Approches de sûreté de fonctionnement sur Ethernet temps réel : application à une nouvelle génération d'ascenseur

Ayoub Soury

▶ **To cite this version:**

Ayoub Soury. Approches de sûreté de fonctionnement sur Ethernet temps réel : application à une nouvelle génération d'ascenseur. Automatique. Université Grenoble Alpes, 2018. Français. NNT : 2018GREAT029 . tel-01881782

HAL Id: tel-01881782
https://theses.hal.science/tel-01881782v1

Submitted on 26 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

**DOCTEUR DE LA
COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES**

Spécialité : Automatique-productique
Arrêté ministériel : 25 mai 2016

Présentée par

Ayoub SOURY

Thèse dirigée par **Jean-Marc THIRIET**
et codirigée par **Denis GENON-CATALOT**

préparée au sein du **Laboratoire de conception et d'intégration des systèmes (LCIS)**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Approches de sûreté de fonctionnement sur Ethernet temps réel : application à une nouvelle génération d'ascenseur

Safety approaches for real time Ethernet : application to new lift generation

Thèse soutenue publiquement le **11 avril 2018**,
devant le jury composé de :

Monsieur, Michel Robert
Professeur des Universités, Université de Lorraine, Président
Madame, Mireille Bayart
Professeur des Universités, Université de Lille 1, Rapporteur
Monsieur, Laurent Cauffriez
Maître de Conférences, HDR, Université de Valenciennes et du Hainaut-Cambrésis, Rapporteur
Monsieur, Frédéric Kratz
Professeur des Universités, INSA Bourges, Examinateur
Monsieur, Jean-Marc Thiriet
Professeur des Universités, Université Grenoble Alpes, Directeur de thèse
Monsieur, Denis Genon-Catalot
Maître de Conférences, Université Grenoble Alpes, Co-Directeur de thèse

Résumé

La conception d'un réseau de communication de sécurité basée sur l'Ethernet temps réel répondant aux exigences de la norme PESSRAL, dérivée de l'IEC 61508, constitue la base de notre travail. Afin d'atteindre cet objectif, nous mettons en œuvre des mécanismes permettant de réduire la probabilité d'erreur et d'atteindre les niveaux d'intégrité de sécurité (SIL) par l'utilisation d'un système électronique déterministe. Avec un seul canal de communication, notre système doit être capable d'intégrer des fonctions critiques et non critiques sans remettre en cause la certification du système.

Lors de cet engagement nous proposons un système de communication industrielle basé sur l'Ethernet temps réel. Les interfaces de communication proposées répondent aux exigences de réactivité, de déterminisme pour garantir les contraintes temporelles imposées par le processus et la norme. Pour assurer la sécurité fonctionnelle des interfaces, nous avons proposé une surcouche de type "safety" qui implémente des fonctions de sécurité selon le concept du canal noir défini dans l'IEC 61508. En nous basant sur ces propriétés, nous avons réussi à classifier les solutions temps réel à base d'Ethernet en trois classes en fonction du temps de cycle. La surcouche "safety", basée sur la redondance de données, a permis de renoncer à la solution de redondance physique. Cette redondance de données duplique le temps de cycle initial du réseau qui satisfait néanmoins aux conditions de sécurité et temporelles de la norme.

Mots-clés: PESSRAL, IEC 61508, système de communication industrielle, Ethernet temps réel, réactivité, déterminisme, sécurité fonctionnelle, couche safety, canal noir.

Abstract

The design of a communication network with a real-time Ethernet-based security that meets the requirements of the PESSRAL standard, derived from IEC 61508, is the basis of our work. In order to achieve this goal, we implement mechanisms reducing the residual error probability and achieving Safety Integrity Levels (SIL) via a deterministic electronic system. Through a single communication channel, our system must be able to integrate critical and non-critical functions without compromising the system certification.

According to this commitment, we suggest an industrial communication system based on real-time Ethernet. The proposed communication interfaces meet the requirements of responsiveness and determinism in order to guarantee the temporal constraints imposed by the process and the standard. To ensure the functional safety of the interfaces, we have proposed a "safety" overlay that implements security functions according to the concept of the black channel defined in IEC 61508. Based on these properties, we have managed to classify the Ethernet-based real-time solutions into three classes in terms of cycle time. The overlay "safety", based on the redundancy of data, made it possible to give up the solution of physical redundancy. This data redundancy duplicates the initial cycle time of the network, which nonetheless satisfies the security and time conditions of the standard.

Keywords: PESSRAL, IEC 61508, industrial communication system, real-time Ethernet, determinism, functional safety, safety layer, black channel.

Remerciements

Je voudrais en premier lieu exprimer toute ma gratitude à mes encadrants de thèse, Jean-Marc Thiriet et Denis Genon-Catalot , non seulement pour leur disponibilité et leur soutien au cours de ces années de thèse, mais aussi pour avoir fait preuve, à plusieurs reprises, de leur confiance en moi et pour m'avoir aidé.

Je remercie les membres du jury : les rapporteurs Mireille Bayart et Laurent Cauffriez, les examinateurs Michel Robert et Frédéric Kratz pour avoir pris le temps d'examiner et d'évaluer mon travail et pour leurs remarques qui me permettent d'améliorer la qualité de ce manuscrit.

Je voudrais également exprimer ma reconnaissance à tous les gens que j'ai eu la chance de côtoyer durant ces quelques dernières années à Valence.

La durée d'une thèse est une performance temporelle qu'il est parfois difficile d'estimer à priori et qui dépend de nombreux "composants" interagissant entre eux. Pour des raisons de productivité (poursuite de la carrière) et de sécurité (santé mentale), il est préférable qu'elle n'excède pas une borne supérieure fixée à trois ans (quatre ans dans mon cas). Ces quelques mots sont donc là pour remercier tous ceux qui m'ont permis de mener à bien ce challenge.

Table des matières

Table des figures

Glossaire

Ack acquittement. 23

ADN4SE atelier de développement & noyau pour systèmes embarqués. 4, 5, 15, 16, 18, 19, 59, 61, 76, 89

AFDX Avionics Full DupleX switched Ethernet. 43

ASIC Application-Specific Integrated Circuit. 46, 52, 71, 73

ASnd Asynchronous Send. 48–50, 52, 57

AZF l'usine d'AZote Fertilisants. 1

BE best effort. 24, 43

BGLE briques génériques du logiciel embarqué. 2, 7, 15

CAL CAN Application. 31, 32

CAN Controller Area Network. 4, 21, 28, 30–33, 35, 46, 61, 63, 64, 87–89, 91

CI ControlNet International. 42

CiA CAN in Automation. 30, 32

CIM Computer Integrated Manufacturing. 64

CIP Control and Information Protocol. 42

CN Controlled Node. 41, 47, 49, 50, 52, 63–65, 67–70, 92

CRC cyclic redundancy check. 79, 80

CSC contrôle sécurité cabine. 17, 62, 63, 84, 89, 96, 97

CSM contrôle sécurité machinerie. 17, 62–64, 89, 96, 97

CSMA/CD Carrier Sens Multiple Access with Collision Detection. 33, 34, 36, 47

DEC Digital Equipment Corporation. 34

DLL Data link layer. 47

E/E/PE Electrical/Electronic/Programmable Electronic. 2, 4, 10, 11, 94

EIP Ethernet\Industrial Protocol. 42, 43, 46

EPL Ethernet PowerLink. 4, 19, 21, 39, 41, 43, 47–52, 56–58, 63–68, 70–73, 75, 76, 82, 84–86, 88, 89, 91, 94

EPSG Ethernet PowerLink Standardization Group. 39

ET event-triggered. 23

EtherCAT Ethernet for Control Automation. 4, 19, 38, 39, 43, 52–58, 63, 72–76, 82, 85, 86, 88, 89

EUC Element Under Control. 95

FDMS Fiabilité, Disponibilité, Maintenabilité, et Sécurité. 8, 24

FPGA Field-Programmable Gate Array. 46, 52, 71, 73

FSN Fonds National pour la société Numérique. 15

FTP File Transfer Protocol. 45

GESA gestion électronique des sécurités ascenseur. 2, 3, 59, 61–65, 76, 82, 84, 85, 88, 89, 93, 95

HTTP HyperText Transfer Protocol. 45

IA Investissements d'Avenir. 2, 15

ICN industrial communication networks. 1

IEA Industrial Ethernet Association. 42

IEC International Electrotechnical Commission. 1, 2, 4, 7, 10, 11, 18, 24, 33, 38, 58, 63, 79–82, 87, 88

intel Intel Corporation. 34

IP Internet Protocol. 15, 19, 34, 47, 88

MAC Medium Access Control. 26, 35

MN Managing Node. 41, 47, 49, 50, 52, 63–65, 67–70, 92

MTU maximum transmission unit. 52

NCS Networked Control System. 20

NMT Network Management. 91

NSR non-safety related messages. 82

NTP Network Time Protocol. 45

ODVA Open DeviceNet Vendors Association. 42

OS système d'exploitation. 94

OSI Open Systems Interconnection. 32, 34, 35, 45–47, 80, 88

Parc centre de recherche de Palo Alto. 33

PC Personal computer. 45

PDO process data object. 91, 92

PDU Protocol Data Unit. 37

PES programmable electronic system. 16–18, 59, 61, 94, 95

PESSRAL Programmable Electronic components and Systems in Safety Related Applications for Lifts. 2–5, 17, 18, 58, 59, 61–63, 82–85, 87, 88, 93–95, 97

PLC Programmable logic controller. 45

PReq PollRequest. 48–50, 52, 56, 68, 69, 91

PRes PollResponse. 49, 50, 52, 56, 69, 91

QoS qualité de services. 22, 43

RAMSS Reliability, Availability, Maintainability, Safety and Security. 8

RC Rate Constrained. 43

RPDO Received Process Data Objects. 92

RTE Real-Time Ethernet. 4, 5, 37, 38, 44, 79, 82, 83, 89

SAE Society of Automotive Engineers. 43

SCC systèmes de contrôle commande. 1, 4, 7, 11, 15, 18–20, 24, 25, 33, 34, 44, 58

SCR systèmes de contrôle commande en réseau. 4, 18–21, 30

SdF sûreté de fonctionnement. 1–5, 7–11, 17, 18, 22–24, 30, 32, 59, 61, 77, 78, 85, 87, 88

SDO Service Data Object. 91

SIL Safety Integrity Level. 3, 11, 18, 30, 34, 84, 87, 94

SoA Start of Asynchronous. 48–50, 52, 56

SoC Start of Cyclic. 48–50, 52, 56, 67, 69

SPIN système de protection intégré numérique. 11

SR safety related messages. 82

STRC systèmes temps réel critiques. 4, 25, 26

TCP Transmission control protocol. 47

TDMA Time Division Multiple Access. 29, 35, 37, 41, 47

TPDO Transmitted Process Data Objects. 91, 92

TRCM Temps réel à contraintes mixtes. 24

TRCR Temps réel à contraintes relatives. 23

TRCS Temps réel à contraintes strictes. 23

TT time-triggered. 23, 31, 43

TT Ethernet Time-Trgiggered Ethernet. 43

TTCAN Time-Triggered Communication on CAN. 31

UDP User datagram protocol. 47

VD Virtual devices. 32

Xerox Xerox Palo Alto Research Center. 33, 34

Introduction générale

Contexte

Depuis les bus de terrain avec des protocoles simples (format ASCII) vers des protocoles reposant sur les réseaux IP et basés principalement sur le standard Ethernet, les réseaux de communication industrielle (industrial communication networks (ICN)s en anglais) connaissent une évolution considérable. Depuis les années 80s, les spécialistes ont déclenché l'apparition de plusieurs protocoles de communication industrielle en augmentant les performances de déterminisme (token bus, token ring, etc.) et adoptés par plusieurs ICNs de l'époque comme : World FIP, Profibus, Sercos, CAN. Tous ces réseaux sont communément appelés bus de terrain ("fieldbuses" en anglais). Aujourd'hui les systèmes de contrôle commande (SCC) reposent de plus en plus sur des réseaux IP permettant ainsi d'augmenter leurs performances.

