contract for each application, enforced using a leaky-bucket shaper, which guarantees for the controlled traffic a maximum burst σ and a maximum rate ρ, i.e., the traffic flow is (σ, ρ)-constrained. In this case, the arrival curve is a concave affine curve, defined as $\gamma_{\sigma,\rho}(t) = \sigma + \rho.t$ for $t > 0$.

B.3 Node Model

To conduct worst-case performance analysis, we need to put constraints on the input traffic through the maximum arrival curve notion. In return, we need to guarantee a minimum offered service within crossed nodes to cover the worst-case behavior and infer upper bounds on performance metrics, e.g., backlog and delay. This is done through the concept of minimum service curve, which has been defined for the first time in the seminal work [91] and more recently adapted in [20] as following.

Definition 5. *(Simple Minimum Service Curve) The function β is the simple service curve for a data flow with the CAF A and the CDF D, iff:*

$$\forall t \geq 0, D(t) \geq \inf_{s \leq t}(A(t) + \beta(t-s))$$

A very useful and common model of service curve is the rate-latency curve $\beta_{R,T}$, with R the minimum guaranteed rate and T the maximum latency before starting the service. This rate-latency function is defined as follows:

$$\beta_{R,T}(t) = [R(t-T)]^+$$

Where $[x]^+$ is the maximum between x and 0. This service curve is easy to define in the case of ***one input/output node*** serving one or many traffic flows coming from the same source and going to the same destination. However, to handle more realistic scenario with a network of nodes, implementing aggregate scheduling, which multiplexes the crossing flows at the input and demultiplexes them at the output, we need to define the ***left-over service curve*** guaranteed to each traffic flow within each crossed node, considering the impact of the other traffic flows in contention, to infer the offered guarantees for each flow. The computation of such a left-over service curve depends on the implemented scheduling policy within each crossed node, and the most common ones are Blind Multiplexing, FIFO and Fixed Priority (FP). It is worth noting that this derivation needs strict service curve property in the general case, except for FIFO and Constant bit rate nodes.

Definition 6. *(Strict service curve) The function β is a strict service curve for a data flow with the CDF $D(t)$, if for any backlogged period[a] $]s,t]$, $D(t)-D(s) \geq \beta(t-s)$.*

[a] A backlogged period $]s,t]$ is an interval of time during which the backlog is non null, i.e., $A(s) = D(s)$ and $\forall u \in]s,t]$, $A(u) - D(u) > 0$

The main results concerning the left-over service curves computation are as follows:

Theorem 2. *(Left-over service curve - Arbitrary Multiplex)[92] let f_1 and f_2 be two flows crossing a server that offers a strict service curve β such that f_1 is α_1-constrained, then the left-over service curve offered to f_2 is:*

$$\beta_2 = (\beta - \alpha_1)_\uparrow$$

where $f_\uparrow(t) = \max\{0, \sup_{0 \leq s \leq t} f(s)\}$

Corollary 6. *(Left-over service curve - FP Multiplex)[93] Consider a system with the strict service β and m flows crossing it, $f_1, f_2, .., f_m$. The maximum packet length of f_i is $l_{i,max}$ and f_i is α_i-constrained. The flows are scheduled by the non-preemptive fixed priority (NP-FP) policy, where $priority\ f_i > priority\ f_j \Leftrightarrow i < j$. For each $i \in \{2, .., m\}$, the strict service curve of f_i is given by:*

$$(\beta - \sum_{j<i} \alpha_j - \max_{k \geq i} l_{k,max})_\uparrow$$

B.4 Performance Analysis

Knowing the arrival and service curves, one may compute the upper bounds on performance metrics for a data flow. Before detailing the main theorems in this part, let us define the main algebraic operations in Network Calculus, i.e., convolution and deconvolution of two functions $f, g \in \mathscr{F}$:

- min-plus convolution:

$$f \otimes g(t) = \inf_{0 \leq s \leq t} \{f(s) + g(t-s)\}$$

- min-plus deconvolution:

$$f \oslash g(t) = \sup_{\forall u \geq 0} \{f(t+u) - g(u)\}$$

For a node with ***one input/output***, these bounds are computed according to the following theorem.

Theorem 3. *(Performance Bounds) Consider a flow constrained by an arrival curve α crossing a system $\mathscr{S}$ that offers a service curve β. The performance bounds obtained at any time t are given by:*
Output arrival curve: $\alpha^*(t) = \alpha \oslash \beta(t)$
Backlog[a]*:* $\forall\ t:\ q(t) \leq (\alpha \oslash \beta)(0) =: v(\alpha, \beta)$
Delay[b]*:* $\forall\ t:\ d(t) \leq \inf\{t \geq 0 : (\alpha \oslash \beta)(-t) \leq 0\} =: h(\alpha, \beta)$

[a] $v(f,g)$: the maximum vertical distance between f and g
[b] $h(f,g)$: the maximum horizontal distance between f and g

The calculus of these bounds is greatly simplified in the case of a leaky bucket arrival curve and a rate-latency service curve. In this case, the delay and backlog are bounded by $\frac{b}{R} + T$ and $b + r * T$, respectively; and the output arrival curve is $b + r(T + t)$.

Afterwards, to extend this result to a ***network of nodes***, one of the strongest result in the Network Calculus framework is the computation of an end-to-end service curve for a tandem of nodes crossed by the same flows. This curve is computed as the convolution of residual service curves in each node, and is used to infer end-to-end performance bounds according to Th. 3. This result is described in the following theorem.

Theorem 4. *(Concatenation-Pay Bursts Only Once) Assume a flow crossing two servers with respective service curves β_1 and β_2. The system composed of the concatenation of the two servers offers a minimum service curve $\beta_1 \otimes \beta_2$ to the flow.*

As an example, for a tandem of nodes with rate-latency service curves, the end-to-end service curve computed according to Th. 4 is also a rate-latency curve, where the rate is the minimum of the crossed node rates and the latency is the sum of their latencies.

This result infer an interesting property known as "Pay bursts Only Once Phenomena". Indeed, the end-to-end delay bound for a data flow, computed using the end-to-end service curve obtained with Th. 4, clearly outperforms the sum of delay bound per node, computed iteratively using Th. 3 and denoted as additive delay bound. The computation of these two bounds show the appearance of the burst term many times in the additive delay bound, as opposed to only once for the other. More recently, the authors in [67] propose an innovative approach, denoted as Pay Multiplex Only Once (PMOO), and the main idea is based on accounting the flow serialization phenomena along the flow path to compute tighter end-to-end delay bound. However, the latter has been proved under blind multiplexing property, which may induce overly pessimistic bounds under FIFO and FP policies.

Appendix C

PMOC Proofs

Contents

C.1 Proof of Theorem 1

Proof. As explained in Section 4.3.1, for any flow i crossing the ring network, there are only two possible convergence points with a f.o.i f: $f.ft$ and $i.ft$. This fact infers three possible categories for an interfering flow i with the f.o.i f: (i) *category 1*: having only one convergence point with f, which is its first hop, i.e., $i.ft$; (ii) *category 2* having only one convergence point with f, which is the first hop of f, i.e., $f.ft$; (iii) *category 3* having two distinct convergence points with f, i.e., $i.ft$ and $f.ft$ if $i.ft \neq f.ft$.

Let's illustrate these three categories with the example of Fig. 4.5. If we consider flow f_1 as the f.o.i, then flows f_2, f_4 and f_3 are in categories 1, 2 and 3, respectively.

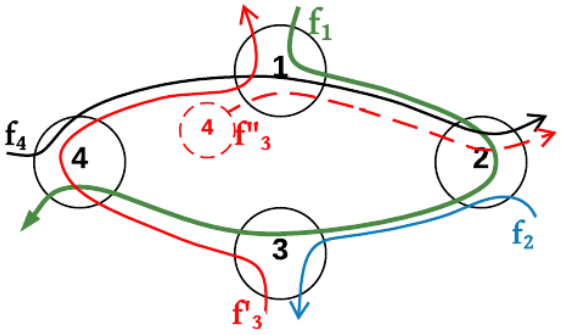


Figure C.1: Cutting virtually the flows of Fig. 4.5

To prove the Th. 1, we need to model an interfering flow i of category 3 by splitting it in two subflows to cut virtually the cyclic dependency with the f.o.i f, as illustrated in Fig. C.1 for flow f_3: (i) $i1$: the subflow of i along its subpath $\mathbb{P}_{i1} = (0, i.ft, i.ft \oplus 1, ..., f.ft \ominus 1)$, which is (σ_i^0, ρ_i)-constrained; (ii) $i2$: the subflow of i along its subpath $\mathbb{P}_{i2} = (f.ft \ominus 1, f.ft, ..., i.ft \oplus (h_i - 1))$, which is $(\sigma_i^{f.ft \ominus 1}, \rho_i)$-constrained. It is worth noting that $i1$ fulfills the conditions of category 1, whereas $i2$ fulfills the ones of category 2. Thus, splitting virtually the flows of category 3 in $\mathbb{K}_f(n)$ in two subflows leads to a transformed set $\overline{\mathbb{K}_f(n)}$. The latter can be rewritten according to the conditions of categories 1 and 2 as follows:

$$\overline{\mathbb{K}_f(n)} = \{i \in \overline{\mathbb{K}_f(n)} / f \ni i.ft\} \cup \{i \in \overline{\mathbb{K}_f(n)} / i \ni f.ft, i.ft \neq f.ft\}$$

Let's explicit $\overline{\mathbb{K}_f(n)}$ through the example of Fig. 4.5. For the f.o.i f_1, the only flow of category 3 is the flow f_3. So, f_3 is virtually splitted as $(f'_3, f"_3)$ as shown in Fig. C.1, where $\mathbb{P}_{f'_3} = \{0,3,4\}$ and $\mathbb{P}_{f"_3} = \{4,1\}$. It is worth noting that according to this model, the virtual node representing the source of flow $f"_3$ is node 4. Moreover, the set of interfering flows with the f.o.i f_1, $\mathbb{K}_{f_1}(3)$, is transformed to $\overline{\mathbb{K}_{f_1}(3)} = \{f_2, f'_3\} \cup \{f_4, f"_3\}$.

Consider a flow of interest f with a subpath $\mathbb{P}_f(n)$. Any crossed node $l \in \mathbb{P}_f(n)$ admits a strict service curve. Hence, according to Def. 6, for any instant $t_l \geq 0$, there exists $t_{l\ominus 1} \leq t_l$ the start of the backlogged period such that:

$$D_f^l(t_l) - D_f^l(t_{l\ominus 1}) + \sum_{i \ni l, i \neq f} (D_i^l(t_l) - D_i^l(t_{l\ominus 1})) \geq \beta^l(\Delta_l) \tag{C.1}$$

where $\Delta_l = t_l - t_{l\ominus 1}$. The time indices are chosen to match the node indices. Then, we sum up the expression in Eq. (C.1) when varying $l \in \mathbb{P}_f(n)$, which infers:

$$\begin{aligned} & \sum_{l\in\mathbb{P}_f(n)} D_f^l(t_l) - D_f^l(t_{l\ominus 1}) \\ \geq & \sum_{l\in\mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{l\in\mathbb{P}_f(n)} \sum_{i\ni l, i\neq f} (D_i^l(t_l) - D_i^l(t_{l\ominus 1})) \end{aligned} \tag{C.2}$$

Knowing the definition of $\overline{\mathbb{K}_f(n)}$, we have:

$$\sum_{l\in\mathbb{P}_f(n)} \sum_{i\ni l, i\neq f} \Leftrightarrow \sum_{i\in\overline{\mathbb{K}_f(n)}} \sum_{l\in\mathbb{P}_f(n)\cap\mathbb{P}_i}$$

Moreover, at the start of a backlogged period s, we have $D_f^{i\oplus 1}(s) = A_f^{i\oplus 1}(s)$, and because of the ring topology, we have $A_f^{i\oplus 1}(s) = D_f^i(s)$; thus, $D_f^{i\oplus 1}(s) = D_f^i(s)$. Consequently, Eq . (C.2) can be simplified as follows:

$$\begin{aligned} & \sum_{l\in\mathbb{P}_f(n)} D_f^l(t_l) - D_f^l(t_{l\ominus 1}) \\ = \; & D_f^{f.ft}(t_{f.ft}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \\ + \; & D_f^{f.ft\oplus 1}(t_{f.ft\oplus 1}) - D_f^{f.ft\oplus 1}(t_{f.ft}) \\ \ldots & \\ + \; & D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-2)}) \\ = \; & D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \\ \geq \; & \sum_{l\in\mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{i\in\overline{\mathbb{K}_f(n)}} \sum_{l\in\mathbb{P}_f(n)\cap\mathbb{P}_i} (D_i^l(t_l) - D_i^l(t_{l\ominus 1})) \end{aligned} \tag{C.3}$$

Based on the definitions of $Mft(i,f,n)$ and $Mlt(i,f,n)$ in Tab. 4.1, Eq. (C.3) can be rewritten as follows:

$$\begin{aligned} & D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \\ \geq \; & \sum_{l\in\mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{i\in\overline{\mathbb{K}_f(n)}} D_i^{Mlt(i,f,n)}(t_{Mlt(i,f,n)}) - D_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1}) \\ \geq \; & \sum_{l\in\mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{i\in\overline{\mathbb{K}_f(n)}} A_i^{Mlt(i,f,n)}(t_{Mlt(i,f,n)}) - A_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1}) \\ \geq \; & \sum_{l\in\mathbb{P}_f(n)} \beta^l(\Delta_l) - \sum_{i\in\overline{\mathbb{K}_f(n)}} \alpha_i^{Mft(i,f,n)\ominus 1}\Big(\sum_{l=Mft(i,f,n)}^{Mlt(i,f,n)} \Delta_l\Big) \end{aligned} \tag{C.4}$$

To substitute the cumulative traffic functions of flows in $\overline{\mathbb{K}_f(n)}$ in Eq. (C.4) by their arrival curves, we have used the causality constraint of cumulative traffic functions, i.e., $\forall t, A_i^k(t) \geq D_i^k(t)$ and the property of the start of backlogged period at $t_{Mft(i,f,n)\ominus 1}$, i.e., $D_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1}) = A_i^{Mft(i,f,n)}(t_{Mft(i,f,n)\ominus 1})$.