Conçus à l'origine pour des communications de processus, ces ICNs sont généralement utilisés dans des SCCs de haut niveau de criticité et sont ainsi caractérisés par leur aspect temps réel au niveau de la communication entre les équipements. Ces SCCs sont des installations où les défaillances du système de communication étaient jusqu'à présent des exceptions rares. Ils sont développés sous l'hypothèse que toutes les entités opératives sont légitimes et correctement installées. Désormais, ils sont devenus, comme les autres systèmes d'information, des cibles potentielles aux attaques en exploitant des vulnérabilités présentes dans les couches logicielles ou protocolaires de leurs équipements. Ces ICNs sont souvent appelés à satisfaire/ atteindre un niveau suffisant de safety. L'automatisation avait un double objectif :

en premier lieu, l'augmentation de la productivité du système technique en ce qui concerne la réduction des coûts, la fiabilité, la disponibilité, la qualité, etc,

en deuxième lieu, l'amélioration de la sécurité directe des intervenants.

L'apparition de la SdF est liée à la révolution industrielle. Elle est considérée comme une nécessité pour les systèmes industriels. L'objectif de la SdF est d'avoir un système : zéro accident, zéro arrêt, zéro défaut et zéro maintenance. Les domaines d'application qui exigent une SdF sont par exemple les domaines suivants : nucléaire, spatial, avionique, automobile, automatisation industrielle, etc. Afin d'atteindre un niveau de SdF dans les systèmes industriels, et d'éviter les accidents industriels, comme SEVESO en Italie (1976), l'usine d'AZote Fertilisants (AZF) à Toulouse (2001), etc., les établissements industriels déploient beaucoup d'effort pendant le processus de conception du système. Ce niveau de SdF est atteint par le respect et l'application des recommandations et des règlements de la SdF spécifiés par des normes et des guides de SdF rigoureux propres à chaque domaine. En effet, pour atteindre un niveau de SdF élevé, il faut faire un compromis entre les mécanismes de SdF nécessaires d'un système et les coûts économiques. L'International Electrotechnical Commission (IEC) développe et prépare des normes internationales reconnues dans le domaine de la SdF. Le but est de fournir des méthodes et outils d'analyse

et évaluation des équipements et de services du système. La SdF est très chère en matière de coût économique. L'apparition de la norme IEC 61508 n'est pas récente. Les industriels l'ont prise en compte depuis 1996. Elle est une norme française depuis 1999 [Ricque and Vieille, 2005].

Les normes dérivées de l'IEC 61508 comprennent par exemple les normes pour les procédés industriels (IEC 61511), le secteur du nucléaire (IEC 61513), la sécurité des machines (IEC 62061 et ISO 13849) ou encore le transport ferroviaire (EN 50126/EN 50128/EN 50129).

Problématique

Cette thèse est financée dans le cadre des Investissements d'Avenir (IA) pour les projets de type briques génériques du logiciel embarqué (BGLE). Cette thèse a été motivée par une collaboration entre SPRINTE, une PME spécialiste dans la conception des ascenseurs des bâtiments, et le laboratoire LCIS, spécialiste en sûreté de fonctionnement des systèmes communicants. L'objectif du travail est de numériser/informatiser le fonctionnement global de l'ascenseur en assurant le niveau de SdF requis et défini dans les standards de SdF pour ce type de système (dorénavant nous l'appelons démonstrateur D2 "ascenseur"). En effet, les normes applicables pour la conception d'un système de sécurité pour l'ascenseur sont :

EN 81-1 : spécifie les prescriptions de sécurité relatives à la conception et à l'installation des ascenseurs électriques,

PESSRAL : la norme relative aux systèmes électroniques programmables intégrés à la chaîne de sécurité d'un ascenseur (Programmable Electronic components and Systems in Safety Related Applications for Lifts (PESSRAL) en anglais). La norme PESSRAL est basée sur les spécifications et les directives fournies par les normes IEC 61508 et EN 81. Elle fait référence aux normes suivantes :

- ISO 22200 qui concerne la compatibilité électromagnétique des produits utilisés dans les ascenseurs.
- IEC 60664 qui spécifie l'isolation des équipements dans les systèmes basse tension.

En effet, la PESSRAL spécifie les exigences matérielles et logicielles dédiées pour assurer un niveau d'intégrité SIL 3. Néanmoins, elle n'a pas attribué de rôles à leurs mises en œuvre. Pour répondre à ces exigences et afin d'atteindre le niveau SIL 3, le système de contrôle d'ascenseur doit intégrer un système Electrical/Electronic/Programmable Electronic (E/E/PE). Ce système s'appelle un système de gestion électronique des sécurités ascenseur (GESA). Il est basé sur l'implémentation des fonctions relatives à la sécurité (safety) de l'ascenseur qui permettent de mettre tout le système dans un état sûr ou de le maintenir dans son état sûr en respectant les événements aléatoires spécifiques. L'état sûr demandé par la norme PESSRAL est défini comme l'ascenseur avec les freins déclenchés et la chaîne de sécurité ouverte. La détermination du niveau d'intégrité de sécurité de chaque composant dans le système a été réalisée selon la norme PESSRAL. Voici quelques exemples qui décrivent des scénarii d'utilisation de l'ascenseur :

Exemple 1 : Intervalle de temps entre la détection d'une erreur système (mémoire, bus ...) et l'ouverture de la chaîne de sécurité de l'ascenseur.

Exemple 2 : Intervalle de temps entre la détection du dépassement de la vitesse maximale de la cabine et le déclenchement du système de freinage.

D'après la norme PESSRAL, cet intervalle de temps correspond au temps de réaction du système qui est défini comme suit : "c'est l'intervalle de temps entre l'apparition d'un défaut du système et le déclenchement de l'action correspondante au niveau de l'ascenseur, qui correspond

à l'intervalle de temps entre une action sur l'ascenseur et la réponse de celui-ci. L'ascenseur étant en permanence dans un état permettant de fournir le niveau de sûreté requis."

Exemple : Suite à une action sur l'ascenseur (due au dépassement de la vitesse maximale par exemple) le temps de réaction du système correspond à l'intervalle de temps entre la détection d'une faute ou d'un état non sûr du système (détection de l'excès de vitesse) et l'établissement d'un état sûr du système (activation du système de freinage), ne doit en aucun cas dépasser 100 ms. Cet intervalle de temps doit contenir tous les évènements dus à l'action sur l'ascenseur comme par exemple : la détection/sortie du capteur, la transmission de l'information, le traitement dans le système GESA, responsable de la sécurité d'ascenseur, la transmission de l'information au système mécanique et l'activation du système de freinage.

Lors de notre engagement dans les travaux du démonstrateur D2 "ascenseur", nous nous sommes focalisés sur un élément de sécurité dans l'ascenseur, "la chaîne de sécurité", qui est un ensemble de contacts en série, dont l'objectif est d'autoriser ou non le déplacement de la cabine d'ascenseur après vérification des conditions de sa sécurité. Actuellement cette chaîne de sécurité repose sur des dispositifs électromécaniques. Celle-ci nécessite donc un grand nombre de câbles qui ont un impact direct sur le coût du produit ainsi que sur sa complexité d'installation et donc son coût d'installation.

Dans un souci d'économie (réduction des coûts d'installation, de maintenance, de certification ...) et de modernisation (gain en fiabilité, robustesse, évolutivité, capacité de détecter et d'identifier les défauts ...), nous proposons la numérisation de cet élément. En effet, nous devons analyser un cas de transition du composant électrique/ électromécanique vers des composants électroniques communicants en réseaux de cette chaîne. Nous souhaitons réaliser les fonctions de sécurité nécessaires non plus par des moyens électromécaniques mais via un système électronique programmable.

Apport de la thèse et principaux résultats

L'objectif du démonstrateur D2 est de concevoir et développer les fonctions de sécurité d'un ascenseur par des systèmes électroniques utilisant un noyau temps réel déterministe avec un objectif de certification vis-a-vis de la norme PESSRAL. Afin d'assurer le niveau de sécurité SIL3 sur l'ensemble de la nouvelle chaîne de sécurité numérique, ceci avec un seul canal de communication, nous devons pour cela utiliser un système de communication certifié SIL3. Dans un but de simplification, le réseau doit véhiculer des messages critiques et non critiques. L'intégration de la pile protocolaire sous le système GESA ne doit pas constituer un trop gros effort d'intégration. En matière normative le réseau sécurisé doit satisfaire les exigences normatives relatives à la communication.

Nous nous sommes impliqués dans la conception des interfaces de communication de sécurité en proposant un modèle correspondant à une pile de communication propre à respecter la SdF et adaptée au noyau déterministe, pour la monter sur un ascenseur. Cette solution doit permettre d'atteindre les niveaux d'intégrité de sécurité (Safety Integrity Level (SIL)). Dans ce cadre, il nous a été demandé d'élaborer un modèle de communication sécurisé à base de l'Ethernet temps réel pour l'automatisation du système de contrôle d'ascenseur en agissant principalement sur sa chaîne de sécurité. L'objectif de ce démonstrateur est donc de concevoir un PESSRAL en intégrant les éléments suivants :

— usage d'un bus sécurité certifié SIL3,

— application de la norme IEC 61508,

— hébergement des fonctions critiques et non critiques sur le même micro-contrôleur,

— capacité à intégrer des fonctions tierces non critiques sans remettre en cause la certification des fonctions critiques.

Organisation du mémoire

Cette thèse contient quatre chapitres. En effet, elle se divise en deux grandes parties. La **première partie** présente la numérisation et l'informatisation de la sécurité des applications électromécaniques. Cette partie contient deux chapitres.

Le **premier chapitre** se focalise sur la SdF en général, et la sécurité fonctionnelle en particulier. En effet, une étude bibliographique a été faite pour étudier avec précision les contraintes de SdF appliquée à l'électromobilité. Cette étude nous a permis de présenter les normes applicables en sécurité fonctionnelle pour réussir sa certification. Parlons de la certification en SdF, l'IEC 61508 s'impose. Et donc, elle sera détaillée dans le premier chapitre afin d'arriver à définir le périmètre fonctionnelle de ses dérivés, comme la PESSRAL . Pour spécifier cette norme, nous détaillons l'application métier : l'ascenseur en expliquant les différents modes du fonctionnement ainsi que les mesures de sécurité appliquées par les spécialistes et parfois imposées par la loi. À la fin du premier chapitre, après avoir défini le cadre du projet intitulé atelier de développement & noyau pour systèmes embarqués (ADN4SE), nous allons présenter notre approche dans l'automatisation d'un ascenseur en remplaçant ses systèmes mécaniques par des systèmes communicants de type système E/E/PE.

Dans le **deuxième chapitre**, nous étudions, en détails, la transition des SCC électromécaniques vers des systèmes communicants et plus intelligents dits systèmes de contrôle commande en réseau (SCR). Cette étude va nous permettre une prise de position par rapport aux observations similaires préexistantes, fondée sur une solution actualisée tenant en compte l'évolution continuelle de ces systèmes. En effet, nous identifions en détails les différentes exigences fonctionnelles d'un SCC (temps réel, déterminisme, disponibilité, réactivité). Ensuite, nous introduisons les critères de SdF dans ces systèmes temps réel critiques (STRC). Cette démarche sert à cerner la problématique autour de la communication dans ce type de systèmes. Pour pouvoir sélectionner les critères de choix du réseau adapté, nous avons étudié la communication dans les ascenseurs de : Schindler, Sprinte et Sodimas, les trois concurrents dans le domaine de la conception des ascenseurs. Nous nous sommes concentrés sur l'introduction de l'Ethernet, comme standard, dans un éventuel système. Cette étude nous a permis de classifier les approches temps réel à base d'Ethernet dites RTE en fonction du temps de cycle. Le choix de la solution retenue a été encouragé par l'utilisation de bus Controller Area Network (CAN) dans les ascenseurs actuels. Nous sommes convaincus que cette transition vers le standard Ethernet est une transition douce qui va encourager les partenaires du projet à l'intégrer dans leurs prochains produits.