On the other hand, rewriting the input arrival curve of a flow i at node k, $\alpha_i^{k\ominus 1}$, using

$\overline{\alpha_i}(\Delta_l) = \rho_i \Delta_l$, infers:

$$\begin{aligned}
\alpha_i^{k\ominus 1}(\sum_{l=1}^{m} \Delta_l) &= \sigma_i^{k\ominus 1} + \rho_i \sum_{l=1}^{m} \Delta_l \\
&= \sigma_i^{k\ominus 1} + \rho_i \Delta_1 + \rho_i \sum_{l=2}^{m} \Delta_l \\
&= \alpha_i^{k\ominus 1}(\Delta_1) + \sum_{l=2}^{m} \overline{\alpha_i}(\Delta_l) \qquad \text{(C.5)}
\end{aligned}$$

Hence, Eq. (C.4) can be rewritten using Eq. (C.5) as follows:

$$\begin{aligned}
& D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \qquad \text{(C.6)} \\
\geq & \sum_{l\in\mathbb{P}_f(n)} [\beta^l(\Delta_l) - \sum_{i\ni l, i\neq f} \alpha_i^{l\ominus 1}(\Delta_l).1_{\{l=Mft(i,f,n)\}} + \overline{\alpha_i}(\Delta_l).1_{\{l\neq Mft(i,f,n)\}}] \\
\geq & \sum_{l\in\mathbb{P}_f(n)} [(R^l - \sum_{i\ni l, i\neq f} \rho_i).(\Delta_l - T^l - \frac{\sum\limits_{i\ni l, i\neq f} \sigma_i^{Mft(i,f,n)\ominus 1} + T^l.\sum\limits_{i\ni l, i\neq f} \rho_i}{R^l - \sum\limits_{i\ni l, i\neq f} \rho_i})]^+ \\
\geq & \min_{l\in\mathbb{P}_f(n)} (R^l - \sum_{i\ni l, i\neq f} \rho_i).[\sum_{l\in\mathbb{P}_f(n)} \Delta_l - \sum_{l\in\mathbb{P}_f(n)} T^l - \sum_{l\in\mathbb{P}_f(n)} \frac{\sum\limits_{i\ni l, i\neq f} \sigma_i^{Mft(i,f,n)\ominus 1} + T^l.\sum\limits_{i\ni l, i\neq f} \rho_i}{R^l - \sum\limits_{i\ni l, i\neq f} \rho_i})]^+
\end{aligned}$$

Knowing the definition of $\overline{\mathbb{K}_f(n)}$, we can easily verify that

$$\sum_{l\in\mathbb{P}_f(n)} T^l.\sum_{i\ni l, i\neq f} \rho_i \Leftrightarrow \sum_{i\in\overline{\mathbb{K}_f(n)}} \rho_i.\sum_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j$$

Hence, Eq. (C.6) becomes:

$$\begin{aligned}
& D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \qquad \text{(C.7)} \\
& \geq \min_{l\in\mathbb{P}_f(n)} (R^l - \sum_{i\ni l, i\neq f} \rho_i).[t_{f.ft\oplus(n-1)} - t_{f.ft\ominus 1} - \sum_{l\in\mathbb{P}_f(n)} T^l \\
& - \sum_{i\in\overline{\mathbb{K}_f(n)}} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i.\sum\limits_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min\limits_{l\in\mathbb{P}_f(n)} (R^l - \sum\limits_{i\ni l, i\neq f} \rho_i)}]^+ \\
& \geq \min_{l\in\mathbb{P}_f(n)} (R^l - \sum_{i\ni l, i\neq f} \rho_i).[t_{f.ft\oplus(n-1)} - t_{f.ft\ominus 1} - \sum_{k\in\mathbb{P}_f(n)} T^k \\
& - \sum_{i\in\overline{\mathbb{K}_f(n)}, f\ni i.ft} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i \cdot \sum\limits_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min\limits_{l\in\mathbb{P}_f(n)} (R^l - \sum\limits_{i\ni l, i\neq f} \rho_i)} \\
& - \sum_{\substack{i\in\overline{\mathbb{K}_f(n)} \\ i\ni f.ft \\ i.ft\neq f.ft}} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i \cdot \sum\limits_{j\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^j}{\min\limits_{l\in\mathbb{P}_f(n)} (R^l - \sum\limits_{i\ni l, i\neq f} \rho_i)}]^+
\end{aligned}$$

Moreover, for each interfering flow i in category 3 splitted as (i_1, i_2), with i_1 and i_2 in categories 1 and 2, we have:

$$\begin{aligned} & \sigma_{i1}^{Mft(i_1,f,n)\ominus 1} + \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_{i1}} T^j \\ + \quad & \sigma_{i2}^{Mft(i_2,f,n)\ominus 1} + \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_{i2}} T^j \\ = \quad & \sigma_i^{i.ft\ominus 1} + \sigma_i^{f.ft\ominus 1} + \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap (\mathbb{P}_{i1} \cup \mathbb{P}_{i2})} T^j \\ = \quad & \sigma_i^0 + \sigma_i^{f.ft\ominus 1} + \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j \end{aligned} \tag{C.8}$$

Using Eq. (C.8) and (C.7), we deduce:

$$\begin{aligned} R^{\mathbb{P}_f(n)} &= \min_{l \in \mathbb{P}_f(n)} (R^l - \sum_{i \ni l, i \neq f} \rho_i) \\ T^{\mathbb{P}_f(n)} &= \sum_{k \in \mathbb{P}_f(n)} T^k + \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^0 . 1_{\{f \ni i.ft\}} + \rho_i \cdot \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j}{R^{\mathbb{P}_f(n)}} \\ &+ \sum_{i \in \mathbb{K}_f(n)} \frac{\sigma_i^{f.ft\ominus 1} . 1_{\{i.ft \neq f.ft / i \ni f.ft\}}}{R^{\mathbb{P}_f(n)}} \end{aligned} \tag{C.9}$$

(C.10)

This finishes the proof of the theorem.

□

C.2 Proof of Corollary 4

Proof. To proof the Cor. 4, we consider the same assumptions and notations considered in Section 4.1 with a multiple-ring topology. In general networks, an interfering flow i can converge with the f.o.i f in several convergence points along its subpath of length n, denoted $conv(i, f, n)$. We need to model these flows by splitting them into several subflows, one subflow at each convergence point. Each subflow i_k, $k \in conv(i, f, n)$ has a path $\mathbb{P}_{i_k}$ and it is $(\sigma_{i_k}^0, \rho_i)-$constrained, where $Mft(i_k, f, n) = i_k.ft = k$ and $\sigma_{i_k}^0 = \sigma_i^{k\ominus 1}$. Thus, splitting the interfering flows in $\mathbb{K}_f(n)$ leads to a transformed set $\overline{\mathbb{K}_f(n)}$.

We follow the same proof steps of of Th. 1 from Eq. (C.1) to Eq. (C.6) in Appendix C.1. Then, Knowing the definition of $\overline{\mathbb{K}_f(n)}$, we can easily verify that

$$\sum_{l \in \mathbb{P}_f(n)} T^l . \sum_{i \ni l, i \neq f} \rho_i \Leftrightarrow \sum_{i \in \overline{\mathbb{K}_f(n)}} \rho_i . \sum_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j$$

Hence, Eq. (C.6) becomes:

$$\begin{aligned} & D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \\ & \geq \min_{l \in \mathbb{P}_f(n)} (R^l - \sum_{i \ni l, i \neq f} \rho_i).[t_{f.ft\oplus(n-1)} - t_{f.ft\ominus 1} - \sum_{l \in \mathbb{P}_f(n)} T^l \end{aligned} \tag{C.11}$$

$$- \sum_{i \in \overline{\mathbb{K}_f(n)}} \frac{\sigma_i^{Mft(i,f,n)\ominus 1} + \rho_i . \sum\limits_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j}{\min\limits_{l \in \mathbb{P}_f(n)} (R^l - \sum\limits_{i \ni l, i \neq f} \rho_i)}]^+$$

We have, $\sum\limits_{i \in \overline{\mathbb{K}_f(n)}} \sigma_i^{Mft(i,f,n)\ominus 1} = \sum\limits_{i \in \overline{\mathbb{K}_f(n)}} \sigma_i^{i.first\ominus 1} = \sum\limits_{i \in \mathbb{K}_f(n)} \sum\limits_{k \in conv(i,f,n)} \sigma_i^{k\ominus 1}$. Furthermore, the common shared path between the flow of interest f and the original interfering flow i, i.e., $\mathbb{P}_f(n) \cap \mathbb{P}_i$, is equal to the shared path between the flow f and each sub-flow $i_k, k \in conv(i,f,n)$, i.e., $\mathbb{P}_f(n) \cap (\bigcup\limits_{k \in conv(i,f,n)} \mathbb{P}_{i_k})$. From this, Eq. (C.11) becomes:

$$\begin{aligned} & D_f^{f.ft\oplus(n-1)}(t_{f.ft\oplus(n-1)}) - D_f^{f.ft}(t_{f.ft\ominus 1}) \\ & \geq \min_{l \in \mathbb{P}_f(n)} (R^l - \sum_{i \ni l, i \neq f} \rho_i). \\ & [t_{f.ft\oplus(n-1)} - t_{f.ft\ominus 1} - \sum_{l \in \mathbb{P}_f(n)} T^l \\ & - \sum_{i \in \mathbb{K}_f(n)} \frac{\sum\limits_{k \in conv(i,f,n)} \sigma_i^{k\ominus 1} + \rho_i . \sum\limits_{j \in \mathbb{P}_f(n) \cap \mathbb{P}_i} T^j}{\min\limits_{l \in \mathbb{P}_f(n)} (R^l - \sum\limits_{i \ni l, i \neq f} \rho_i)}]^+ \end{aligned} \tag{C.12}$$

This finishes the proof of the theorem.

□

Appendix D

Résumé Français

La complexité et le besoin en bande passante des architectures de communication avionique ne cessent de croitre avec le nombre des calculateurs et l'expansion des données échangées. La technologie AFDX a été introduite pour offrir des communications haut débit (100Mbps) pour les avions de nouvelle génération. Cependant, ce réseau commuté est déployé de manière entièrement redondante, ce qui conduit à des quantités importantes de câbles, augmentant le poids et les coûts d'intégration. Pour faire face à ces problèmes, on propose dans cette thèse l'intégration d'un réseau Ethernet en anneau comme une solution principale pour diminuer le poids et la complexité liés au câblage. Dans ce contexte, notre objectif est de concevoir et valider un nouveau réseau de communication avionique, AeroRing, basé sur de l'Ethernet Gigabit avec une topologie anneau. Ce choix a été fait pour plusieurs raisons:

- L'Ethernet est une technologie mature et peu coûteuse. De plus, elle offre une large bande passante.
- la topologie en anneau diminuera la complexité du câblage, par rapport aux réseaux commutés, ce qui réduit les poids et augmente l'efficacité du système, i.e., moins de consommation de carburant.
- le haut niveau de disponibilité offert par la topologie anneau en raison des différentes solutions de redondance, spécifiées dans les documents IEC62439-1 / 7. Cette topologie fournit un chemin redondant implicite en introduisant une seule connexion supplémentaire entre les deux nœuds des extrémités, par rapport aux topologies ligne ou étoile [19].

Pour atteindre cet objectif, un benchmarking des solutions Ethernet (RTE) les plus pertinentes supportant les topologies anneau vis-à-vis des besoins en avionique a été réalisé, en évaluant en particulier les principaux indicateurs de performance (IP) spécifiés dans le document IEC 61784-2 [1]. Ce benchmarking a révélé que chacune des solutions RTE existantes ne satisfait que certaines exigences, mais qu'il n'y a pas de meilleure solution en termes de toutes les exigences.

Par conséquent, nous avons spécifié une nouvelle solution RTE, AeroRing, pour garantir les niveaux requis de performances et de disponibilité, tout en conservant la compatibilité IEEE802.3 et en réduisant les efforts de configuration. Les principales caractéristiques innovantes d'AeroRing sont les suivantes: (i) mécanisme d'accès distribué permettant l'échange simultané de données, pour augmenter la bande passante offerte et l'utilisation des ressources; (ii) un mécanisme distribué de gestion des pannes évitant tout point de défaillance central, ce qui permet de fournir des niveaux de fiabilité et de disponibilité élevés; (iii) communication à base d'événement améliorant la flexibilité du système et diminuant la complexité de l'implémentation, en évitant tout besoin de synchronisation; (iv) Gestion de la QoS (Quality of Service) prenant en compte des contraintes hétérogènes sur les données, grâce à un algorithme de routage orienté QoS (qualité de service).

Pour analyser les effets d'une telle proposition sur les performances temporelles de l'avionique, nous avons modélisé cette solution en utilisant le formalisme du Calcul Réseau (Network Calculus), en se basant tout d'abord sur des approches itératives existantes pour les topologies anneaux. L'évaluation de performance préliminaire a révélé que ces méthodes conduisent à des bornes excessivement pessimistes, et par conséquent à un passage à l'échelle et une utilisation de ressources limitées.

Pour permettre le calcul des bornes maximales plus précises sur les délais de bout en bout, nous avons introduit une nouvelle approche d'analyse globale, Pay Multiplexing Only at Convergence points (PMOC), qui prend en compte les phénomènes de sérialisation de flux, en considérant l'impact des flux interférents seulement aux points de convergence. Les premiers résultats ont mis en évidence l'amélioration des bornes calculées avec notre approche, par rapport aux autres méthodes. Ceci a permis d'améliorer les performances, en termes de passage à l'échelle et d'utilisation des ressources.

Par la suite, pour analyser le niveau de fiabilité d'AeroRing, nous avons mené une étude de fiabilité où le niveau de fiabilité d'AeroRing a été quantifié analytiquement, en fonction de plusieurs paramètres. Les résultats obtenus ont montré le niveau de fiabilité élevé d'AeroRing, satisfaisant les exigences de l'avionique.

Enfin, la validation d'AeroRing via une configuration représentative d'un réseau de communication avionique d'un A380 a été menée. Les résultats obtenus ont mis en évidence la capacité d'AeroRing à garantir les exigences avioniques, en termes de déterminisme, passage à l'échelle, utilisation des ressources et fiabilité.

D.1 Exigences Avionique

Les principales exigences avionique concernent à la fois les aspects techniques et les coûts:

- **Large bande passante-** le nombre de périphériques et de fonctions intégrés est de plus en plus important, ce qui augmente la quantité de données échangées. Par conséquent, pour faire face à cette expansion croissante du réseau, une large bande passante est nécessaire. En effet, comme la loi de Moore pour la puissance des processeurs, la complexité des systèmes avionique double tous les 5 ans et pour garantir une longue durée de vie des systèmes avioniques (20 ans en moyenne), il faut une large bande

passante pour permettre un développement futur;

- **Déterminisme-** le réseau doit se comporter de manière prévisible et garantir des délais minimales et maximales pour tout type de trafic. Ainsi, le système doit pouvoir fournir des informations précises dans un temps borné. En outre, les systèmes avioniques sont des systèmes temps-réel dure où les messages critiques doivent être transmis à temps, même en présence de messages non critiques. Ensuite, une gestion de la qualité de service doit être offerte;

- **Modularité-** Cette exigence est liée à la flexibilité et à l'échangeabilité des composants logiciels et matériels. Une étape importante vers l'amélioration de la modularité du système avionique a été satisfaite avec l'adoption de l'architecture IMA [14], i.e., les composants élémentaires communs peuvent être configurés pour s'adapter à des applications avioniques différentes. Cette fonctionnalité vise à minimiser les efforts de (re) configuration pour faciliter la maintenance du système et ses progrès au cours des années. Dans le cas spécifique de l'AFDX, le paradigme event-triggered événement favorise une telle exigence;

- **Fiabilité et Disponibilité-** Le réseau doit être tolérant aux pannes et satisfait les niveaux de fiabilité et de disponibilité nécessaires pour empêcher les nœuds en pannes d'affecter le fonctionnement normal. Par exemple, des mécanismes de redondance sont mis en place pour que le réseau AFDX récupère les pertes de paquets et les nœuds en pannes;

- **Résistance Physique et Electromagnétique-** les équipements avioniques sont soumis à de contraintes physiques dures telles que les vibrations, une grande variation de degrés de température et les interférences électromagnétiques. Par conséquent, le réseau doit être très résistant physiquement et en particulier au niveau des connecteurs et des câbles.

En outre, le choix d'un réseau avionique doit être efficace pour répondre aux exigences de conception à moindre coût. Ainsi, les exigences économiques sont principalement:

- **Coût-** Aujourd'hui, le réseau de communication peut atteindre 30% du coût total d'un avion, et ce nombre continuera de croître. Ainsi, un bon choix de réseau avionique est crucial pour optimiser le coût global de l'avion. La flexibilité et la configurabilité des composants réduisent la durée du cycle de développement et facilitent les processus progressifs de conception et de maintenance. En outre, l'utilisation des technologies sur étagère (COTS) implique une réduction des coûts de développement et de déploiement.

- **Compatibilité avec l'Ethernet-** pour faciliter son adoption et son interopérabilité avec le réseau coeur actuel, i.e., l'AFDX;

Cependant, le principal défi pour les solutions Ethernet prenant en topologie anneau consiste à concilier les différentes exigences avionique, en particulier le déterminisme et la disponibilité, tout en réduisant les efforts de reconfiguration et les coûts de déploiement. Pour

atteindre cet objectif, nous avons suivi une méthodologie de conception spécifique, détaillée dans la section suivante.

D.2 Méthodologie

Dans cette section, nous détaillons notre méthodologie pour concevoir et valider une solution Ethernet en anneau. Nous avons suivi une approche ńHaut-Basż, qui part des spécifications de haut niveau de la solution avionique ciblée pour converger progressivement vers la conception et la validation.

- **Évaluation les solutions Ethernet en anneau existantes:** Avant de spécifier une solution personnalisée pour répondre aux besoins avionique, nous avons commencé par une analyse approfondie des solutions existantes. Par conséquent, nous avons mené un benchmarking qualitatif et quantitatif des solutions Ethernet temps-réel (ETR) les plus pertinentes par rapport aux principales exigences en avionique. L'analyse des principaux indicateurs de performance présentés dans la norme IEC 61784 [1] a révélé l'inexistence d'une solution parfaite répondant à toutes les exigences. Cependant, cette étape nous a permis d'identifier les mécanismes les plus efficaces et d'inférer un niveau de spécification élevé de notre solution.

- **Spécification d'un nouveau réseau RTE pour l'avionique:** L'idée principale est de combler l'écart entre les solutions RTE existantes pour satisfaire les contraintes avionique. Ceci consiste principalement à garantir des niveaux élevés de fiabilité, de disponibilité et de performance temporelle tout en conservant la compatibilité IEEE802.3 et en réduisant les coûts et les efforts de configuration.

- **Analyses de performance et de fiabilité:** Pour les systèmes embarqués en avionique, il est essentiel que le réseau de communication satisfasse les exigences de certification où les contraintes temps-réel et le niveau de fiabilité doivent être garantis. Par conséquent, nous devons étudier les performances temporelles et la fiabilité de notre solution en utilisant des méthodes adéquates couvrant le cas le plus défavorable.

  - Pour mener l'analyse de performance temporelle, nous avons sélectionné le Calcul Réseau.
  - Pour mener l'analyse de fiabilité de notre solution, nous avons utilisé les réseaux de Petri Stochastiques "Stochastic Active Networks" (SANs).

- **Validation:** Pour avoir la preuve de conception de notre proposition, nous devons valider ses performances et sa fiabilité par une étude de cas avionique réaliste. Pour atteindre cet objectif, nous avons considéré un réseau avionique représentatif d'un A380 et effectué des analyses comparatives avec le réseau AFDX actuel et les solutions ETR les plus pertinentes.

D.3 Performance de l'Ethernet dans un Environnement Hostile

Dans l'avionique, les réseaux sont exposés à des interférences élevées qui peuvent dégrader ou empêcher le fonctionnement normal des réseaux et augmenter le taux d'erreur bit.

Afin de mesurer la robustesse et les performances de la technologie Gigabit Ethernet dans tels environnements, nous avons réalisé des tests d'interférence électromagnétique (IEM) sur une simple configuration. Ce système se compose de deux PC connectés par le 1000BASE-T. L'idée est d'exposer le système à différents degrés et types d'interférence et à mesurer la résistance du 1000BASE-T en utilisant trois types de câbles: un câble non blindé catégorie 5, un câble blindé catégorie 6 et un double câble AFDX. De plus, les interférences sont générées dans deux types d'environnements: une conduite cylindrique qui canalise les interférences ou une chambre anéchoïque contenant une puissante antenne.

On a obtenu les résultats suivants:

- Dans la chambre anéchoïque, les tests ont été faits sur les câbles de catégories 6 et AFDX avec des puissances de 3, 10 et 20 V/m. Les interférences générées n'ont pas altérés le bon fonctionnement du réseau et aucune perte n'a été détectée.

- Dans la conduite cylindrique, les tests ont été effectués sur les câbles AFDX et UTP cat 5 avec une puissance de 20 V / m. Aucune perte n'a été détectée sur le câble AFDX. Cependant, dans le cas de UTP cat 5, les premières pertes ont été détectées à une fréquence de 70 Mhz. Les pertes ont augmenté lors de l'augmentation de la fréquence, jusqu'à une perte complète de connexion avec des fréquences d'environ 100 Mhz. Par la suite, les pertes ont commencé à diminuer jusqu'à disparition avec des fréquences supérieures à 120 Mhz.

Fig. D.1 montre le taux d'erreur en fonction du numéro de d'une fenêtre glissante, où chaque fenêtre contient 100000 paquets.

Si la PHY ne peut pas corriger les erreurs et détecte que la qualité de la communication est dégradée, le 1000BASE-T passe d'un mode 1000 Mbps à un mode inférieur (100 Mbps ou 10 Mbps) pour mieux contrôler la communication et génère plus de redondances pour corriger les données. Le taux d'erreur élevé, i.e., égale ou proche de 1 dans la figure ref figure: expérimentation2, correspond à une perte de connexion, où les deux PHY effectuent le mécanisme d'auto-négociation pour négocier un mode de transmission capable de résister aux interférences , i.e., réduit le débit des données et augmente la redondance. Par conséquent, la disparition des pertes peut avoir deux explications: 1) les deux PHY ont pu négocier un mode de fonctionnement capable de résister aux interférences; 2) les interférences générées étaient décorrélées du signal transmis, i.e., n'ont pas affecté le signal.

D.4 Les Solution ETR à Base d'Anneau vs les Exigence Avioniques

Dans cette section nous allons identifier les principaux paramètres qui impactent les performances et fiabilité des solutions ETR. Ensuite, nous introduisons une nouvelle classification des solutions existantes du point de vue avionique.

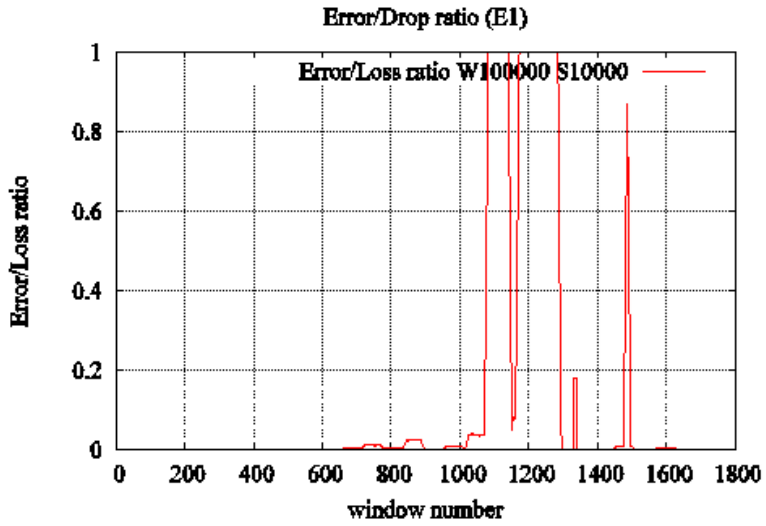


Figure D.1: Taux d'erreur moyen en fonction du nombre de la fenêtre

D.4.1 Taxonomie

Nous avons identifié deux caractéristiques principales: le paradigme de communication et les protocols de redondance.

D.4.1.1 Paradigme de communication

Ce paramètre est d'une importance capitale pour quantifier l'effort de reconfiguration nécessaire à la solution. Il indique le niveau de modularité offert par la solution sélectionnée, une exigence clé dans l'avionique. Les deux principaux paradigmes sont l'event-triggered et le time-triggered [**?**]. Le paradigme event-triggered est très flexible et facilite la reconfiguration du système, mais introduit en même temps de l'indéterminisme. D'autre part, le paradigme time-triggered est hautement prévisible, mais présente des limitations en termes de reconfiguration du système.

D.4.1.2 Protocols de redondance

Ce paramètre affecte particulièrement le niveau de disponibilité du réseau, mais aussi les coûts de déploiement. Nous identifions principalement deux classes de solutions de redondance, statiques et dynamiques. Le premier est généralement basé sur un réseau totalement dupliqué, où les deux sont utilisés en parallèle pour augmenter la disponibilité avec un temps de redondance nul, mais aussi augmente les coûts de déploiement. De l'autre coté, les solutions de redondance dynamique ont été introduites pour diminuer les coûts de déploiement, en utilisant un chemin redondé en cas de pannes, mais ils doivent offrir un petit temps de reconfiguration pour garantir la disponibilité.