La **deuxième partie** s'intéresse à la vérification fonctionnelle du modèle de communication proposé à base d'Ethernet. Cette partie va être décrite dans les deux chapitres suivants.

Dans le **troisième chapitre**, après avoir proposé les différents choix, nous procédons à la mise en œuvre et la vérification fonctionnelle de chaque solution. Au fait, notre engagement dans un autre démonstrateur s'intéressant aux robots coopératifs "co-botique" était conséquent au retrait de la PME SPRINTE du démonstrateur D2. Dans ce nouveau démonstrateur, notre choix du protocole EPL hérité de CAN parait insuffisant en terme temporels. En effet, la communication industrielle dans ce démonstrateur exige un temps de réponse inférieur à 1 ms. Après validation du protocole EPL sur le D2, nous avons validé le choix du protocole EtherCAT qui nous permet d'améliorer les performances temporelles. Cette vérification temporelles nécessite une prise en

considération de l'aspect "safety", vu la criticité des systèmes considérés. Alors, une approche de SdF basée sur le concept du "black channel" mettant l'accent sur la communication est proposée à la fin du chapitre trois. En effet, nous avons proposé certaines mesures qui nous ont permis de développer un ensemble de fonctions de safety qui s'intègrent dans le modèle de communication à base de RTE proposé.

Le **quatrième chapitre** va nous permettre de valider notre approche en nous plaçant dans les conditions extrêmes imposées par la PESSRAL avec des scénarii concrets. Ce chapitre représente le lien entre les deux premières parties. En effet, il représente l'intégration de la deuxième partie dans la première. Dans ce chapitre, nous allons identifier les différents modes de défaillance dans le système de contrôle d'ascenseur afin de spécifier les fonctions de protections de chaque mode de défaillance. En général, il s'agit d'une analyse des modes de défaillance du système d'ascenseur. À la fin de ce chapitre, nous présentons la synthèse de notre contribution par rapport à l'existant câblé. En effet, le réseau devient déterministe et qualifié en SdF puisqu'il garantit temporellement les temps d'arrêts d'urgences exigés par la norme PESSRAL. Ce résultat facilite la phase de certification du produit dans le cadre du projet ADN4SE.

Chapitre 1

Vers le remplacement d'architectures de sécurité électromécaniques par des architectures de sécurité numériques programmables

Sommaire

1.1 Introduction

La défaillance du service rendu par un SCC aboutit potentiellement à une catastrophe d'ordre économique, écologique ou humain. La conception de ces systèmes est alors régie par des normes et des guides de sûreté rigoureux édictés selon le domaine d'application. Le respect de ces normes est un élément essentiel afin de certifier ou qualifier le système avant son utilisation. Un objectif est de pouvoir accorder une confiance justifiée dans le système et les services qu'il rend. La terminologie associée à la SdF est particulièrement riche, parfois complexe et nécessite quelques rappels. Dans ce chapitre, nous commençons par définir la SdF en nous appuyant sur la littérature et surtout sur les travaux de Jean-Claude Laprie : [Laprie, 1985, Laprie, 1989, Laprie *et al.*, 1996, Avižienis *et al.*, 2001, Avižienis *et al.*, 2004, Arlat *et al.*, 2006]. Ensuite nous passons à la définition de la sécurité fonctionnelle en nous basant sur la norme générique IEC 61508. Nous citons

quelques normes dérivées de l'IEC 61508 en fonction du domaine (machine, nucléaire, automobile, ferroviaire, etc.). Cette section va nous permettre de décrire notre applicatif (l'ascenseur). Dans la section 1.4, nous introduisons les techniques d'automatisation pour le système de contrôle d'un ascenseur. Enfin, dans la section 1.5, nous détaillons le cadre du projet en rappelant les objectifs du projet BGLE dans lequel nous nous sommes engagés, en tant que laboratoire de recherche.

1.2 Sécurité fonctionnelle appliquée à l'électromobilité et contraintes de SdF

1.2.1 Taxonomie de la SdF

Selon [Laprie, 1989] [Villemeur, 1988], la SdF est considérée comme la science de défaillances. En effet, elle concerne la connaissance et l'évolution des défaillances afin de les mesurer, les prévenir et les maitriser. Jean-Claude Laprie et al. dans [Laprie *et al.*, 1996, Avižienis *et al.*, 2001] ont défini la SdF comme "la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre, le service délivré par un système est son comportement tel que perçu ou requis par ses utilisateurs. Comme le montre la figure 1.1, ils ont identifié les trois principaux éléments de la SdF :

— Attributs : des propriétés pour évaluer la SdF.

— Entraves : des évènements qui empêchent un système d'être sûr.

— Moyens : des solutions qui améliorent la sûreté du système afin d'assurer le niveau requis.

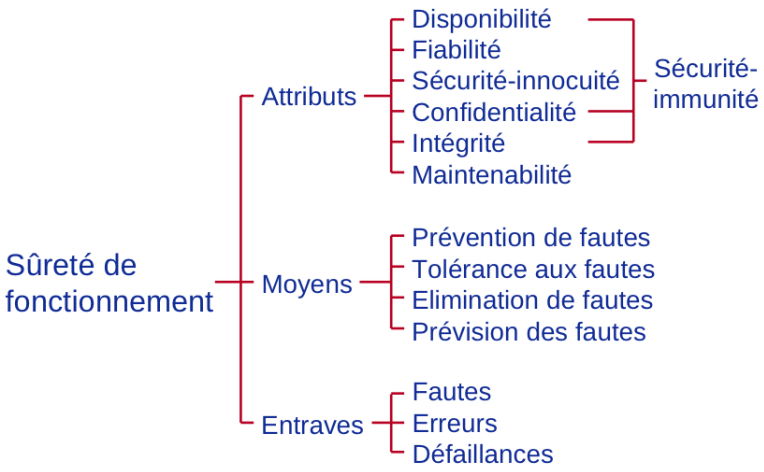


FIGURE 1.1 – Arbre de la sûreté de fonctionnement [Avižienis *et al.*, 2004]

1.2.1.1 Attributs

Pour qu'un système puisse être considéré comme sûr, il doit satisfaire certaines propriétés et critères. Cet ensemble de critères s'appelle des **attributs**. Ces attributs sont appelés Fiabilité, Disponibilité, Maintenabilité, et Sécurité (FDMS) ou Reliability, Availability, Maintainability, Safety and Security (RAMSS) en anglais [Villemeur, 1988, Laprie, 1989, Storey, 1996].

— **Fiabilité (Reliability)** : [Villemeur, 1988, Laprie, 1989] c'est l'aptitude d'un système à accomplir une fonction requise (bon fonctionnement) dans des conditions données pendant une durée donnée. Ce service doit être assuré pour toute la durée d'exécution du système ou jusqu'à un arrêt dans un état sûr (continuité de service). Cette durée de vie comprend la phase de fonctionnement nominal, mais aussi celle de fonctionnement dégradé.
— **Maintenabilité (Maintainability)** : [Villemeur, 1988, Laprie, 1989] c'est l'aptitude d'un système à être maintenu ou rétabli sur un intervalle de temps donné et dans des conditions d'utilisation données qui permettent au système d'accomplir une fonction requise (fonctionnement nominal). La maintenance est assurée à travers ; soit, l'élimination des fautes ayant produit des erreurs identifiées (maintenance curative), soit l'élimination des fautes par anticipation (maintenance préventive).
— **Disponibilité (Availability)** : [Villemeur, 1988, Laprie, 1989] c'est l'aptitude d'un système à être en état de délivrer un service correct (accomplir une fonction requise) au moment de sa demande (pendant un intervalle de temps donné) dans des conditions données. Elle est directement liée à la fiabilité et la maintenabilité (la disponibilité est le fait d'être prêt à utiliser). Les techniques qui assurent la disponibilité seront détaillées dans le chapitre 3.
— **Sécurité-innocuité (Safety)** : [Villemeur, 1988, Laprie, 1989] c'est l'aptitude d'un système à éviter l'occurrence des événements critiques ou catastrophiques sur l'environnement et/ou les personne et à assurer que le système va fonctionner.
— **Sécurité (Security)** :[Villemeur, 1988, Laprie, 1989] c'est l'aptitude d'un système à se protéger contre les défaillances relatives à des fautes intentionnelles (tentatives d'intrusion, divulgation non autorisée, etc.).

Les termes "sûreté" et "sécurité" ont la même racine latine "securitas" signifiant "sûr". Dans le cadre de la SdF, la sécurité/sûreté peut avoir deux significations distinctes (en anglais la sûreté est "safety/safe" tandis que la sécurité est security/secure) [Laprie, 1985]. La sécurité et la sûreté sont des attributs indissociables. Cependant, la sécurité constitue un domaine très vaste qui comprend différentes approches telle que la cryptologie. Cet aspect de sécurité ne sera pas étudié dans cette thèse.

Dans la littérature, d'autres attributs de SdF ont été identifiés comme par exemple ; la testabilité, la diagnosticabilité, la survivabilité, la résilience, etc. qui n'entrent pas dans le périmètre fonctionnel du projet. Par conséquent ils ne seront pas pris en compte dans cette thèse.

1.2.1.2 Entraves

Les événements qui peuvent affecter le fonctionnement d'un système, dégrader sa SdF (i.e. l'empêcher d'atteindre le niveau requis de la SdF) ou menacer sa SdF voire détruire le système sont appelés les entraves de la SdF. Selon [Laprie *et al.*, 1996, Avižienis *et al.*, 2001] ces entraves peuvent être réparties en 3 catégories : les défaillances, les fautes, et les erreurs. Leurs définitions sont récursives car ces entraves sont corrélées comme les montre la figure 1.2.

Défaillance (Failure) : traduit toute déviation du service délivré de l'accomplissement de la fonction requise.

Erreur (Error) : C'est la cause de la défaillance. Une erreur peut être de nature logicielle (variable erronée, pointeur non initialisé) ou de nature matérielle (usure). Il s'agit d'un comportement erroné qui affecte une partie de l'état du système qui est susceptible de conduire à une défaillance.