Plusieurs protocols de redondance pour les solutions ETR basées sur la topologie anneau ont été proposés et cités dans IEC62439-1/7. Les protocoles statiques les plus pertinents sont le Parallel Redundancy Protocol (PRP) [35] et leHigh-availability Seamless Redundancy protocol (HSR) [35]; alors que les principaux protocoles dynamiques sont leDistributed Redundancy

Protocol (DRP) [36], le Media Redundancy Protocol (MRP) [37] et le Ring-based Redundancy Protocol (RRP) [38].

D.4.2 Classification

Une classification différente est présentée ici pour distinguer les principales solutions de RTE du point de vue de l'avionique, comme le montre la Fig. Ref fig: classes. Quatre classes de solutions RTE supportant la topologie des anneaux ont été identifiées:

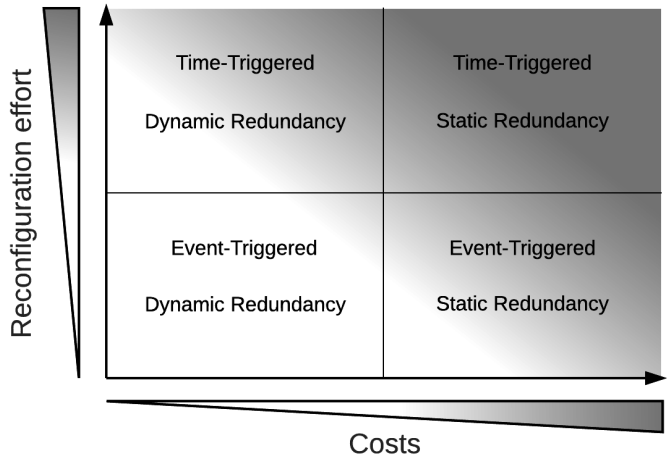


Figure D.2: Classification des solutions ETR basées sur le paradigme de communication et les mécanismes de redondance

- Event-triggered avec redondance statique: cette classe représente le réseau avionique actuel basé sur le standard AFDX. Cette solution réduit l'effort de reconfiguration tout en augmentant les coûts de déploiement. Par conséquent, il est considéré comme une référence pour l'analyse comparative des solutions RTE les plus pertinentes;
- Time-triggered avec redondance statique: une solution représentative dans cette classe est le Time Triggered Ethernet (TTE) [17]. Cette solution offre des niveaux élevés de déterminisme et de disponibilité, mais augmente en même temps les coûts de déploiement et l'effort de reconfiguration. Par conséquent, cette solution sera écartée;
- Time-triggered avec redondance dynamique: deux solutions intéressantes peuvent être identifiées dans cette classe, EtherCAT [52, 53] et Profinet/IRT [54]. Cette classe de solutions réduit les coûts de déploiement, mais augmente en même temps l'effort de reconfiguration;
- Event-triggered avec redondance dynamique: on peut identifier dans cette classe l'Ethernet/IP avec DLR [48]. Cette classe permet de réduire l'effort de reconfiguration, tout en mettant en œuvre une solution de redondance dynamique pour réduire les coûts de déploiement. Du point de vue pratique, cette classe devrait contenir la meilleure solution pour l'avionique en termes de modularité et de coûts.

Le tableau D.1 représente un récapitulatif de la comparaison qualitative et quantitative des principales solution ETR en se basant sur les principaux indicateurs de performances définis dans IEC 61784-2 [1].

Protocols	Reliability	Availability	Predictability	Modularity	Costs
EtherCAT	Medium	High	High	Low	High
PROFINET/IRT	Medium	High	High	Low	High
Ethernet/IP with DLR	High	Low	Low	High	Medium

Table D.1: Benchmarking des solutions ETR en topologies anneau

D.5 Spécification d'AeroRing

L'objectif principal d'AeroRing est de permettre une architecture de communication homogène pour l'avionique tout en réduisant l'effort de configuration et les coûts de déploiement. Par conséquent, AeroRing satisfait les principales exigences avionique comme suit:

- Garantir un processus de déploiement simple et une intégration a moindre coût en raison de sa *Compatibilité avec IEEE 802.3* et les textit différentes topologies proposées basées sur les anneaux, i.e., des topologies mono-anneau et multi-anneau simples ou dupliquées;
- Fournir un haut niveau de **modularité** et réduire l'effort de (re)configuration, en mettant en œuvre un *paradigme de communication event-triggered*;
- Favoriser la **predictability** à l'aide du mécanisme de *routage à base de qualité de service* et le *trafic policeur*, pour gérer les contraintes de données hétérogènes;
- Offrant un haut niveaux de **disponibilité** et de **fiabilité** grâce à un *Protocole de redondance dynamique*.

D.5.1 Caractéristiques Principales

AeroRing est un réseau basé sur la technologie Ethernet avec une architecture Daisy-Chain bidirectionnelle, permettant de connecter des équipements ń Ethernet-Compliant ż via des ń End-Systems ż, appelés T AeroRing. La figure D.3 illustre un exemple de ce type de topologie.

En plus de cette architecture de base, AeroRing supporte une deuxième architecture, appelé multiple-ring comme le montre le figure D.4. L'idée clé est de rassembler les nœuds dans des anneaux périphériques en fonction de leurs données échangées. Ce fait diminuera les délais de bout en bout, qui dépendent de la longueur du chemin de données

Un T-AeroRing est un commutateur Ethernet duplex complet à 3 ports avec les caractéristiques principales suivantes:

- ***Transmission en Cut-Through:*** un T-AeroRing commence à transférer le paquet juste après son identification, i.e., quand l'en-tête du paquet est décodé. Cette technique permet de réduire les délais de transmission;

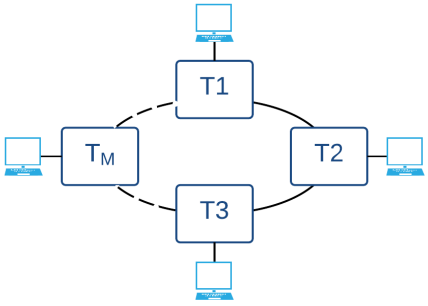


Figure D.3: L'architecture réseau *mono-ring*

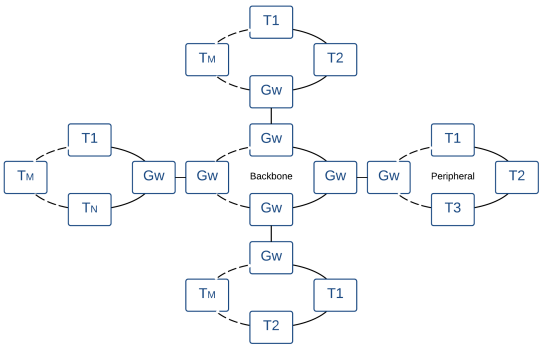


Figure D.4: L'architecture réseau *Multiple-ring*

- ***Politique de service à priorités statique:*** les paquets sont mis dans les files d'attente de chaque port de sortie des T-AeroRings en fonction de leurs priorités. Les files d'attente sont sélectionnées pour la transmission selon leurs niveaux de priorité. Ensuite, pour chaque file d'attente, l'ordre de transmission est First In First Out (FIFO). Les priorités sont définies selon la norme IEEE 802.1p où le tague 802.1Q (champ 3 bits) est utilisé pour définir les quatre classes de priorité;

- ***Traffic policer:*** Pour garantir les performances temps-réel, un T-AeroRing implémente des mécanismes de contrôle de trafic, basés sur la méthode Leaky Bucket. Chaque trafic dépassant son contrat associé est écarté pour garantir le déterminisme de la communication;

- ***Routage:*** chaque T-AeroRing construit sa table de routage sur la base de messages de contrôle échangés entre les T-AeroRings interconnectés, en utilisant un mécanisme d'auto-configuration distribué. Chaque T-AeroRing implémente deux modes de routage pour transmettre ses paquets générés en fonction de leurs priorités: (i) l'envoi sur les deux ports de l'anneau pour les classes de trafic prioritaires, pour permettre un niveau de fiabilité élevé; (ii) envoyer sur le port correspondant au chemin le plus court pour les classes de trafic de moyenne et basse priorité, pour offrir un niveau de performance élevé, i.e., un court délai;

- ***Protocol de Redondance d'AeroRing (ARRP):*** ARRP intègre des mécanismes dynamiques pour la détection des pannes et la reconfiguration des tables de routage, en fonction des messages de contrôle. De plus, ARRP permet l'utilisation complète de la topologie anneau, i.e., l'anneau n'est pas transformée en une ligne en bloquant certains ports, grâce à ses mécanismes de filtrage afin d'éviter le bouclage infini des messages.

D.5.2 Mécanismes Temps-Réel et Gestion de QoS

Dans cette section, nous décrivons d'abord les types de flux de données pris en charge par AeroRing. Ensuite, nous détaillons l'algorithme de routage.

D.5.2.1 Types des Flux de Données

AeroRing garantit la gestion de la QoS grâce à l'implémentation de la politique de service "Priorité statique", qui prend en charge les types de flux de données suivants:

- ***Données HRT***: ce trafic a le niveau de priorité le plus élevé (N1) et est généralement généré par les applications temps-réel avec des contraintes temporelles dure. Ce type de flux de données est envoyé sur les deux ports de l'anneau pour assurer un niveau de fiabilité élevé et est identifié par un numéro de séquence de 2 octets, essentiel pour filtrer les messages redondants dans la T-AeroRing destinataire;

- ***Données SRT***: Ce trafic est principalement envoyé par des applications temps-réel souple, telles que les transferts audio ou vidéo, a le niveau de priorité moyenne (N2). Ce type de flux de données est envoyé sur le port de l'anneau correspondant au chemin le plus court pour garantir un niveau de performance élevé, i.e., un délai de transmission court;

- ***Données NRT***: ce trafic correspond à des applications non temps-réel, telles que le transfert de fichiers, et le niveau de priorité le plus bas (N3). Ce type de flux de données est envoyé sur le port de l'anneau correspondant au chemin le plus court pour garantir un niveau de performance élevé.

Les priorités sont traitées selon la spécification IEEE 802.1Q. De plus, le champ VID est utilisé pour identifier l'anneau périphérique à laquelle le message est destiné dans le cas d'une architecture multiple-ring.

D.5.2.2 Routage

Sur la base de la description des ports T-AeroRing dans la figure D.5, chaque message sera transmis comme suit:

- Les messages reçus du port 3, i.e., de l'équipement connecté, sont transmis au port 1 ou/et 2 selon leur niveau de priorité, comme suit:

  – Les messages de priorité N1 sont transmis via les deux ports.

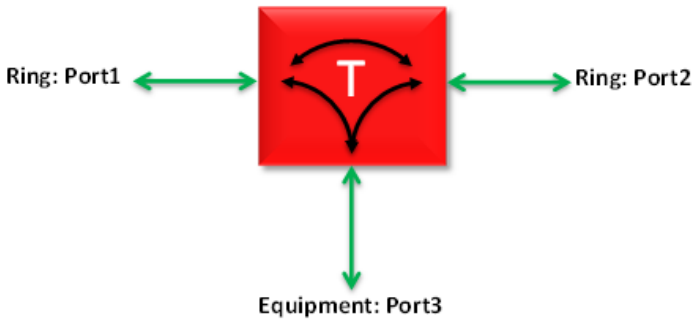


Figure D.5: Les différents ports d'un *T-AeroRing*

  - Pour les messages de priorités N2 et N3, nous distinguons deux cas: i) si le destinataire finale appartient au même anneau périphérique que la source, les messages sont envoyés via le port correspondant au chemin le plus court; ii) sinon, les messages sont envoyés via le port correspondant au chemin le plus court vers la passerelle.
  - Les messages de broadcast avec les priorités N2 et N3 sont transmis par un port prédéfini ou par un port sélectionné au hasard.

- Les messages reçus du port 1 ou 2 sont traités selon leur niveau de priorité et leur adresse de destination. Si l'adresse de destination correspond à l'équipement connecté au T-AeroRing, le message est envoyé au port 3; sinon, les messages sont transmis au port opposé. Il convient de noter que chaque message avec priorité N1 est envoyé au port 3 uniquement si son réplica n'a pas encore été reçu.

D'autre part, les messages sont traités dans la passerelle comme suit:

- Pour les messages reçus d'un port anneau, i.e., le port 1 ou 2, on distingue trois cas:
  - Les messages tagués 802.1Q (ou non tagués) sont transmis selon le VID (resp. MAC). Si le VID correspond au VID de l'anneau périphérique (resp. Le MAC se trouve dans la table de routage de la passerelle), alors les messages sont transmis dans l'anneau périphérique; sinon, ils sont transmis dans l'anneau cœur (backbone).
  - Les messages de broadcast global sont transmis dans l'anneau périphérique et backbone.