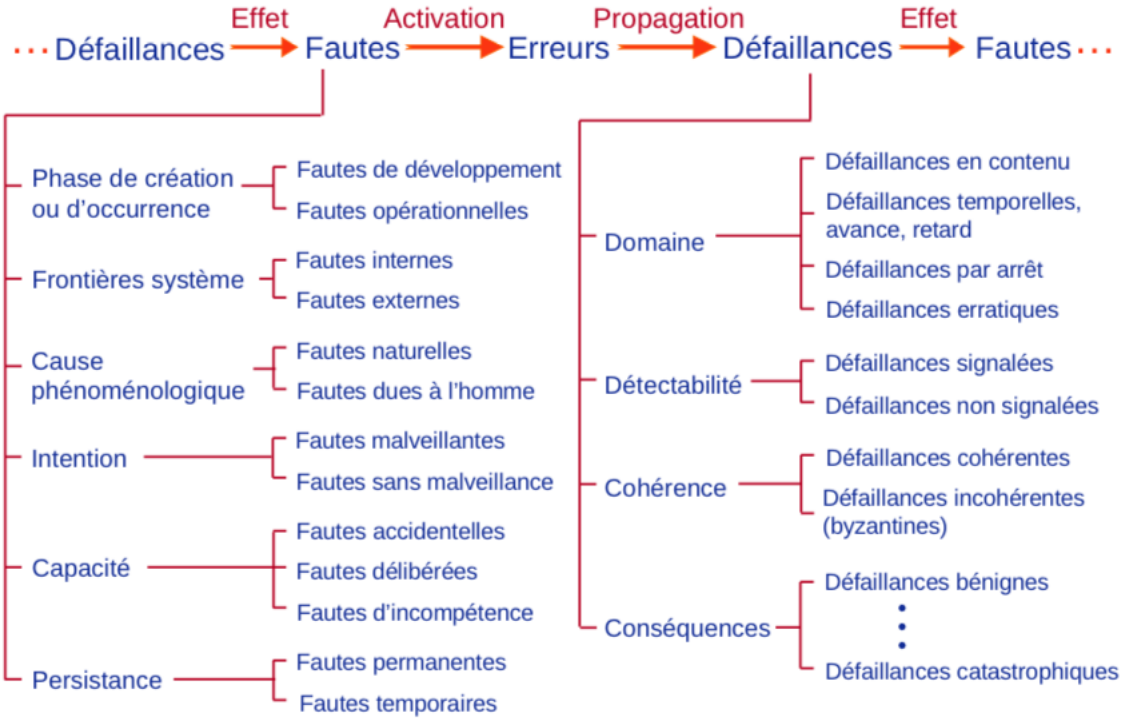


FIGURE 1.2 – Relation entre les trois types d'entraves de la SdF ; chaîne causale [Laprie *et al.*, 1996]

Faute (Fault) : C'est la cause de l'erreur. Selon [Avižienis *et al.*, 2001], la faute représente le premier élément de la chaîne causale entre les entraves de la SdF qui conduit à une panne. [Laprie, 1985] a pu classifier ces fautes selon deux principaux critères : origines (physiques/humaines, internes/externes, etc.) et persistance (permanentes/temporaires) :

1.2.1.3 Moyens

Dans le but d'atteindre un niveau requis de SdF, la conception d'un système sûr de fonctionnement doit se baser sur la garantie de l'exhaustivité des spécifications par rapport aux fonctions du système. Fonder cette garantie uniquement sur des tests n'est pas concevable. En effet, la complexité des systèmes visés complique de plus en plus l'analyse complète de tous les comportements et les erreurs. Pour casser la chaîne de défaillance (Faute–Erreur–Défaillance), le système doit se fonder sur quatre solutions éprouvées [Laprie, 1985]. Ce sont des techniques à mettre en œuvre pour atteindre un niveau de SdF requis et réduire les effets des entraves de la SdF :

— **Prévention des fautes** : Empêcher et éviter, par construction, l'occurrence de fautes, grâce aux méthodologies de développement et de bonnes techniques d'implantation.

— **Tolérance aux fautes** : Méthode fondée sur le principe d'admettre l'occurrence de fautes sans toucher au fonctionnement nominal du système en fournissant, par redondance (réaliser la même fonction par des moyens différents), un service conforme à la spécification du système. Il s'agit de la maîtrise de l'impact d'une faute.

— **Élimination des fautes** : Réduire et minimiser la présence de fautes au cours de deux phases du systèmes : phase de développement et phase d'utilisation.

— **Prévision des fautes** : Estimer, anticiper et évaluer la présence, la création et les conséquences de fautes en s'assurant que leur impact sur le service reste acceptable (basée sur des analyses et des études probabilistes).

1.2.2 Sécurité fonctionnelle et certification : l'IEC 61508 et ses normes dérivées

L'IEC 61508 est cohérente avec la convergence observée entre différents secteurs industriels : manufacturier, nucléaire, ferroviaire, aéronautique, développement de logiciels. C'est une norme générique qui s'applique à la réalisation de **fonctions de sécurité** en utilisant des technologies E/E/PE afin d'atteindre un niveau de **sécurité fonctionnelle** requis. Mais qu'est ce qu'une fonction de sécurité ?

Une fonction de sécurité est une fonction qui doit être implémentée dans un système E/E/PE concerné par la sécurité, dont le but est d'atteindre ou de maintenir un état sûr pour les équipements contrôlés, dans le cadre d'un évènement dangereux particulier. Dans ce cas, nous disons que ce système est destiné à atteindre l'intégrité de sécurité requise par la fonction de sécurité. La norme IEC 61508, dans sa quatrième partie, définit **la sécurité fonctionnelle** comme "un sous-ensemble de la sécurité globale, relatif aux équipements et aux SCC associés, qui dépend du fonctionnement correct de systèmes E/E/PE" [Smith and Simpson, 2004].

Exemple : Un équipement de protection thermique, utilisant un capteur de température dans les enroulements d'un moteur électrique pour déclencher le moteur avant une surchauffe, est un exemple de sécurité fonctionnelle. En revanche, fournir une isolation pour supporter de hautes températures n'est pas un exemple de sécurité fonctionnelle (bien que ce soit néanmoins un exemple de sécurité et puisse protéger exactement du même risque).

L'analyse de risques doit être identifiée par le spécificateur. Cette analyse permet au concepteur d'avoir une description détaillée des risques significatifs pour les équipements et les éventuels systèmes de contrôle associés. Cette description permet de déterminer si la sécurité fonctionnelle est nécessaire pour assurer une protection adéquate contre chaque risque significatif. Dans ce cas la sécurité fonctionnelle doit être prise en compte de manière appropriée lors de la conception. L'IEC 61508 définit la notion de niveau de sécurité intégrée(en anglais SIL) . En effet, cette notion de niveau, peut se définir comme une mesure de SdF qui permet de déterminer les recommandations concernant l'intégrité des fonctions de sécurité à assigner aux systèmes E/E/PE concernant la sécurité [Marszal and Scharpf, 2002]. Comme le montre le tableau 1.1 il existe quatre niveaux de SIL : le SIL4 étant le niveau de sécurité le plus élevé, le SIL1 le moins élevé.

La SdF vise à se prémunir des défaillances technologiques ou humaines. Le processus de conception est alors spécifié par des normes et des guides de SdF rigoureux propres à chaque domaine d'application :

— IEC 60880, IAEA pour le nucléaire : cette norme est destinée aux systèmes de protection et de sauvegarde des réacteurs nucléaires. En France, elle a été appliquée pour la conception et la qualification du système de protection intégré numérique (SPIN) [Chabrol, 2006]

— DO-178B pour l'aéronautique : cette norme fixe les conditions de sécurité applicables aux systèmes critiques de l'avionique.

— SAE J2640, J2056 pour l'automobile.

Type de conséquence	**SIL**	**Description et facteur de réduction du risque**
Catastrophique	4	Conséquence très importante sur la communauté et sur les employés (pertes humaines multiples) entraînant une réduction du danger de 10 000 à 100 000.
Critique	3	Conséquence très importante sur un individu de la communauté (perte d'une seule vie humaine) entraînant une réduction du danger de 1 000 à 10 000
Marginal	2	Protection importante de l'installation, de la production et des employés (blessures majeures à une ou plusieurs personnes) entraînant une réduction du danger de 100 à 1000.
Négligeable	1	Faible protection de l'installation et de la production (dommage matériel ou blessures mineures) entraînant une réduction du danger de 10 à 100.

TABLE 1.1 – Définition des 4 niveaux de sécurité intégrée (SIL) et leurs conséquences

Les normes dérivées de l'IEC 61508 comprennent par exemple les normes pour les procédés industriels (IEC 61511), le secteur du nucléaire (IEC 61513), la sécurité des machines (IEC 62061 et ISO 13849) ou encore le transport ferroviaire (EN 50126/EN 50128/EN 50129). Dans ce domaine, les normes EN 5012x sont basées sur le cycle de vie du système et ont été écrites afin d'adapter les exigences de la norme générique IEC 61508 aux contraintes de ce secteur. Le respect des prescriptions des normes EN 5012x suffit à assurer la conformité à la norme IEC 61508 sans qu'une évaluation complémentaire soit nécessaire.

1.3 Application métier : cas d'un ascenseur

Un ascenseur est composé des éléments principaux suivants : la gaine, la cabine, la machinerie, les portes, le moteur, le frein et les boitiers de commande. La figure 1.3 illustre la description technique d'un système de contrôle d'ascenseur qui représente les principaux composants ou les sous systèmes d'un ascenseur.

Seul deux acteurs sont susceptibles d'interagir avec le système comme indiqué dans la figure1.4 :

— l'utilisateur pour l'usage principal,

— le technicien pour la maintenance

Nous définissons les états de sécurité de l'ascenseur lorsque :

Le système est en mode nominal : l'ensemble des composants (matériel et logiciel) du système se trouvent dans un état de fonctionnement nominal c'est-à-dire en fonctionnement non dégradé,

le système est en mode sécurisé ce mode correspond à la coupure de l'alimentation du moteur et au déclenchement des freins. Cet état fait suite à une défaillance d'un des composants du système.

1.3.1 Différents modes du fonctionnement nominal d'un ascenseur

Le système possède trois modes transitoires qui permettent son démarrage, l'ensemble des diagnostiques nécessaires et la vérification de l'état de sécurité des composants avant la mise

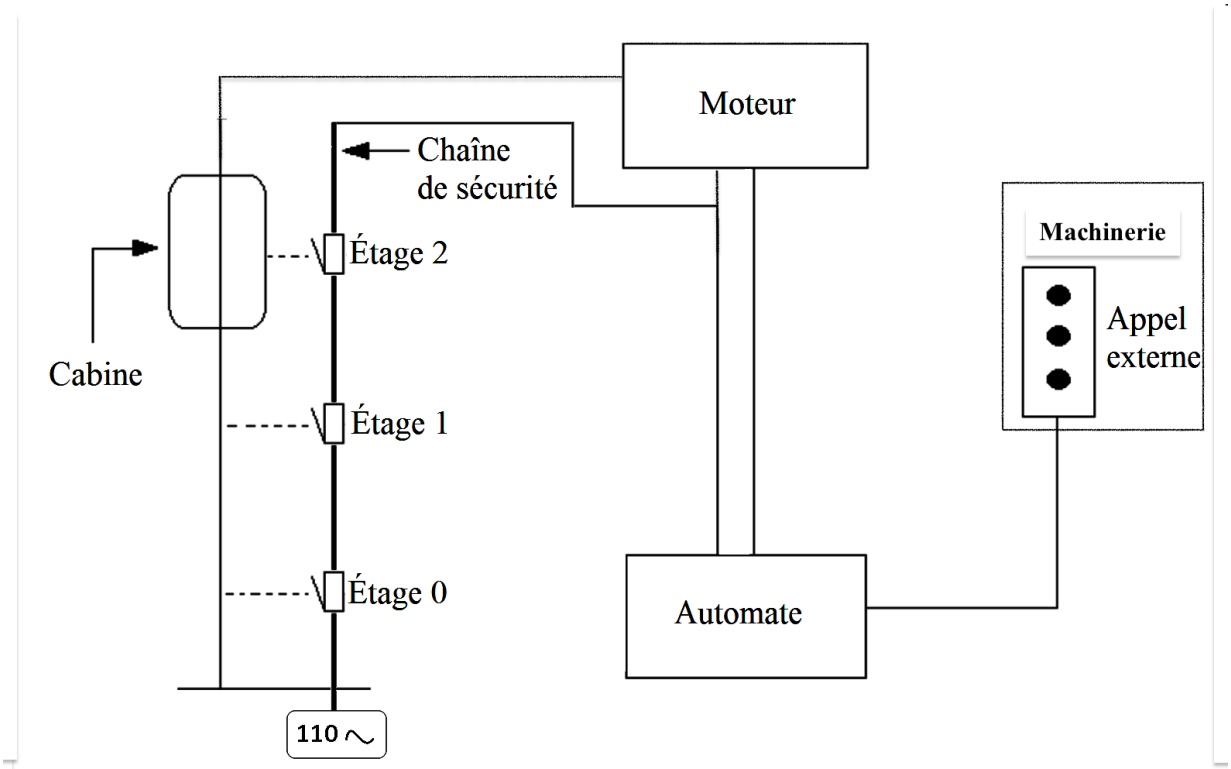


FIGURE 1.3 – Description technique d'un ascenseur

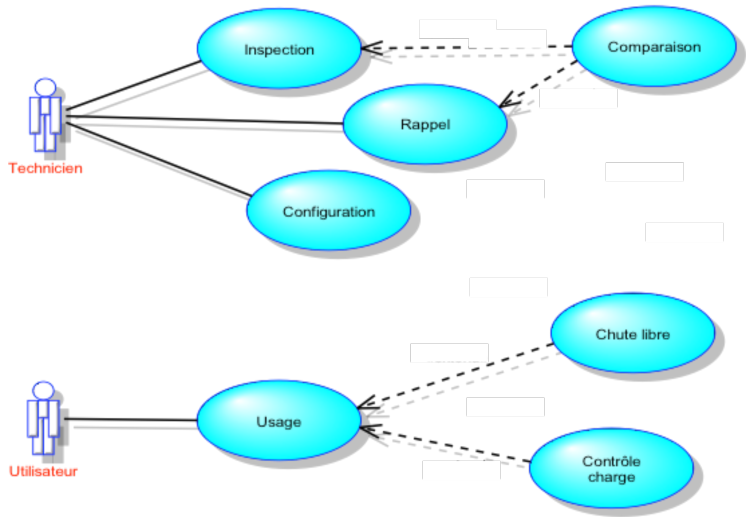


FIGURE 1.4 – Cas d'utilisation général

en place du système. Une fois que ces vérifications sont réalisées, le système entre en mode d'utilisation. Nous distinguons trois modes d'utilisation selon l'acteur qui souhaite utiliser le système (technicien ou utilisateur normal).