- Les messages reçus de l'anneau cœur (resp. Périphérique) sont transmis selon l'adresse MAC (resp. VID). Si l'adresse MAC (resp. VID) se trouve dans la table de routage de la passerelle ou est une adresse de broadcast, les messages sont transmis dans la l'anneau périphérique (resp. Backbone), selon leurs niveaux de priorité de manière similaire à un T-AeroRing; sinon, les messages sont rejetés conformément aux règles de filtrage qui seront détaillées plus tard. Pour les messages non 802.1Q reçus du port 3, nous avons le même comportement de passerelles, sauf lorsque l'adresse MAC n'est pas dans la table

de routage de la passerelle. Dans ce cas particulier, ils sont transmis dans l'anneau cœur par un port anneau sélectionné de manière aléatoire ou par défaut.

D.5.2.3 Mécanismes Temps-Réel

Le comportement temps-réel d'AeroRing et la garantie de délais des données livrées sont favorisés en raison des fonctionnalités implémentées dans les T-AeroRings. Tout d'abord, la technique de transmission "Cut Through" permet de réduire le temps de transmission le long du réseau, ce qui améliore le délai de livraison maximum de bout en bout. Ensuite, le trafic policeur empêche la saturation du réseau par un équipement déficient, ce qui garantit le déterminisme des communications. En outre, l'algorithme de routage implémenté permet de supporter la transmission du flux de données SRT et NRT sur le chemin le plus court, ce qui diminue les délais de transmission et les flux de données HRT sur les deux chemins pour augmenter le niveau de fiabilité. Enfin, la politique de service Priorité Statique assure l'isolement temporel entre les données de criticité mixtes avec diverses contraintes temporelles et garantit un délai borné pour la classe de trafic HRT.

D.5.3 Sureté de Fonctionnement et Tolérance aux Pannes

AeroRing implémente des mécanisme de détection de pannes et de reconfiguration distribués, ce qui permet d'améliorer la fiabilité et la disponibilité du réseau. Ces mécanismes sont basés sur un échange de messages de contrôle, qui ont le niveau de priorité le plus élevé N0. Les messages de contrôle sont identifiés par la valeur de type "0x9000". La figure D.6 montre la structure d'un message de contrôle, où le champ CTL identifie le type de message de contrôle. En outre, AeroRing implémente des mécanismes de filtrage, qui permettent de détecter les données N1 redondantes aux noeuds de destination et de supprimer des messages non valides du réseau, afin d'éviter un bouclage infini des messages.

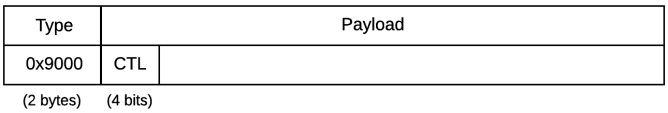


Figure D.6: Structure of a control message

D.5.3.1 Détection de Pannes

Afin de réduire les messages de contrôle, un T-AeroRing déduit que son voisin est opérationnel s'il lui envoie des données. Par conséquent, tout T-AeroRing doit considérer une connexion comme interrompue avec un voisin, s'il ne reçoit aucun message de son voisin pendant une certaine période appelée "période de détection". En pratique, si un T-AeroRing n'a pas de données à transmettre à son voisin, il annonce périodiquement son statut à ce voisin via l'envoi de messages de contrôle. Ces messages ont la structure représentée dans la Fig. 3.7 avec le champ CTL "0000".

Ces messages de contrôle pour annoncer l'état aux voisins sont envoyés périodiquement lorsque:

- Le *T-AeroRing* n'a pas de données à envoyer sur l'un de ses ports anneau pendant une période appelée *période d'annonce*;

- Le *T-AeroRing* n'a reçu aucun message de données ou de contrôle de l'un de ses voisins pendant une durée égale à *période de détection*. Dans ce cas, le T-AeroRing indique à son voisin par un message de contrôle que la connexion est considérée comme interrompue.

Une connexion interrompue est considérée opérationnelle à nouveau, si le T-AeroRing commence à recevoir des messages de son voisin. Dans ce cas, le mécanisme d'auto-configuration mettra à jour les tables de routage.

D.5.4 Mécanisme d'Auto-Configuration

Pour réduire l'effort de configuration et faciliter l'adoption de la nouvelle solution ETR, AeroRing offre un service de configuration automatique jusqu'à ce que tout le réseau soit opérationnel. Ce service est basé sur une simple affectation d'adresses et un processus de découverte de topologie de réseau dynamique.

Les messages sont routés dans les anneaux périphériques en fonction des adresses MAC et de l'anneau cœure en fonction du VID. Par conséquent, les tables de routage des T-AeroRings et des passerelles périphériques se composent des adresses MAC et les tables de routage des passerelles de l'anneau cœure se composent des VID. Ces tables de routage permettent de sélectionner le port correspondant au chemin le plus court (ports 1 ou 2) pour une destination. Ils sont construits sur la base de messages de contrôle échangés entre les nœuds. La structure de ces messages de contrôle est illustrée dans la Fig. D.7.

Type	Payload					
0x9000	0001	gw	NBAD	ADD1	..... ADDN-1	ADDN
(2 bytes)	(4 bits)	(6 bytes)	(12 bits)	(6 bytes)		(6 bytes)

(a)

Type	Payload					
0x9000	0002	NBAD	ADD1	ADD2	..... ADDN-1	ADDN
(2 bytes)	(4 bits)	(12 bits)	(12 bits)	(12 bits)		(12 bits)

(b)

Figure D.7: (a) Structure d'un message de contrôle de l'anneau périphérique; (b) Structure d'un message de contrôle de l'anneau cœure.

Les messages de contrôle utilisés pour construire les tables de routage dans un anneau périphérique (resp. cœure) sont identifiés par le champ CTL "0001" (resp. "0002"):

- Gw est utilisé par la passerelle pour spécifier son adresse MAC;

- NBAD compte le nombre d'adresses insérées dans ADDx;

- ADDx est utilisé par les T-AeroRings (resp. passerelles) pour insérer leurs adresses MACs (resp. VIDs);

Les messages de contrôle utilisés pour construire les tables de routage sont gérés comme suit:

- À chaque changement de topologie, le T-AeroRings détectant cet événement envoie périodiquement des messages de contrôle sur les deux ports anneau avec la plus haute priorité, pour mettre à jour les tables de routage des autres T-AeroRings interconnectés. Le champ NBAD est mis à zéro et les champs ADDx sont vides;
- Chaque T-AeroRing contribue à la construction des tables de routage lors de la réception des messages de contrôle par:
    1. Incrementer le compteur NBAD et insérer son adresse à la fin de la liste ADDx pour respecter l'ordre physique;
    2. Calculer le nouveau FCS;
    3. Relayer le message au T-AeroRing suivant;
    4. Mettre à jour sa table de routage;
    5. De plus, la passerelle insère, en plus de son adresse dans ADDx, son adresse dans le champ gw pour permettre son identification par les autres T-AeroRings.
- Le T-AeroRing détectant le changement de topologie arrêtera la transmission périodique lors de la réception d'un message de contrôle d'un autre T-AeroRing sur le même port. Cela signifie qu'il a un voisin de ce côté et ce n'est plus le dernier nœud du segment;
- La transmission des messages de contrôle s'arrête complètement dans le réseau lorsque l'anneau est fermé.

De plus, chaque passerelle périphérique transmet sa table de routage à la passerelle cœur pour permettre à celle-ci d'acheminer les messages inter-anneaux.

Le mécanisme d'auto-configuration est exécuté effectué par les passerelles dans l'anneau cœur de manière similaire aux T-AeroRings dans une anneau périphérique, en utilisant le message de contrôle avec le type 0002 et les VID au lieu des adresses MAC.

D.5.5 Filtrage

Pour profiter pleinement des chemins redondants de l'anneau, AeroRing permet des communications sur les deux sens de l'anneau en mettant en œuvre certaines règles de filtrage afin d'éviter le bouclage infini des messages.

Semblable à la solution Ethernet standard, les T-AeroRings suppriment les messages erronés à la réception grâce au champ FCS. Cependant, si l'erreur n'est pas détectée en fonction du champ FCS, et elle se produit sur l'en-tête, la trame doit être éliminée du réseau pour éviter une boucle infinie de messages. Les messages sont filtrés de l'anneau dans un T-AeroRing lors de leur réception, s'ils satisfont à l'une des conditions suivantes:

- L'adresse MAC source est celle du T-AeroRing;
- L'adresse MAC destination correspond à celle du T-AeroRing;

- Aucune des deux adresses ne font partis de l'anneau, i.e., La condition peut être vérifiée en utilisant les tables de routage.

Du côté des passerelles, les messages sont filtrés si le VID n'est pas dans les tables de routage des passerelles.

En plus de ces mécanismes, chaque T-AeroRing gère les messages N1 dupliqués pour délivrer uniquement le premier réplica valide reçu. Lorsqu'une destination reçoit un message avec la priorité N1, elle stocke le couple <src MAC, numéro de séquence> dans une table, pour permettre d'identifier et de rejeter ses répliques. Une fois que ce dernier est reçu, ou après un délai d'attente, le couple <src MAC, numéro de séquence> mémorisé est supprimé de la table.

D.6 Évaluation de Performance d'AeroRing

D.6.1 Méthodes d'Analyse Conventionnelle et leurs Limites

Pour le cas particulier du réseau anneau, seules quelques techniques ont été proposées. Les deux approches intéressantes qui ont été proposées sont: le ***Time Stopping Method*** [22] et ***Backlog-based Method*** [20].

Le ***Time Stopping*** a été proposée dans [22] et se compose de deux étapes. Tout d'abord, on supposera une borne finie sur la burstiness des flux transmis pour obtenir un ensemble d'équations pour calculer les borne de délai. Ensuite, les conditions de faisabilité pour résoudre ces équations sont définies.

La ***Backlog-based*** méthode a d'abord été proposée dans [25] et plus récemment généralisée dans [20]. Elle donne un backlog maximal lorsque l'on considère des noeuds non conservateurs de travail, ce qui correspond à la quantité totale de données présente dans le réseau à tout moment.

Afin de montrer les limitations de chacune des solutions. Nous considérons le cas particulier de communications broadcasts. La Time Stopping méthode permet de calculer les bornes lorsque le taux d'utilisation maximal du réseau est inférieur à $\frac{2}{(M-1)}$ (M est la taille du réseau). Comme on peut le voir dans la figure D.8, le taux d'utilisation maximum pour la méthode Time Stopping a tend vers 0, lorsque $M \to \infty$. Cela implique que le réseau doit être sous-utilisé pour satisfaire la condition de stabilité du réseau, ce qui limite l'efficacité et l'utilisation des ressources.

D'autre part, avec l'approche Backlog-based, le backlog et le délai de bout en bout deviennent des fonctions polynomiales de la variable M (nombre de noeuds) de degré 3 et 4, respectivement. Ce fait implique un délai de bout en bout croissant en $\theta(M^4)$, comme le montre la figure D.8.

La méthode Time Stopping limite effectivement les performances du réseau en termes d'utilisation des ressources, i.e., le taux d'utilisation diminue dramatiquement lorsque la taille du réseau augmente; alors que la méthode Backlog-based limite le passage à l'échelle du système, i.e., le nombre de nœuds est très limité pour garantir les délais temporels.

Pour surmonter ces limitations, nous présentons dans la section suivante une nouvelle méthode d'analyse pire cas des réseaux en anneau avec des dépendances cycliques, en comptant les phénomènes de sérialisation des flux le long de leurs chemins.

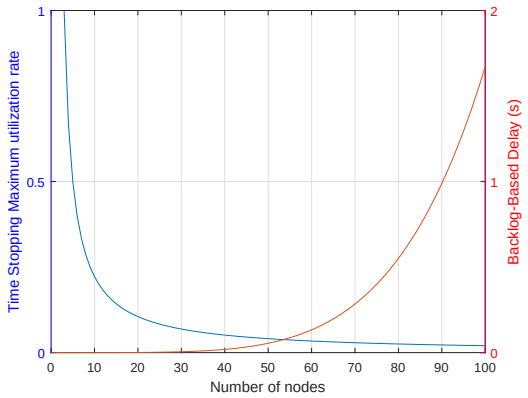


Figure D.8: Le taux d'utilisation maximal pour la Méthode Time Stopping et la born de délai maximal pour la méthode Backlog-based vs le nombre de noeuds.

D.6.2 Pay Multiplexing Only at Convergence Points

Cette approche consiste à considérer les phénomènes de sérialisation des flux sur le chemin d'un flux d'intérêt *f.d.i*, en payant les bursts des flux interférents uniquement aux points de convergence[1]. Des concepts similaires ont été développés dans la littérature pour les réseaux non cycliques, i.e., Sans dépendances cycliques, telles que Pay Bursts Only Once (PBOO) dans [20] et Pay Multiplexing Only Once (PMOO) dans [67] [68]. Cependant, affiner les bornes de délai des réseaux cycliques est toujours un problème ouvert dans la littérature, et une telle approche n'existe pas encore pour les réseaux cycliques. L'idée principale de cette méthode est de gérer un tel problème pour les réseaux en anneaux et généraux.

D.6.3 Courbe de Service pour un Flux d'Intérêt

La première étape de l'approche PMOC consiste à définir la courbe de service garantie pour un *f.d.i* le long de l'un de ses sous-chemins dans un réseau en anneau. Le théorème 5 montre cette courbe sous un multiplexage arbitraire.