1.3.1.1 Fonctionnement du mode utilisateur

Après sa mise sous tension, le système se trouve en mode démarrage. Dans cette phase de démarrage, le système assure la configuration de différents composants ainsi que leur liaison. Ensuite, il procède à la vérification des différents états d'autres composants. Tant que ce mode de démarrage est activé, les freins doivent être en position fermée (enclenchée). Le système de freinage qui bloque le déplacement de la cabine se déclenche uniquement si l'ensemble des opérations précédentes se sont déroulées avec succès. Par la suite, le système passe automatiquement en mode auto diagnostique. Ce mode comprend des opérations de test qui sont reproduites de façon périodique. Si la phase d'auto diagnostique se termine avec succès le système passe automatiquement en mode utilisateur. Dans ce mode seules les commandes des boitiers cabines sont fonctionnelles.

1.3.1.2 Fonctionnement du mode inspection

Le passage à ce mode doit se faire après l'appel explicite du technicien à ce mode. Le passage en mode inspection doit neutraliser le fonctionnement normal de l'ascenseur et toutes les commandes provenant du boitier de rappel et des boitiers de cabine. Seules les commandes provenant du boitier d'inspection sont fonctionnelles. Le technicien qui intervient sur le toit de la cabine dispose d'un boitier de commande pour faire déplacer la cabine dans la gaine et commander les portes de la cabine. Le retour en fonctionnement normal ne peut se faire que par retour en position normal du commutateur. Pour faire déplacer la cabine en montée ou en descente, le technicien doit maintenir une pression sur le bouton du sens souhaité et sur le bouton d'activation. Les commandes d'ouverture ou de fermeture se font par l'appui maintenu du bouton du sens demandé. L'ordre est transmis à l'opérateur par commande réseau (actuellement le CANOpen) ou par relais statiques. Lors d'une demande de déplacement en inspection, si les portes ne sont pas fermées, un ordre de fermeture sera donné. Dans ce mode, des protections spéciales doivent être prises en compte :

- Une réserve minimale doit être garantie à chaque extrémité de la gaine. Si celle-ci n'existe pas, elle doit être recréée par des limites en gaine,
- Mode rappel non activé,
- La balustrade doit être relevée pour assurer la sécurité du technicien sur le toit,
- Les déplacements ne sont autorisés que si la chaîne de sécurité est fermée,
- La vitesse ne doit pas dépasser 0,63 m/s,
- La vitesse ne doit pas dépasser 0,3 m/s si la cabine se déplace dans un sens et qu'il reste moins de 2 mètres d'espace vital. (EN 81-20),
- La cabine doit s'arrêter s'il reste moins de 1m80 entre le toit de la cabine et le plafond (EN 81-20). Des commandes en montée restent possibles si le technicien appuie de nouveau sur le bouton montée tant que l'on n'atteint pas les limites autorisées.

1.3.1.3 Fonctionnement du mode rappel

La commande "rappel" se trouve en machinerie ou dans un coffret au palier pour les ascenseurs sans machinerie. Elle permet au technicien de faire déplacer la cabine en montée ou en descente pour placer la cabine à un étage afin de faire sortir les personnes bloquées. Pour faire déplacer la cabine en montée ou en descente, le technicien doit maintenir une pression sur le bouton du sens souhaité. Un voyant lui indique si la cabine se trouve dans la zone de porte d'un niveau. Dans ce mode, des protections spéciales doivent être prises en compte :

— la balustrade doit être pliée,

— impossibilité de se déplacer en rappel si on a le mode inspection,

— les déplacements ne sont autorisés que si la chaîne de sécurité est fermée et

— la vitesse ne doit pas dépasser 0,3 m/s.

1.4 Utilisation des technologies numériques pour l'automatisation des systèmes de contrôle commande

Depuis des années, le raccordement des armoires électroniques aux capteurs et aux actionneurs posait un problème. Pour chaque mesure, il fallait au moins deux fils. Pour éviter les chutes en ligne susceptibles de perturber les mesures, l'âme en cuivre devait être d'un certain diamètre. Le cheminement à proximité de câbles de puissance était cause de parasites. La correction d'une erreur de raccordement n'était pas une tâche simple. Pour remédier à ces problèmes, une idée est apparue progressivement. Elle consiste à incorporer au capteur (ou à l'actionneur) une électronique lui permettant de communiquer plus facilement. Nous parlons alors des capteurs /actionneurs "intelligents". Les raccordements ne sont plus effectués en fil-à-fil mais par l'intermédiaire d'un bus de terrain. Contrairement à l'ancienne technique du raccordement bifilaire, l'information n'est pas transportée sous forme analogique, mais en numérique. Les gains en fiabilité et simplification du câble sont indéniables. De plus, les informations annexes peuvent être données (paramètre de réglage, information concernant la maintenance, etc.) [Brenier, 2001]. L'art de l'autonomie des machines s'exerce en concevant et en réalisant des systèmes particuliers appelés systèmes de contrôle. Ces systèmes sont réalisés à l'aide des techniques de l'informatique, c'est à dire par programmation (mixage entre automatique et informatique). L'introduction des techniques de communication permet de diminuer les coûts d'infrastructure. Ainsi le remplacement des câbles par des bus dits "de terrain" a évolué depuis les protocoles simples de type Modbus (format ascii) vers le standard de l'Internet Protocol (IP). Le domaine de l'ascenseur a subi les mêmes évolutions que celui de la voiture ; coût de fabrication, réduction du câblage, optimisation énergétique, etc. avec des contraintes normatives. Dans une installation industrielle, les échanges automatiques de données entre les systèmes de contrôle et système informatique sont fréquents. Ils se produisent dans les deux sens. En effet l'utilisation accrue de l'informatique dans des domaines divers a aboutit à la création de multiples spécialités telles que les systèmes temps réel critiques [Chabrol, 2006], système embarqués critique, etc. Ces systèmes de contrôle ne cessent de profiter des progrès technologiques au niveau des composants physiques et logiciels. En général, un SCC est un système qui commande et contrôle (supervise) un autre système. Ces systèmes sont composés de calculateurs, de capteurs et d'actionneurs et de plus en plus souvent d'un système de communication. L'apparition de ces systèmes a fait évoluer les systèmes embarqués. Aujourd'hui, nous parlons plutôt de système embarqué temps réel. C'est un système qui met en valeur la contribution à la considération du comportement des plates-formes d'exécution logicielles temps réel [Lelionnais, 2014]. Les systèmes embarqués critiques regroupent deux propriétés qui sont "embarqué" et "critique". La notion d'embarqué se traduit par son enfouissement dans un autre système plus global. La deuxième propriété "critique" concerne les applications dont la défaillance d'un des composants peut occasionner des dégâts matériels importants ou des catastrophes humaines, écologiques ou économiques. Ces systèmes sont généralement des systèmes "autonomes" et "réduits" (en poids, volume, puissances de calcul, etc.) [Zammali, 2016]. Ces systèmes sont conçus pour être enfouis dans d'autres systèmes et afin de réduire leurs coûts, tous les systèmes embarqués critiques se caractérisent selon [Armoush, 2010] par :

— un poids et un volume réduits,

— des capacités mémoire et puissances de calcul réduites,

— une consommation énergétique réduite.

1.5 Présentation du contexte industriel

l'ADN4SE est un projet de recherche et développement débuté en septembre 2012 et financé dans le cadre des IA-BGLE, il s'est terminé en décembre 2015 et a regroupé 12 industriels : Sherpa Engineering (coordinateur), Safran, Schneider Electric, Alstom Transport, CEA-Saclay, CETIM, Krono-Safe (porteur de projet), BA Systèmes, Obeo, Itris Automation Square, INRIA Rocquencourt et notre laboratoire de conception et d'intégration des systèmes (LCIS) comme le montre la Figure 1.5.

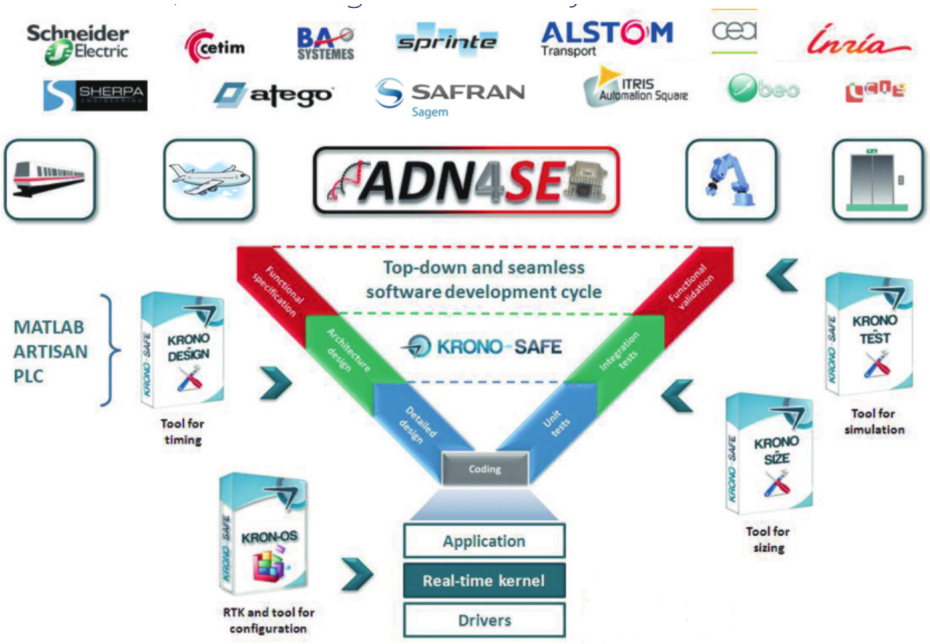


FIGURE 1.5 – Atelier de Développement & Noyau pour Systèmes Embarqués : Projet R&D

Budget global : 16,9 M euro

Funding : 7,2 M euro

Durée : 40 mois

Call : Fonds National pour la société Numérique (FSN)

Date début : Septembre 2012

Coordinateur : Sherpa engineering

L'objectif du projet ADN4SE portait sur la réduction des coûts de conception des logiciels embarqués en fournissant des outils de développement logiciel, de blocs logiciels génériques et des démonstrateurs (aéronautique, ferroviaire et d'automatisation industrielle) les mettant en œuvre en assurant un très haut niveau de sûreté logicielle.

1.5.1 Tâche du laboratoire LCIS

Dans le cadre du démonstrateur ascenseur (D2) pour le projet ADN4SE, nous avons analysé un cas de transition du composant électrique/électromécanique vers des composants électroniques communicants en réseaux. Nous nous sommes focalisés sur un élément de sécurité (chaîne de sécurité) qui est un ensemble de contacts en série, dont l'objectif est d'autoriser ou non le déplacement de la cabine d'ascenseur. Le présent système a pour objectif d'assurer la sécurité de la cabine d'un ascenseur.

Actuellement cette chaîne de sécurité repose sur des dispositifs électromécaniques comme le montre la Figure 1.6. Celle-ci nécessite donc un grand nombre de câbles qui ont un impact direct sur le coût du produit ainsi que sur sa complexité d'installation et donc son coût d'installation. Schneider Electric (partenaire dans le projet ADN4SE) fournit un certain nombre de ces équipements tels que des contacteurs et des variateurs de vitesse. Sprinte (constructeur d'ascenseurs et partenaire dans l'ADN4SE) est de ce point de vue un client et un intégrateur de différents produits, dont ceux de Schneider Electric.

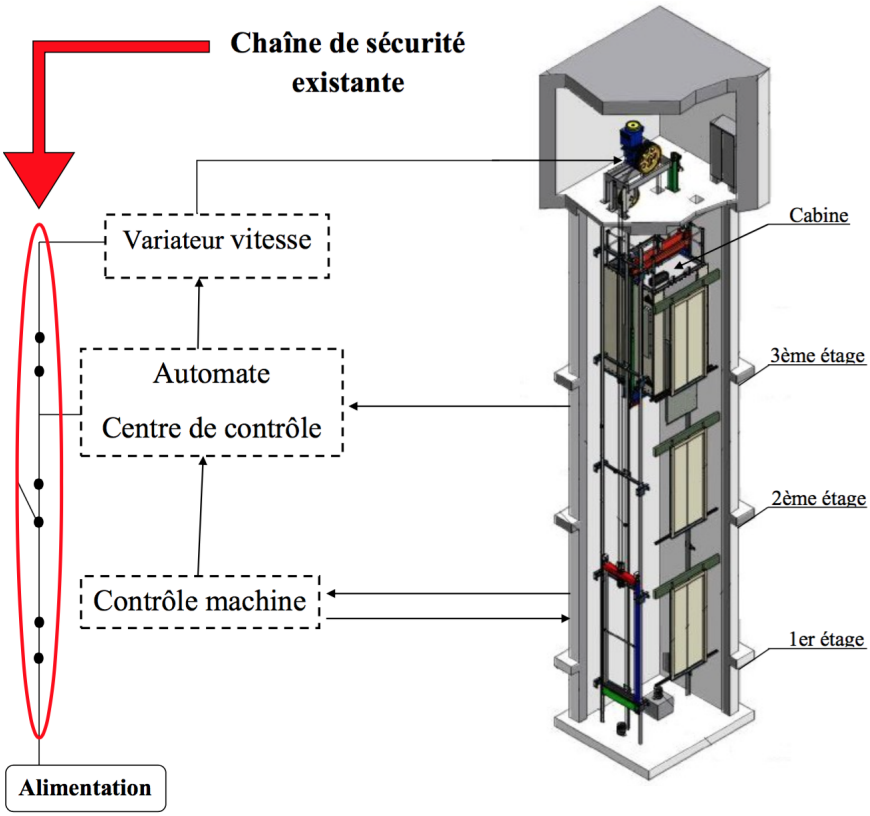


FIGURE 1.6 – Chaîne de sécurité dans les ascenseurs français.

Dans un souci d'économie (réduction des coûts d'installation, de maintenance, de certification ...) et de modernisation (gain en fiabilité, robustesse, évolutivité, capacité à détecter et identifier les défauts ...), nous souhaitons réaliser les fonctions de sécurité nécessaires non plus par des moyens électromécaniques mais via un programmable electronic system (PES). L'objectif du démonstrateur D2 est de concevoir et développer les fonctions de sécurité d'un ascenseur par des systèmes électroniques utilisant le noyau temps réel déterministe KronOS et les outils associés avec un objectif de certification vis-a-vis de la norme PESSRAL (en français "Conception et mise

au point des systèmes électroniques programmables dans les applications liées à la sécurité des ascenseurs").

Nous nous sommes impliqués dans la conception des interfaces de communication de sécurité en proposant un modèle de simulation correspondant à une pile de communication propre à respecter la SdF et adaptée au noyau déterministe, pour la monter sur un ascenseur. L'objectif de ce démonstrateur est donc de concevoir un PESSRAL en intégrant les éléments suivants :

— Usage d'un bus sécurité certifié SIL3.

— Application de la norme IEC 61508.

— Héberger des fonctions critiques et non critiques sur le même microcontrôleur.

— Capacité à intégrer des fonctions tierces non critiques sans remettre en cause la certification des fonctions critiques.

Pour atteindre notre objectif, nous avons pu identifier quelques contraintes qui doivent être satisfaites dans la réalisation du système :

Contraintes structurelles Le système électronique programmable (PES en anglais) est constitué de deux sous systèmes contrôle sécurité cabine (CSC) et contrôle sécurité machinerie (CSM). Ces deux sous systèmes communiquent via un bus sécurisé.

Contraintes d'évolution À terme, cette réalisation servira de base à la réalisation industrielle d'un système de protection électronique et numérique pour ascenseur.

Contraintes de développement L'OS utilisé est KronOS. Le langage de programmation est PsyC pour la description des modules (agents) et C pour le comportement des modules. La chaîne d'outillage de la société Krono-safe sera utilisée.

Contraintes de qualité Le démonstrateur doit se conformer à la norme PESSRAL en matière de sûreté de fonctionnement. Le niveau d'intégrité de sécurité le plus élevé étant SIL3.

1.5.2 Apport de Kron-OS en matière de SdF

La capacité de cloisonnement spatial et temporel fournie par le noyau Kron-OS nous permet d'intégrer les fonctions critiques et non critiques dans la même application. L'architecture logicielle a été orientée dans ce sens. La cohabitation de plusieurs SIL de niveau d'intégrité de sécurité différent dans le même PES est rendu de la même manière possible. La non interférence des fonctions critiques et non critiques est assurée par l'architecture (modularité, interface) et par l'OS (cloisonnement spatial et temporel).

Notre objectif est d'assurer le niveau de sécurité SIL3 sur l'ensemble de la chaîne de sécurité, ceci avec un seul canal de communication. Par conséquent, nous devons proposer un modèle de communication temps réel qui satisfasse les exigences normatives relatives au système de communication (éventuellement atteindre le niveau SIL3). Dans un but de simplification, notre réseau doit véhiculer des messages critiques et non critiques. L'intégration de la pile protocolaire sous Kron-OS ne doit pas constituer un trop gros effort d'intégration.

Un certain nombre de choix a été réalisé en respectant le périmètre restrictif du démonstrateur. Ces choix sont soumis aux contraintes suivantes :

— Contraintes logicielles (compatibilité de Kron-OS avec certains microcontrôleurs, disponibilité de la pile de communication pour le modèle de communication proposé).

— Coût des équipements.

— Disponibilité de certains équipements (ex : variateur de vitesse).

— Niveau de certification des équipements.

— Contraintes normatives.

1.6 Conclusion

Dans ce premier chapitre, nous avons introduit les notions basiques de SdF ainsi que la sécurité fonctionnelle liée aux systèmes électromobiles. Une étude de la littérature concernant la norme IEC 61508 a été présentée pour pouvoir identifier les caractéristiques de la SdF liés aux SCC en général. Ceci nous a permis de décrire notre applicatif qui est le système de contrôle d'ascenseur. Ce système représente la cible principale de nos travaux de thèse. L'objectif principal de nos travaux, dans le cadre du projet ADN4SE est de proposer un modèle de communication temps réel qui réponde aux exigences SIL3 de la norme PESSRAL propre aux systèmes de contrôle d'ascenseur. Ce modèle doit montrer ses capacités fonctionnelles et de sécurité ainsi que sa facilité d'intégration sur d'autres outils dans le cadre du projet. L'originalité de cette approche est l'utilisation des systèmes électroniques programmables pour gérer la sécurité. En effet, le chapitre suivant sert à identifier le périmètre fonctionnel du SCC en introduisant le système de communication. Dans le chapitre 2 nous allons analyser les différentes solutions utilisées dans les SCR.

Chapitre 2

Vers des systèmes communicants et intelligents

Sommaire

2.1 Introduction

Les systèmes industriels actuels tendent à distribuer les fonctions de communication, les fonctions de calcul et celles de contrôle à leurs différents composants. La connexion traditionnelle point à point a été la meilleure solution pendant des décennies. Cependant, avec l'extension des

fonctions et les installations physiques modernes, cette architecture point à point a touché les limites. Ce qui rend cette architecture inappropriée pour répondre à des nouvelles exigences telles que la modularité, la décentralisation de la commande et l'intégration du diagnostic. Dans une première partie de ce chapitre nous allons présenter les SCC. Ensuite nous expliquons les problèmes des réseaux dédiés (i.e. débit et distance limités, coût d'installation et maintenance élevé, problème d'évolutivité, etc.). Nous allons décrire les spécifications d'un réseau industriel afin de faciliter le passage des systèmes électromécaniques vers des systèmes communicants et plus intelligents. Nous nous concentrons sur l'évolution et la transition de la communication industrielle dans ce type de système, en particulier dans le système de contrôle d'ascenseur. Dans la deuxième partie du chapitre, nous allons présenter une synthèse de l'évolution du réseau IP afin de supporter tous les nouveaux services telle que la convergence IP et la mise en œuvre des contraintes de temps réel et de déterminisme. Ensuite, nous allons introduire une classification de différents protocoles qui utilisent Ethernet comme standard et qui sont adaptés aux SCR, en particulier les protocoles TTech, EPL et EtherCAT dans le cadre du projet ADN4SE. Enfin, nous expliquerons les critères du choix de protocoles utilisés dans le projet ADN4SE et qui seront évalués dans le chapitre 3.

2.2 Réseaux industriels dans les systèmes de contrôle-commande

2.2.1 Passage de la transmission analogique du système de contrôle commande à la transmission numérique

Au début de leur apparition, les SCC étaient analogiques composés d'architectures et de composants simples. Depuis l'apparition des technologies numériques, la communication numérique a participé à l'évolution du domaine en quelques décennies. Grâce à l'introduction de la communication au niveau du SCC, l'échange des données dans le système devient plus intelligent et joue un rôle important pour le fonctionnement du système. Ces systèmes sont appelés SCR selon [Addad, 2011] ou systèmes commandés en réseau selon [Ghostine, 2008] (Networked Control System (NCS) en anglais).

Comme le montre la figure 2.1, un SCR est composé d'un ensemble de capteurs/actionneurs dans la partie opérative du processus physique et des contrôleurs dans la partie commande du système. la liaison entre ces deux parties est assurée par un réseau de communication. Tous ces composants sont distribués autour d'un medium partagé qui permet de :

— assurer un lien de communication pour envoyer l'information entre chaque producteur et les consommateurs concernés,

— répondre aux contraintes de temps réel imposées par l'application,

— faciliter la maintenance (détection des fautes, simplification du câblage),

— participer à la reconfiguration (réorganisation des échanges suite à la défaillance d'un composant ou suite à l'adjonction d'un nouveau composant).