[1] Dans les réseaux anneaux, deux flux peuvent se joindre à un nœud, appelé le point de convergence, puis disjoint après avoir un sous-chemin commun pour se rejoindre à nouveau dans un autre point de convergence.

Theorem 5. *(Courbe de Service dans un Réseau en Anneau avec un Multiplexage Arbitraire) La courbe de service offerte à un flux f le long de son sous-chemin, $\mathbb{P}_f(n)$, avec des courbes de service stricts du type rate-latence $\beta_{R,T}$ et des courbes d'arrivée en leaky bucket $\alpha_{\sigma,\rho}$, est une courbe rate-latence, avec une capacité $R^{\mathbb{P}_f(n)}$ et une latence $T^{\mathbb{P}_f(n)}$, défini comme suit:*

$$R^{\mathbb{P}_f(n)} = \min_{k\in\mathbb{P}_f(n)} [R^k - \sum_{i\ni k, i\neq f} \rho_i] \tag{D.1a}$$

$$T^{\mathbb{P}_f(n)} = \sum_{k\in\mathbb{P}_f(n)} T^k + \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^0.1_{\{f\ni i.ft\}} + \rho_i \cdot \sum_{k\in\mathbb{P}_f(n)\cap\mathbb{P}_i} T^k}{R^{\mathbb{P}_f(n)}}$$
$$+ \sum_{i\in\mathbb{K}_f(n)} \frac{\sigma_i^{f.ft\ominus 1}.1_{\{i\ni f.ft/i.ft\neq f.ft\}}}{R^{\mathbb{P}_f(n)}} \tag{D.1b}$$

Où $1_{\{cdt\}}$ est égual à 1 si cdt est vrai, sinon zéro.

Le corollaire 7 montre la courbe de service sous un multiplexage Priorité Statique.

Corollary 7. *(Courbe de Service dans un Réseau en Anneau avec un Multiplexage Priorité Statique)*

La courbe de service offrait un flux d'intérêt f le long de son sous-chemain , $\mathbb{P}_f(n)$ avec des courbes de service stricts du type rate-latence $\beta_{R,T}$ et des courbes d'arrivée en leaky bucket $\alpha_{\sigma,\rho}$, est une courbe rate-latence, avec une capacité $R^{\mathbb{P}_f(n)}$ et une latence $T^{\mathbb{P}_f(n)}$, défini comme suit:

$$\begin{aligned} R^{\mathbb{P}_f(n)} &= \min_{k\in\mathbb{P}_f(n)} [R^k - \sum_{i\ni hp_f^k} \rho_i] \\ T^{\mathbb{P}_f(n)} &= \sum_{k\in\mathbb{P}_f(n)} (T^k + \frac{\max_{i\in lp_f^k} L_{max}(i)}{R^k}) \\ &+ \sum_{i\in\mathbb{K}_{\leq f}(n)} \frac{\sigma_i^0.1_{\{f\ni i.ft\}} + \rho_i \cdot \sum_{k\in\mathbb{P}_f(n)\cap\mathbb{P}_i} (T^k + \frac{\max_{j\in lp_f^k} L_{max}(j)}{R^k})}{R^{\mathbb{P}f(n)}} \\ &+ \sum_{i\in\mathbb{K}_{\leq f}(n)} \frac{\sigma_i^{f.ft\ominus 1}.1_{\{i\ni f.ft/i.ft\neq f.ft\}}}{R^{\mathbb{P}f(n)}} \end{aligned} \tag{D.2}$$

D.6.4 Calcul de la Born Maximale du Délai

La deuxième étape de l'approche PMOC consiste à calculer les bornes de délai. Pour cela, on met toutes les contraintes du système d'un réseau en anneau, qui dépend de certaines variables, i.e., les bursts propagées et des services offerts:

Courbe de ServiceConstraint

$$\overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T} = \overbrace{\begin{bmatrix} c1_{f1} \\ \vdots \\ c1_{fh_f} \\ \vdots \end{bmatrix}}^{C1} + \overbrace{\begin{bmatrix} a1_{f,1} & \cdots & a1_{f,h_f} & \cdots \\ \vdots & \ddots & \ddots & \\ a1_{fh_f,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}}^{A1} \times \overbrace{\begin{bmatrix} \sigma_f^{f.ft\oplus 1} \\ \vdots \\ \sigma_f^{f.ft\oplus(h_f-1)} \\ \vdots \end{bmatrix}}^{\sigma}$$

Où T est le vecteur des latences des services offerts(D.1b)), $A1$ est la matrice des coefficients des bursts inconnus propagés et $C1$ est le vecteur des constantes, i.e, les latences et les bursts initiaux.

Courbe d'Arrivée de Sortie

$$\overbrace{\begin{bmatrix} \sigma_f^{f.ft\oplus 1} \\ \vdots \\ \sigma_f^{f.ft\oplus(h_f-1)} \\ \vdots \end{bmatrix}}^{\sigma} = \overbrace{\begin{bmatrix} c2_{f1} \\ \vdots \\ c2_{fh_f} \\ \vdots \end{bmatrix}}^{C2} + \overbrace{\begin{bmatrix} a2_{f,1} & \cdots & a2_{f,h_f} & \cdots \\ \vdots & \ddots & \ddots & \\ a2_{fh_f,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}}^{A2} \times \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

Où σ est le vecteur des bursts inconnus propagés, $A2$ est la matrice des coefficients des latences inconnus offertes, et $C2$ est le vecteur des constantes, i.e., les bursts initiaux σ_f^0.

Delay bound

$$\overbrace{\begin{bmatrix} EED^{\mathbb{P}_f(2)} \\ \vdots \\ EED^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{EED} = \overbrace{\begin{bmatrix} c3_{f1} \\ \vdots \\ c3_{fh_f} \\ \vdots \end{bmatrix}}^{C3} + \overbrace{\begin{bmatrix} T^{\mathbb{P}_f(2)} \\ \vdots \\ T^{\mathbb{P}_f(h_f)} \\ \vdots \end{bmatrix}}^{T}$$

Où $C3$ est le vecteur des constantes.

Ce système matriciel est transformé en suivant:

$$\begin{cases} (Id - A1 \times A2) \times T = C1 + A1 \times C2 \\ EED = C3 + T \end{cases} \tag{D.3}$$

À partir du system matriciel dans (D.3), on déduit une condition nécessaire et suffisante sur l'existence de borne maximale de délai pour chaque *f.d.i* en fonction des débits des flux: la matrice $(Id - A1 \times A2)$ in $\mathbb{M}^*$ doit être inversible.

D.6.5 Comparaison avec l'État de l'Art

Dans cette section, nous comparons les bornes de délai obtenues avec l'approche PMOC par rapport aux approches existantes, i.e., Time Stopping et Backlog-based.

Fig. D.9 montre une comparaison des différentes approches en fonction de la taille du réseau. Comme on peut le constater, l'approche PMOC offre des bornes de délai plus serrées pour des réseaux à grande échelle, tout en garantissant le délai de flux par rapport aux méthodes conventionnelles, par exemple, le délai PMOC est de 0.3 ms comparé à 33.8 ms et 1.6 s pour Time-Stopping et Backlog-Based méthodes pour un réseau de 100 nœuds. Ainsi, la taille maximale du réseau respectant le délai du flux est d'environ 20 et 27 nœuds avec les méthodes Backlog-Based et Time-stop, respectivement, alors qu'elle atteint 100 nœuds avec l'approche PMOC.

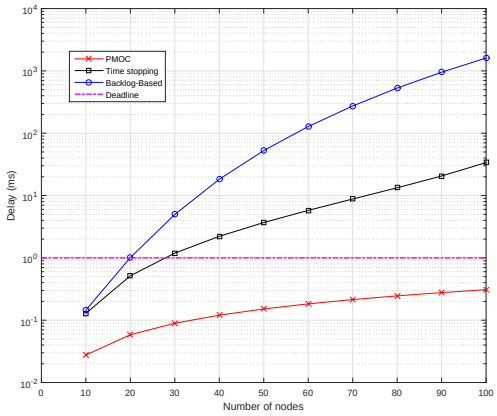


Figure D.9: Borne de maximale délai vs nombre de nœud.

Fig. D.10 illustre l'impact de l'augmentation de la congestion sur les différentes méthodes. Comme on peut le voir, la méthode Time Stopping diverge pour un taux d'utilisation global autour de 22.22%; alors qu'il atteint 55.55% avec notre approche, qui correspond à $\frac{M}{2(M-1)}$. Cependant, un taux d'utilisation maximal peut être atteint avec la méthode Backlog-Based, même si les délais sont trop pessimistes, par exemple, 1.22 s pour $U_{max} = 99\%$. De plus, le taux d'utilisation maximum du réseau respectant l'échéance des flux n'est que d'environ 7.1% et 19.36% avec les méthodes Backlog-Based et Time Stopping, contre 54.6% avec PMOC.

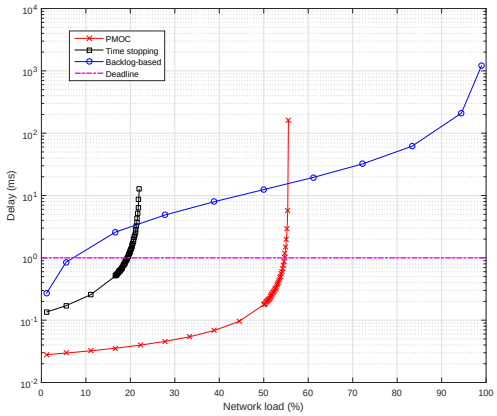


Figure D.10: Borne de maximale délai vs taux d'utilisation du réseau

Cette analyse comparative montre que l'utilisation de l'approche PMOC améliore les performances du réseau, en termes d'utilisation des ressources et de passage à l'échelle du réseau, par rapport aux approches classiques.

D.7 Analyse de fiabilité d'AeroRing

Dans cette section, nous allons quantifier analytiquement le niveau de fiabilité garanti d'AeroRing en fonction de plusieurs aspects, tels que la taille du réseau, la fiabilité des équipements et la durée de mission. Nous avons sélectionné les réseaux de Petri, et plus particulièrement Stochastic Activity Networks (SANs) [80, 81, 82, 83, 75] afin de modéliser notre système, qui sera construit et résolu analytiquement en utilisant l'outil Moëbius [86].

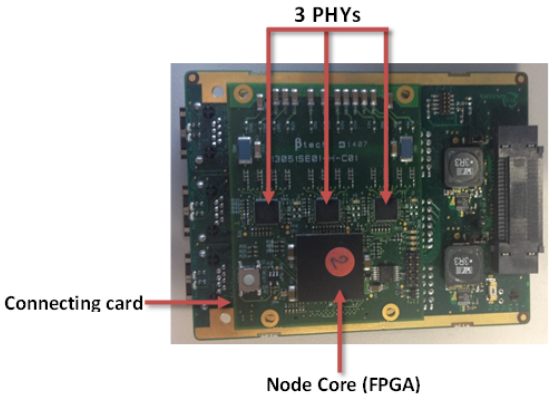


Figure D.11: A real T-AeroRing prototype with the different components

La figure D.11 montre les différentes entities qui composent notre T-AeroRing prototype. De plus de ces entities, on l'entitie alimentation qui n'est pas représentée sur cette photo.

D.7.1 Stratégie de Modélisation

Afin de modéliser notre système, nous avons divisé notre modèle de système en sous-modèles SAN classés en 4 catégories:

- ***Cat1.*** modélise l'occurrence de panne des différentes entités;
- ***Cat2.*** modélise l'impact de la panne sur le réseau;
- ***Cat3.*** selon le mode de défaillance, le sous-modèle Cat3 détermine si la panne est correctement traitée ou non;
- ***Cat4.*** ce sous-modèle est utilisé pour évaluer la défaillance du système et il prend en compte l'état du réseau, e.g., les pannes survenus, pour décider si le système est défaillant ou non.

D.7.2 Résultats Numériques

Dans cette section, nous détaillons quelques résultats numériques de l'analyse de sensibilité du niveau de fiabilité d'AeroRing.

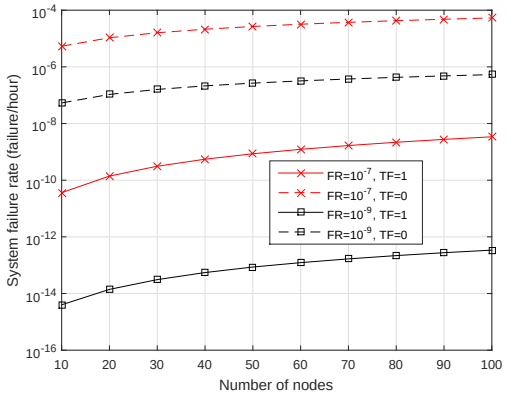


Figure D.12: Taux de panne du système mono-ring vs la taille du réseau

La figure D.12 montre le taux de panne du système mono-ring en fonction de la taille du réseau avec un taux de panne équipement $FR = 10^{-7}$ et 10^{-9} et zéro ou une tolérance de panne ($FT = 0$ ou 1).

Il est clair que le taux de panne système est considérablement réduit lorsque le système tolère une panne. Ce scénario correspond au cas où les nœuds critiques sont dupliqués dans le réseau. Le taux de panne du système dépend également de la taille du réseau et des taux de panne des équipements *FR*. l'augmentation du nombre d'équipements de réseau augmente la probabilité d'occurrence de pannes au niveau du système, ce qui à son tour diminue la fiabilité du système.