Dans les anciennes générations de SCC, l'acquisition de données était transmise via un lien analogique, i.e. un lien correspond à un seul signal. Or, les nouveaux systèmes traitent de plus en plus de mesures. L'installation d'un câble par signal amènerait alors à des difficultés d'installation et de maintenance et des coûts économiques élevés. Le remplacement de ces câbles analogiques dans les systèmes numériques par un réseau paraît une nécessité. L'introduction de la communication dans ce type de systèmes peut s'expliquer par différents avantages :

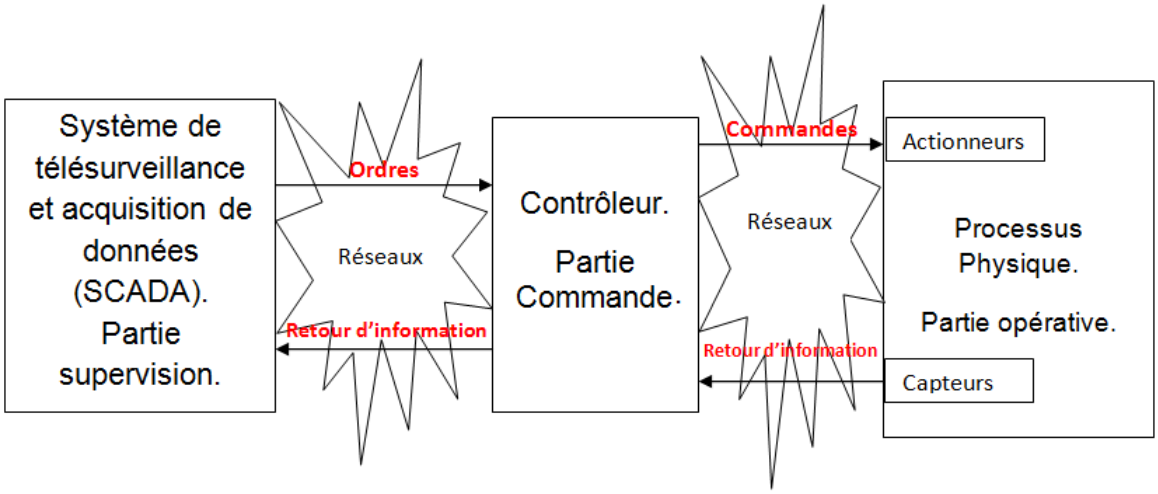


FIGURE 2.1 – Schéma simplifié d'un réseau de contrôle industriel.

— quantité d'information très importante : la communication numérique permet d'envoyer des flux binaires en série sur un seul câble au lieu d'utiliser un câble pour chaque variable comme c'était le cas en analogique. Elle permet ainsi d'échanger différentes informations entre composants distribués de plus en plus intelligents. Un technicien voulant tester/vérifier le fonctionnement d'un système/composant de terrain n'est plus obligé de se déplacer localement. Il est capable de faire le diagnostic, la maintenance et la supervision à distance grâce aux réseaux de communication.

— robustesse : les signaux transmis dans un système analogique (i.e. signaux transmis depuis un capteur) sont susceptibles d'être perturbés lors de leur acheminement à cause des parasites liées à la perturbation électromagnétique (la non compatibilité électromagnétique). Ces signaux peuvent subir une dégradation majeure lors de leur transmission et avant d'arriver à leur destination ce qui peut provoquer la falsification du comportement des autres composants (i.e. contrôleur). De plus, l'aspect binaire de la communication la rend plus robuste par rapport à l'existant. De plus ces communications sont riches des techniques de codage et de vérification d'erreur. La détection d'erreur dans un message est toujours possible et la demande de retransmission d'un message erroné est aussi possible. En conséquence, les transmissions numériques sont plus résistantes aux perturbations avec une capacité de détection et de correction d'erreur.

— flexibilité et évolution accrue : pour anticiper les plausibles évolutions matérielles d'un système industriel, son système de communication doit être flexible afin de faciliter les évolutions de l'architecture physique et logique du système ainsi que la combinaison d'équipements de différents fournisseurs.

— interopérabilité : les constructeurs fabriquent leurs propres équipements qui ne sont compatibles qu'avec leurs propres protocoles, ce qui entraîne des difficultés majeures dans le marché : faire communiquer ensemble des composants de différents constructeurs reste toujours une tâche difficile voire impossible dans certains cas. Généralement, les produits d'un seul constructeur ne sont pas suffisants pour couvrir tous les besoins d'un SCR. L'utilisation de produits de plusieurs fournisseurs devient alors incontournable. La communication numérique à base d'Ethernet (voir section 2.4.1 36) est venue pour remédier à cette difficulté.

— réduction du coût et possibilité de distribution : la même paire de câble permet de connecter plusieurs composants avec leurs configurations afin de former un réseau sur un medium partagé. Par conséquent, la quantité des câbles utilisés sur les longues distances est réduite notamment le coût d'installation. La configuration d'un réseau permet de créer une certaine hiérarchie et d'adopter plusieurs protocoles d'échange d'information, ce qui évite toute ambiguïté. Par conséquent, la réduction des coûts de câblage est assurée grâce aux techniques de multiplexage (plusieurs signaux sont émis sur un même support).

Selon [Addad, 2011], il existe trois types de protocoles d'échange utilisés dans les SCR :

— Maître esclave : avec ce type de protocole, un esclave ne peut émettre d'information que lorsqu'il est interrogé par un maître. En effet, le maitre interroge d'une manière cyclique les esclaves qui prennent la parole à tour de rôle pour accéder au medium et pouvoir répondre au maître en émettant les données correspondantes (i.e. EPL, Profibus DP).

— Client/ Serveur : le client interroge le serveur d'une manière autonome qui répond à sa requête (Modbus TCP).

— Producteur consommateur : chaque fois qu'une variable change d'état, le nœud producteur émet une information qui sera envoyée au consommateur intéressé par la variable (worldFIP).

2.2.2 Exigences d'un système de contrôle commande et propriétés du réseau industriel adapté

En bureautique comme en industrie, les besoins des utilisateurs de réseaux informatiques évoluent régulièrement. Dans la bureautique, le paramètre de la qualité de service le plus connu est le débit maximum de données autorisée. La solution filaire dominante est Ethernet qui a continuellement évolué depuis son origine dans les années 80. Pourtant, son développement dans les applications industrielles de communication entre équipements d'entrées-sorties et équipements de traitement des données est relativement récent. Ces applications utilisaient essentiellement des solutions dites de bus de terrain (par exemple, CAN, Interbus ou Sercos). Celles-ci prennent en compte des paramètres de qualité de services (QoS) spécifiques à l'industrie tels que **la réactivité** ou **le déterminisme** [Limal, 2009]. Ce n'est que récemment qu'Ethernet est devenu financièrement et techniquement intéressant. Il est désormais annoncé comme le remplaçant des bus de terrain actuel (voir section 2.4.1 page 36). Le système de contrôle d'ascenseur doit intégrer un nouveau réseau de terrain qui satisfasse aux nouvelles exigences des clients sur les paramètres de qualité de services. Parmi celles-ci, l'exigence de réactivité est particulièrement contraignante, d'autant qu'elle s'accompagne d'**une exigence de disponibilité**. Une coupure de communication doit non seulement être compensée, mais elle doit l'être de manière à ce que l'exigence de réactivité continue d'être respectée. Avec certaines applications manufacturières, le passage en position de repli (non productive) engendré par une défaillance a une conséquence financière qui incite à rechercher une meilleure disponibilité. Avec une application de déplacement des personnes, les conséquences possibles des problèmes de communication sur la sécurité des biens et des personnes imposent de considérer la disponibilité comme composante de la SdF. Les trois sous-sections suivantes détaillent les trois caractéristiques du réseau de terrain : la réactivité, le déterminisme et la disponibilité.

2.2.2.1 Exigence de réactivité

Les performances d'un réseau industriel sont liées aux exigences du système qu'il commande. Le délai de réaction d'un système est sa capacité de réactivité. Ce délai est défini comme le délai

entre l'occurrence d'"événement cause" (entrée) et son correspondant "événement conséquence" (sortie). Comme l'illustre la figure 2.2, un délai de réaction représente le temps nécessaire pour mettre à jour l'état d'un signal. Ce délai représente le changement d'état d'un signal d'entrée qui représente l'état d'un capteur et sa conséquence sur le signal de sortie qui commande un actionneur du processus. Cette illustration est inspirée de celle utilisée par Gaëlle Marsal dans son mémoire de thèse [Marsal, 2006].

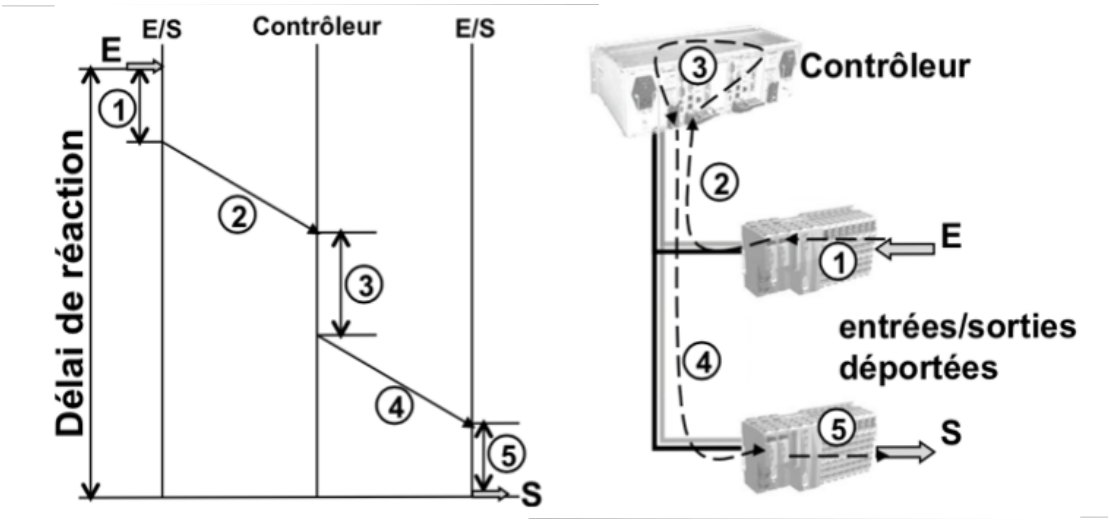


FIGURE 2.2 – Exemple illustrant le terme de délai de réaction[Limal, 2009].

Dans cet exemple, le délai de réaction regroupe :

1. le délai d'acquisition d'un signal d'entrée E sur une interface d'entrées-sorties déportée ou sur un contrôleur de terrain,
2. le délai de transmission S à un contrôleur pour traitement,
3. le délai de traitement,
4. le délai de transmission vers un équipement d'entrées-sorties (ou un contrôleur de terrain) tiers et
5. le délai de mise à jour d'un signal de sortie S par l'intermédiaire de son fond de panier.

La réactivité du système va dépendre en partie de la capacité du réseau de terrain à respecter les contraintes temporelles données voire exigées (temps maximum pour lequel le système peut être qualifié de réactif). Par conséquent, nous dirons qu'un réseau de terrain est réactif s'il respecte la contrainte temporelle dans des conditions d'utilisation fixées (temps de propagation d'un signal sur le réseau physique). Pour mettre au point un système de contrôle réactif, deux paradigmes différents peuvent être utilisés dans un réseau de communication. L'auteur de [Kopetz, 1991] compare les deux approches par rapport aux propriétés temporelles dans un réseau de communication tandis que [Limal, 2009] l'étend à l'utilisation dans les réseaux industriels :

— Réseau event-triggered (ET) : est un réseau basé sur les évènements. Il se focalise sur l'occurrence des évènements dans un espace temps continu. Dans cette approche, l'émetteur connaît le moment de transmission (émission) de message. Cet évènement peut imposer de reporter un traitement en cours (problème de collision, non déterminisme, congestion, ...). Dans ce cas, la détection d'erreur est basée sur un time-out déclenché par l'émetteur qui attend un message d'acquittement (Ack) du récepteur. L'intérêt d'une telle approche est que

l'utilisation du réseau est optimisée. La difficulté est d'être capable d'évaluer la robustesse de la caractéristique de réactivité d'un réseau event-triggered dans des conditions d'utilisation limites (en cas de nombreux évènements simultanés). Une politique d'ordonnancement des tâches doit être prise en compte pour gérer les différents rythmes d'occurrences d'évènements dans le réseau. Les approches ET ne satisfont pas certaines propriétés liés à la SdF. En effet, le déterminisme et la ponctualité peuvent ne pas être garanties à causes des asynchronismes incontrôlés [Chabrol, 2006].