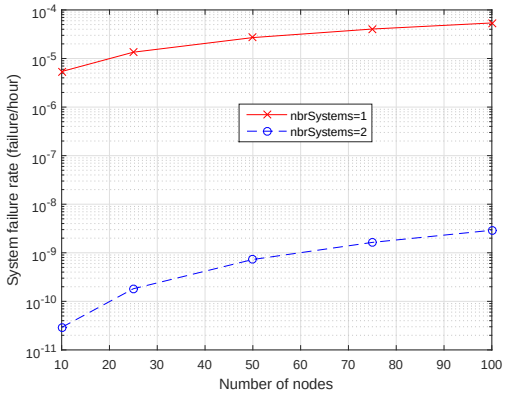


Figure D.13: Taux de panne du système Mono-ring vs taille du réseau et la redondance

La figure D.13 montre comment la duplication du système améliore la fiabilité globale du système. Dans ce scénario, un anneau ne tolère aucune faute et l'équipement FR est 10^{-7}. Comme on peut le voir, le taux de panne du système est considérablement amélioré lorsque le système est dupliqué pour les différentes tailles de réseau. De plus, le taux de panne est presque nul lors de l'ajout d'une troisième réplique.

Les résultats obtenus montrent que le niveau de fiabilité d'AeroRing répond aux con-

traintes avioniques, i.e., DAL-A, lors de l'utilisation d'une topologie mono-ring dupliquée. De plus, les résultats montrent que toutes les entités T-AeroRing ont presque le même impact sur la fiabilité du système, et que la topologie à multiple-ring a un niveau de fiabilité comparable à celui de la topologie mono-ring.

D.8 Validation sur une Étude de Cas Avionique

Dans ce chapitre, la validation des performances AeroRing, i.e., les niveaux de déterminisme et de fiabilité, est réalisée à travers une étude de cas avionique réaliste. Les performances d'AeroRing sont comparées au réseau avionique actuel d'un A380 basé sur l'AFDX et aux solutions ETR les plus pertinentes.

L'étude de cas considérée est un réseau avionique cœur représentatif d'un A380. Comme le montre la figure D.14, le réseau se compose de 8 switches AFDX reliant 54 end-systèmes (chacun relie entre 6 à 13 end-systèmes). Cette configuration est basée sur 432 différents VL répartis dans trois classes de trafic différentes (TC).

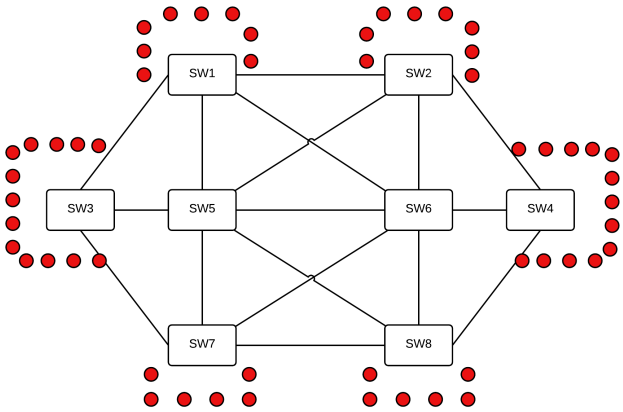


Figure D.14: Un réseau avionique cœur représentatif d'un A380

Afin d'étudier les performances d'AeroRing, nous remplaçons le réseau AFDX actuel par un réseau AeroRing. Ensuite, nous comparons les résultats obtenus avec le réseau AFDX et les solutions ETR les plus pertinentes. Il est à noter que les implémentations actuelles de l'AFDX et les solutions ETR considérées sont à base de 100 Mbps. Cependant, pour mener notre comparaison, nous augmentons leur vitesse à 1Gbps.

D.8.1 AeroRing vs AFDX et les Solutions ETR

Figure D.15 montre les bornes de délai de bout en bout maximales des différentes classes de trafic pour l'AFDX, AeroRing avec deux stratégies de service (FIFO et PS), EtherCAT et Profinet IRT. Toutes les solutions respectent les contraintes temporelles des différentes classes de trafic et AeroRing surpasse les solutions RTE, avec les deux stratégies de service. De plus, la politique de service SP surpasse AFDX pour TC1 et TC2, et offre un délai légèrement supérieur pour TC3. Ces résultats montrent les performances temporelles élevées d'AeroRing en référence à AFDX et aux solutions ETR.

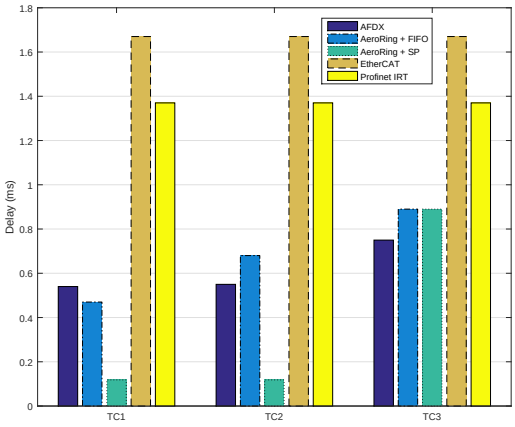


Figure D.15: Borne de délai maximale par classe de trafic

De plus, la figure D.16 montre une comparaison du temps de redondance maximum d'AeroRing et des différentes solutions RTE. Comme on peut le voir, AeroRing offre le meilleur temps de redondance. Ce résultat montre la haute disponibilité d'AeroRing, principalement due au mécanisme de redondance dynamique ARRP.

Tout d'abord, le mécanisme de détection de défaillance local mis en œuvre dans AeroRing garantit un temps de détection des défaillances plus rapide que les mécanismes de détection de défaillance centralisés ou globaux, où les messages de contrôle doivent traverser tout le réseau, comme implémenté dans EtherCAT et Profinet IRT. De plus, l'utilisation d'un seul message de contrôle, prioritaire et envoyé par le n¡ud détectant l'échec, permet de réduire le temps de récupération sous AeroRing.

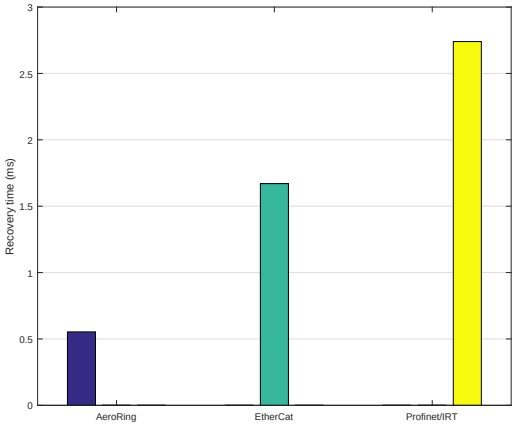


Figure D.16: Maximum recovery time

D.8.2 Fiabilité d'AeroRing

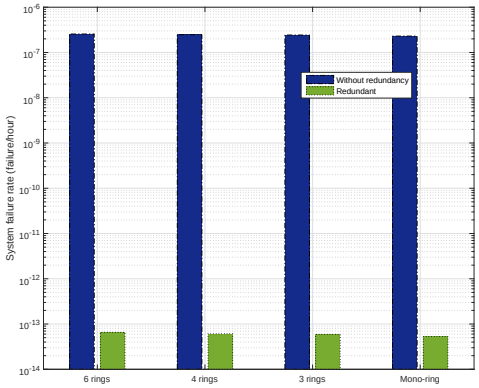


Figure D.17: Fiabilité d'AeroRing pour différentes topologies

La figure D.17 montre le taux de pannes d'un réseau AeroRing avec ou sans redondance selon plusieurs topologies du réseau. Comme on peut le voir, un réseau AeroRing redondant offre un niveau de fiabilité élevé avec un taux de panne inférieur à 10^{-13}, ce qui satisfait le niveau DAL-A requis dans l'avionique, i.e., taux de panne inférieur à 10^{-9}. De plus, nous pouvons remarquer que les différentes topologies ont des niveaux de fiabilité similaires.

D.9 Conclusion

Pour répondre aux besoins émergents de l'avionique, nous avons proposé dans cette thèse un nouveau réseau de communication avionique, nommé AeroRing, basé sur la technologie Gigabit Ethernet et une topologie en anneau. Ce réseau garantit l'interopérabilité avec l'AFDX grâce à sa conformité avec l'IEEE 802.3, ce qui facilitera son adoption. De plus, la topologie en anneau diminue la complexité du câblage, par rapport à la topologie commutée, tout en permettant un niveau de disponibilité élevé grâce au chemin redondant. Bien que l'intégration d'une telle solution présente de nombreux avantages en termes de réduction du poids et des coûts, elle introduit en même temps de nombreuses questions auxquelles il faut répondre pour être adoptée en avionique, principalement liées au déterminisme et fiabilité. Pour atteindre cet objectif, nous avons suivi une méthodologie spécifique avec les principales étapes suivantes.

Tout d'abord, nous avons conçu AeroRing pour intégrer diverses fonctionnalités favorisant les exigences avioniques comme: le time-triggered pour garantir la modularité, le traffic shaping et la politique de service SP pour favoriser le déterminisme et le protocole de gestion de redondance dynamique ARRP pour garantir la disponibilité.

Deuxièmement, pour prouver les performances temporelles d'AeroRing, nous avons modélisé notre solution en utilisant le Calcul Réseau en introduisant une nouvelle approche, nommé Pay Multiplexing Only at Convergence Points (PMOC), en prenant en compte les phénomènes de sérialisation des flux et en payant les bursts des flux interférents uniquement

aux points de convergence. Cette méthode a permis d'améliorer les bornes de délai, le passage à l'échelle et l'utilisation des ressources.

Troisièmement, pour évaluer le niveau de disponibilité d'AeroRing Nous avons défini et calculé les indicateurs de performance associés (le temps de détection de panne et le temps de de redondance), induits par le protocole ARRP.

Quatrièmement, pour mesurer le niveau de fiabilité d'AeroRing, nous avons effectué des analyses de fiabilité basées sur des réseaux de Petri stochastiques (SAN).

Enfin, nous avons analysé et validé les performances d'AeroRing pour un cas avionique réaliste. Les résultats ont montré qu'AeroRing surpasse les solutions ETR existantes en terme déterminisme et de disponibilité. De plus, AeroRing satisfait le niveau DAL-A requis dans l'avionique.

Bibliography

[1] Industrial Communication Networks - Profiles - Part 2: Additional Fieldbus Profiles for Real-Time Networks Based on ISO/IEC 8802-3. International standard, International Electrotechnical Commission, July 2014.

[2] Jean-Bernard Itier. A380 Integrated Modular Avionics - The History, Objectives and Challenges of the Deployment of IMA on A380. Technical report, ARTIST2 meeting on Integrated Modular Avionics, 2007.

[3] Hamdi Ayed. *Analyse et Optimisation des Réseaux Avioniques Hétérogènes*. PhD thesis, École Doctorale Mathématiques, Informatique et Télécommunications (Toulouse); 142547247, 2014.

[4] AFDX Protocol Tutorial. AFDX/ARINC 664 (1500-049), 2005.

[5] L Buckwalter. Avionics Databuses. Avionics Communications. *Inc., Leesburg*, 2003.

[6] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. *IEEE Std 802.3-2005 (Revision of IEEE Std 802.3-2002 including all approved amendments)*, pages 1–2695, Dec 2005.

[7] Gigabit Ethernet Alliance. Gigabit Ethernet: 1000BASE-T (whitepaper). Technical report, 2007.

[8] AEE Committee et al. Aircraft Data Network Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network, ARINC Specification 664. *Annapolis, Maryland: Aeronautical Radio*, 2002.

[9] INC Aeronauticcal radio. ARINC Specification 429 part 1-17. In *An ARINC document*, 2007.

[10] Cary R. Spitzer. MARK33 Digital Information Transfer System. *Avionics: Elements, Software and Functions*, 2016.

[11] Daniel A. Martinec. ARINC 429 Tutorial. *CRC Press LLC*, 2001.

[12] Yasemin Isik. ARINC 629 Data Bus Standard on Aircrafts. *Recent Researches in Circuits, Systems, Electronics, Control & Signal Processing*, pages 191–195, 2010.

[13] Hyun-Ho Choi, Jung-Min Moon, In-Ho Lee, and Howon Lee. Carrier Sense Multiple Access with Collision Resolution. *IEEE Communications Letters*, 17(6):1284–1287, 2013.

[14] Dinh Khanh Dang. *Analyse de Performance des Technologies sans Fil pour les Systèmes Embarqués Avioniques de Nouvelle Génération*. PhD thesis, Toulouse, ISAE, 2014.

[15] Ahmed Akl, Thierry Gayraud, and Pascal Berthou. A New Wireless Architecture for In-Flight Entertainment Systems Inside Aircraft Cabin. *International Journal on Advances in Networks and Services*, 4(1&2):159–175, 2011.

[16] WiMedia Alliance. Ecma-368 High Rate Ultra Wideband Phy and Mac Standard. *ECMA,*, 2008.

[17] Hermann Kopetz, Astrit Ademaj, Petr Grillinger, and Klaus Steinhammer. The Time-Triggered Ethernet (TTE) Design. In *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on*, pages 22–33. IEEE, 2005.

[18] Hermann Kopetz. Event-Triggered Versus Time-Triggered Real-Time Systems. *Operating Systems of the 90s and Beyond*, pages 86–101, 1991.

[19] Oliver Kleineberg and Markus Rentschler. Redundancy Enhancements for Industrial Ethernet Ring Protocols. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1–8. IEEE, 2010.

[20] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer Science & Business Media, 2001.

[21] Simon Perathoner, Ernesto Wandeler, and et al. Influence of Different Abstractions on the Performance Analysis of Distributed Hard Real-Time Systems. *Design Automation for Embedded Systems*, 2009.

[22] Rene L Cruz. A Calculus of Delay Part II: Network Analysis. *IEEE Trans. Inform. Theory*, 1991.

[23] Anna Charny and Jean-Yves Le Boudec. Delay Bounds in a Network with Aggregate Scheduling. In *Quality of Future Internet Services*. Springer, 2000.

[24] Bengt Jonsson, Simon Perathoner, Lothar Thiele, and Wang Yi. Cyclic Dependencies in Modular Performance Analysis. In *Proceedings of the 8th ACM international conference on Embedded software*, pages 179–188. ACM, 2008.

[25] L Tassiulas and L Georgiadis. Any Work-Conserving Policy Stabilizes the Ring with Spatial Re-use. *IEEE/ACM Trans. Netw.*, 1996.

[26] Charles Spurgeon. *Ethernet: the Definitive Guide*. " O'Reilly Media, Inc.", 2000.

[27] Norman Abramson. THE ALOHA SYSTEM: Another Alternative for Computer Communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285. ACM, 1970.

[28] Xinggang Fan, Zhi Wang, and Youxian Sun. How to Guarantee Factory Communication with Switched Ethernet: Survey of its Emerging Technology. In *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, volume 3, pages 2525–2530 vol.3, Nov 2002.

[29] Akihiro Takagi, Shinichi Yamada, and Shohei Sugawara. CSMA/CD with Deterministic Contention Resolution. *IEEE Journal on Selected Areas in Communications*, 1(5):877–884, 1983.

[30] Wei Zhao and Krithi Ramamritham. Virtual Time CSMA Protocols for Hard Real-Time Communication. *IEEE Transactions on Software Engineering*, (8):938–952, 1987.

[31] Wei Zhao, John A Stankovic, and Krithi Ramamritham. A Window Protocol for Transmission of Time-Constrained Messages. *IEEE Transactions on computers*, 39(9):1186–1203, 1990.

[32] M. Felser. Real-Time Ethernet - Industry Prospective. *Proceedings of the IEEE*, 2005.

[33] Jorg Sommer, Sebastian Gunreben, Frank Feller, Martin Kohn, Ahlem Mifdaoui, Detlef Saß, and Joachim Scharf. Ethernet–A Survey on its Fields of Application. *IEEE Communications Surveys & Tutorials*, 12(2):263–284, 2010.

[34] J-D Decotignie. Ethernet-based Real-Time and Industrial Communications. *Proceedings of the IEEE*, 93(6):1102–1117, 2005.

[35] *IEC 62439-3, Industrial Communication Networks - High Availability Automation Networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)*. 2016.

[36] *IEC 62439-6, Industrial Communication Networks - High Availability Automation Networks - Part 6: Distributed Redundancy Protocol (DRP)*. 2012.

[37] *IEC 62439-2, Industrial Communication Networks - High Availability Automation Networks - Part 2: Media Redundancy Protocol (MRP)*. 2012.

[38] *IEC 62439-7, Industrial Communication Networks - High Availability Automation Networks - Part 7: Ring-based Redundancy Protocol (RRP)*. 2011.

[39] Real-Time Ethernet: P-NET on IP: Proposal for a Publicly Available Specification for Real-Time Ethernet. Doc. IEC 65C/360/NP, 2004.

[40] Brian Field, Taieb F Znati, and Daniel Mosse. V-net: A framework for a Versatile Network Architecture to Support Real-Time Communication Performance Guarantees. In *INFOCOM'95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, volume 3, pages 1188–1196. IEEE, 1995.

[41] Real-Time Ethernet: Vnet/IP: Proposal for a Publicly Available Specification for Real-Time Ethernet. Doc. IEC 65C/352/NP, 2004.

[42] International Electrotechnical Commission, Real Time Ethernet Modbus-RTPS, Proposal for a Publicly Available Specification for Real Time Ethernet. document IEC 65C/341/NP, 2004.

[43] Gerardo-Pardo Castellote and Real-Time Innovations Inc. Peter Bolton. Distributed Real-Time Applications Now Have Data Distributed Protocol [Online], URL:"https://info.rti.com/hubfs/docs/RTC_Feb02.pdf". *Communication Update*, February 2002.

[44] Modicon MODBUS Protocol Reference Guide [Online], url:"http://modbus.org/docs/pi_mbus_300.pdf". June 1996.

[45] Real-Time Ethernet: TCnet (Time-Critical Control Network): Proposal for a Publicly Available Specification for Real-Time Ethernet. Doc. IEC 65C/353/NP, 2004.

[46] Paul Brooks. Ethernet/IP-Industrial Protocol. In *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, volume 2, pages 505–514. IEEE, 2001.

[47] Viktor Schiffer. The CIP Family of Fieldbus Protocols and its Newest Member-Ethernet/IP. In *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, pages 377–384. IEEE, 2001.

[48] A. Moldovansky, S. Balasubramanian, and B. Batke. Introduction to Device Level Ring. *ODVA 2009 CIP Networks Conference*, 2009.

[49] Application Guide, armorstart dlr reference architecture [online], url: "http://literature.rockwellautomation.com/idc/groups/literature/documents/at/290e-at001_-en-p.pdf". November 2012.

[50] Real-Time Ethernet: EPL (Ethernet Powerlink): Proposal for a Publicly Available Specification for Real-Time Ethernet. Doc. IEC 65C/356a/NP, 2004.

[51] J. Grieu. *Analyse et Evaluation de Techniques de Commutation Ethernet pour l'Interconnexion de Systemes Avioniques.* PhD thesis, INP, Toulouse, 2004.

[52] Dirk Jansen and Holger Buttner. Real-Time Ethernet: the EtherCAT Solution. *Computing and Control Engineering*, 15(1):16–21, 2004.

[53] EtherCat - the Ethernet Fieldbus [Online], URL:"www.ethercat.org".

[54] Raimond Pigan and Mark Metter. *Automating with PROFINET: Industrial Communication Based on Industrial Ethernet*. Wiley-VCH, 2008.

[55] SERCOS - the Automation Bus, url:"www.sercos.com/technology/sercos3.htm".

[56] R Schlesinger and A Springer. VABS-A New Approach for Real Time Ethernet. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 4506–4511. IEEE, 2013.

[57] Gunnar Prytz. A Performance Analysis of EtherCAT and PROFINET IRT. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 408–415. IEEE, 2008.

[58] Jérémy Robert, Jean-Philippe Georges, Eric Rondeau, and Thierry Divoux. Analyse de Performances de Protocoles Temps-Réel Basés sur Ethernet. In *Sixième Conférence Internationale Francophone d'Automatique, CIFA*, 2010.

[59] PA Manoj Kumar and B Sathish Kumar. A Study on the Suitability of Ethernet/IP and EtherCAT for Industrial Time Critical Applications. *International Journal of Future Computer and Communication*, 2(2):76, 2013.

[60] Lucia Seno, Stefano Vitturi, and Claudio Zunino. Real Time Ethernet Networks Evaluation Using Performance Indicators. In *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–8. IEEE, 2009.

[61] Juergen Jasperneite, Markus Schumacher, and Karl Weber. Limits of Increasing the Performance of Industrial Ethernet Protocols. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pages 17–24. IEEE, 2007.

[62] M. Schumacher, J. Jasperneite, and K. Weber. A New Approach for Increasing the Performance of The Industrial Ethernet System PROFINET. In *WFCS*, 2008.

[63] Ernesto Wandeler, Alexander Maxiaguine, and Lothar Thiele. On the Use of Greedy Shapers in Real-Time Embedded Systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(1):1, 2012.

[64] M. Fidler. Survey of deterministic and stochastic service curve models in the network calculus. *IEEE Communications Surveys Tutorials*, 12(1):59–86, First 2010.

[65] M. D. Schroeder, A. D. Birrell, and et al. Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-point Links. *IEEE J. Sel. Areas Commun.*, 1991.

[66] D. Starobinski, M. Karpovsky, and L. Zakrevski. Application of Network Calculus to General Topologies Using Turn-Prohibition. *IEEE/ACM Trans. Netw.*, 2003.

[67] Jens B Schmitt, Frank A Zdarsky, and Ivan Martinovic. Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once. In *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference-*, pages 1–15. VDE, 2008.

[68] Anne Bouillard, Bruno Gaujal, Sébastien Lagrange, and Éric Thierry. Optimal Routing for End-to-end Guarantees Using Network Calculus. *Performance Evaluation*, 65(11):883 – 906, 2008. Performance Evaluation Methodologies and Tools: Selected Papers from ValueTools 2007.

[69] W.C. Carter. A time for Reflection. In *In Proceeding of the IEEE 12th Int. Symp. Fault-Tolerant Computing*, Santa Monica, California, USA, 1982. FTCS-12.

[70] Manuel Barranco. *Improving Error Containment and Reliability of Communication Subsystems Based on Controller Area Network (CAN) by Means of Adequate Star Topologies.* PhD thesis, UNIVERSITAT DE LES ILLES BALEARS, 2010.

[71] Stefan Poledna. *Fault-Tolerant Real-Time Systems: The Problem of Replica Determinism - System Model and Terminology.* Kluwer Academic Publishers, 1996.

[72] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model Checking.* MIT press, 1999.

[73] RTCA DO. Do-254, design assurance guidance for airborne electronic hardware, 2000.

[74] RTCA. SC 167. *DO-178, Software considerations in Airborne Systems and equipment certification.* RTCA, Incorporated, 1992.

[75] Lorrie Tomek, Varsha Mainkar, Robert M Geist, and Kishor S Trivedi. Reliability Modeling of Life-Critical, Real-Time Systems. *Proceedings of the IEEE*, 82(1):108–121, 1994.

[76] James L Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall PTR, 1981.

[77] WG Bouricius, W Ct Carter, and PR Schneider. Reliability Modeling Techniques for Self-Repairing Computer Systems. In *Proceedings of the 1969 24th national conference*, pages 295–309. ACM, 1969.

[78] Thomas F Arnold. The Concept of Coverage and its Effect on the Reliability Model of a Repairable System. *IEEE Transactions on Computers*, 100(3):251–254, 1973.

[79] Joanne Bechta Dugan and Kishor S. Trivedi. Coverage Modeling for Dependability Analysis of Fault-Tolerant Systems. *IEEE Transactions on Computers*, 38(6):775–787, 1989.

[80] John F Meyer, Ali Movaghar, and William H Sanders. Stochastic Activity Networks: Structure, Behavior, and Application. In *International Workshop on Timed Petri Nets*, pages 106–115. IEEE Computer Society, 1985.

[81] Ali Movaghar. *Performability Modeling with Stochastic Activity Networks.* PhD thesis, Ann Arbor, MI, USA, 1985. AAI8520952.

[82] William Harry Sanders. *Construction and Solution of Performability Models Based on Stochastic Activity Networks*. PhD thesis, 1988.

[83] J Couvillion, Roberto Freire, Ron Johnson, W Douglas Obal, Muhammad A Qureshi, Manish Rai, William H Sanders, and Janet E Tvedt. Performability Modeling with UltraSAN. In *Petri Nets and Performance Models, 1991. PNPM91., Proceedings of the Fourth International Workshop on*, pages 290–299. IEEE, 1991.

[84] A Movaghar. Stochastic Activity Networks: A New Definition. In *Proc. of the IASTED Int. Conf. on Modeling and Simulation (MS'97)*, pages 27–30, 1997.

[85] M Abdollahi and A Movaghar. Application of Stochastic Activity Networks on Network Modelling. In *SoftCOM02. 10th International Conference on Software, Telecommunications and Computer Networks, Split, Dubrovnik, Croatia*, 2002.

[86] William H. Sanders and High performance Computing. UltraSAN - User's Manual, Version 3.0, 1995.

[87] D.J. Klinger, Y. Nakada, and M.A. Menendez. *At&t Reliability Manual*. Springer US, 1999.

[88] Industrial Communication Networks - Profiles - Part 1: Fieldbus Profiles. International standard, International Electrotechnical Commission, July 2014.

[89] Cheng-Shang Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[90] Rene L Cruz. A Calculus for Network Delay. I. Network Elements in Isolation. *Information Theory, IEEE Transactions on*, 37, 1991.

[91] Rajeev Agrawal, Rene L. Cruz, and et al. Performance Bonds for Flow Control Protocols. *IEEE/ACM Transactions on Networking (TON)*, 1999.

[92] Anne Bouillard, Laurent Jouhet, and Eric Thierry. Service Curves in Network Calculus: Dos and Don'ts. Technical report, 2009.

[93] Anne Bouillard, Nadir Farhi, and Bruno Gaujal. Packetization and Aggregate Scheduling. Technical report, INRIA, 2011.