— Réseau time-triggered (TT) : la propagation des événements est assurée via des messages d'état d'entrée/sortie qui vont être transmis cycliquement (même si l'état ne change pas). Par conséquent, l'utilisation du réseau n'est pas optimisée, comme c'était le cas avec un réseau ET. Par contre, ce changement ne met pas la réactivité du réseau en défaut. Au contraire, la contrainte temporelle de réactivité à respecter par le réseau pourra facilement être comparée à un multiple du temps de cycle dans des conditions d'utilisation données. L'exécution cyclique du traitement se fait dans un intervalle de temps prédéfini avec une échéance prédéterminée (instant d'activation et instant de fin) [Chabrol, 2006].

Les notions de système réactif et de système temps réel sont souvent assimilées [Pailler, 2006]. Selon [De Carvalho, 1996, Chabrol, 2006], les différentes catégories d'un système temps réel sont :

— Temps réel à contraintes strictes (TRCS) : la garantie de toutes les contraintes temporelles est essentielle dans de tels systèmes. Le non-respect d'au moins une, peut causer des erreurs graves qui engendrent des conséquences dangereuses. En amont, le dimensionnement du système, la définition des conditions d'utilisation et l'analyse de l'ensemble des scénarios représentent les éléments importants permettant de satisfaire toutes les contraintes (aucune tolérance accordée à un dépassement d'échéance).

— Temps réel à contraintes relatives (TRCR) : cette catégorie du système TR est moins exigeante. Elle tolère, dans une certaine limite des dépassements ou manquements d'échéance sans provoquer des conséquences dangereuses (exemple les applications multimédia : nombre d'image par seconde dans une vidéo). La métrique de ces systèmes se mesure par des analyses probabilistes.

— Temps réel à contraintes mixtes (TRCM) : Il s'agit d'un système avec un mixage des contraintes strictes et relatives.

2.2.2.2 Exigence de déterminisme

la SdF des logiciels exige que le système soit déterministe. En effet, le déterminisme est un principe fondamental pour la SdF des SCC. [Felser, 2005] a défini qu'un système devient déterministe "si pour chaque état du système et chaque combinaison des entrées, il existe une seule combinaison des sorties et l'état suivant du système peut être déterminé". Nous distinguons 3 variantes de déterminisme pour les SCC : déterminisme causal, déterminisme système et déterminisme temporel [Chabrol, 2006].

— Déterminisme système : pour les systèmes temps réel critique, ce type de déterminisme correspond à la garantie d'une terminaison. Une tâche doit obligatoirement se terminer.

— Déterminisme causal : si l'état suivant du système ainsi que la combinaison des sorties sont connus pour un état du système et une combinaison d'évènement d'entrée « c'est le principe d'après lequel tout phénomène régi par une ou plusieurs lois, entraîne les mêmes effets pour des causes et des conditions identiques. » Cette variante implique les propriétés de prédictibilité et de reproductibilité. Ce déterminisme causal est assuré (garanti) par le processus de contrôle du système (les contrôleurs) [Limal, 2009].

— Déterminisme temporel : "c'est une exigence préalable qui sert à majorer le temps d'exécution d'une tâche dans son environnement complet" en général. Avec cette définition, on peut connaitre le délai de réaction pour l'émission des combinaisons de sorties quel que soit l'état d'entrée. Il faut que ces délais soient inférieurs à la contrainte temporelle. Si cette propriété de déterminisme n'est pas respectée, le système n'est plus déterministe et sera qualifié best effort (BE) (sans garantie de délai de réception) [Felser, 2005], i.e. un système BE ne pourra pas garantir les contraintes temporelles imposées par le processus à contrôler [Limal, 2009, Chabrol, 2006].

Si on s'appuie sur ces deux concepts de déterminisme et de réactivité, on peut dire que **le déterminisme temporel et la réactivité** sont directement liés aux réseaux de communication dans le système. Par conséquent, le délai de réactions et de transmission, doivent être connus et bornés.

2.2.2.3 Exigence de disponibilité

Les processus critiques requièrent un niveau relativement élevé de SdF. Les auteurs de [Arlat *et al.*, 2006] étendent le quatuor FDMS de la SdF aux attributs suivants : fiabilité, disponibilité, maintenabilité, safety, confidentialité et intégrité. Pour la communication, on s'intéresse particulièrement à la disponibilité, l'intégrité et la sécurité-innocuité (le terme "safety" va être utilisé dans la suite du rapport pour désigner la sécurité-innocuité). La norme IEC 61784 [61784-3, 2016] propose l'utilisation des protocoles spécifiques indépendants du support de communication pour garantir l'intégrité et assurer le niveau de sécurité demandé. Alors que la fiabilité et la confidentialité seront héritées des équipements utilisés.

La disponibilité du réseau permet au SCC de garantir son déterminisme temporel et sa réactivité. La norme IEC 61508-4 impose "l'existence de plus de moyens qui sont strictement nécessaires pour accomplir une fonction requise dans une unité fonctionnelle ou pour représenter des informations par des données" . Par conséquent, cette disponibilité nécessite la redondance des composants logiciels et/ou matériels.

2.2.3 Caractéristiques d'une communication dans les systèmes temps réel critiques

Dans l'industrie moderne, les STRC nécessitent un réseau capable de supporter des transmissions avec des contraintes temporelles fortes. Selon [Malcolm *et al.*, 1990] et [Malcolm and Zhao, 1995]. Cette communication doit répondre aux exigences suivantes :

— Le délai de transfert peut subir des variations en fonction de la charge du réseau. Cette variation est appelée gigue. Cette gigue doit être bornée et garantie dans les SCC afin de respecter des exigences temporelles sur la communication. Ce délai de transfert se compose de :

- un délai d'attente au niveau émetteur : le traitement au niveau du système d'exploitation à l'interface réseau et de la file d'attente.
- un délai de propagation dans le réseau (dépend de la nature du canal de transmission).
- un délai de traitement au niveau de récepteur (file d'attente et traitement par le noyau).

— La ponctualité : "avant l'heure ce n'est pas l'heure, après l'heure ce n'est plus l'heure". La communication dans un STRC doit garantir la délivrance d'un service (transmission d'un

message) en respectant une date d'échéance maximale. Si cette condition temporelle n'est pas garantie, la communication est considérée erronée.

— Le déterminisme : un réseau déterministe correspond à borner le délai de transfert d'une part et de s'assurer qu'un évènement entraine le même effet pour des conditions données de l'autre part.

Dans les STRC, les réseaux représentent une ressource critique. Contrairement aux réseaux traditionnels qui s'intéressent à maximiser les bandes passantes et optimiser les délais de diffusion, la communication temps réel est fondée sur le respect de la propriété de ponctualité. Une communication industrielle peut être caractérisée par les propriétés suivantes :

— Topologie : c'est la spécification spatiale d'un réseau de communication, on peut distinguer les topologies suivantes : le bus, l'anneau, l'étoile, le maillage, etc. Comme on peut trouver des topologies hybrides.

— Mode de transmission : deux modes à distinguer :

  - Réseau à diffusion (broadcast ou multicast) qui repose sur une organisation spatiale en bus, en anneau ou en étoile. Toutes les stations se connectent à un seul canal de communication (partage). Un message émis est propagé à l'ensemble du réseau avec un mécanisme d'adressage qui permet de distinguer entre les stations et identifier le destinataire du message.
  - Réseau point à point qui est fondé sur une connexion directe entre stations (canal n'est partagé que par deux stations).

  La répartition spatiale des équipements industriels (capteurs, automates, moteurs, etc.) favorise le mode de transmission réseau à diffusion pour deux raisons principales :

  - Minimiser le risque d'incohérence des données (nombre de messages échangés réduit par rapport au mode point à point).
  - Réduire la gigue et par conséquent limiter l'impact sur l'aspect temps réel du système et ne pas causer des retards suite au réseau.

— Débit : c'est la capacité d'un réseau à transférer des données sur un intervalle de temps.

— Longueur d'un message : c'est la taille ou la quantité en bit/octet de données à transmettre dans le réseau. Cette longueur est variable en fonction du réseau et du protocole utilisé. En fonction de cette longueur et de la configuration du réseau (architecture et protocole utilisés), le concepteur peut estimer les unités de temps nécessaires pour la transmission et fixer la période de transmission des données.

— Criticité des messages : les messages échangés dans un réseau peuvent être de deux niveaux de criticité en fonction de contraintes temporelles appliquées au système de communication. En effet, il y a les messages avec des contraintes temps réel dur où le non respect de contraintes temporelles cause une défaillance critique dans le système. Ce qui impose une garantie temporelle pour la communication afin d'éviter des conséquences lourdes (catastrophe suite au dysfonctionnement) qui peuvent toucher le système entier. La deuxième catégorie de messages, correspond aux messages avec des contraintes souples ou sans contraintes. Le système en général tolère la perte de ce type de message puisque, ils ne sont pas qualifiés critiques, et donc la perte n'engendre pas des conséquences dangereuses.

— Messages périodiques ou sporadiques : les messages périodiques correspondent à la transmission des données selon une période fixée au début de chaque communication (intervalle de temps constant qui correspond à une période d'envoi). Exemple : Il s'agit de transmettre les données des capteurs (surveiller l'état des portes dans le système de contrôle

d'ascenseur, surveiller l'état de la chaîne de sécurité du système GESA, etc.). Par contre les messages sporadiques correspondent à une communication apériodique employée suite à une production d'un événement selon un intervalle de temps stochastique. Exemple : modification de l'architecture d'un réseau suite à un ajout/défaillance d'un nœud (capteur) dans le réseau.

Afin de répondre aux exigences temporelles imposées sur la communication dans les STRC, on trouve des mécanismes qui sont appelés protocoles d'accès au médium (en anglais "Medium Access Control (MAC)"). Ces protocoles contrôlent et arbitrent l'accès au réseau. Le canal de communication représente le seul moyen d'interaction entre différentes stations du réseau. Afin d'éviter la perturbation du signal causée par une collision de messages, une seule transmission est autorisée sur le canal. En effet ce sont les protocoles MAC qui gèrent cette autorisation. Les protocoles MAC dans un système temps réel ont subi pas mal d'amélioration afin de s'adapter aux différentes technologies et aux besoins industriels depuis leur apparition dans les années 70. Selon [Kurose *et al.*, 1984], ces protocoles sont classés en deux catégories : Les protocoles par contention (contention based protocol) et les protocoles à accès contrôlé (controlled-access protocol) comme le montre la figure 2.3.

2.2.3.1 Les protocoles par contention

Ces protocoles essaient de résoudre le conflit de transmission simultanée en se basant sur les approches suivantes : l'approche probabiliste, l'approche temporelle, l'approche par priorité, l'approche par adresse.

— Approche probabiliste : c'est la catégorie la plus simple. En cas d'une interférence (collision), les protocoles probabilistes se basent sur la technique de retransmission. Cette retransmission est déclenchée après un délai d'attente aléatoire (principe de protocole ALOHA et CSMA). La différence entre ces deux familles de protocoles c'est que la première est incapable d'observer le réseau avant de procéder à l'envoi et est donc incapable d'éviter la collision avant la transmission (les messages sont immédiatement transmis dès qu'ils sont générés). Alors que la deuxième famille est caractérisée par sa capacité d'observer le canal avant la transmission afin d'éviter la collision.

2.2.3.1.1 CSMA/CD (Carrier Sens Multiple Access with Collision Detection) : (sans évitement de collision) il s'agit d'un protocole probabiliste ce qui le rend inapproprié pour un contexte temps réel. Ce protocole permet d'observer l'état du réseau avant

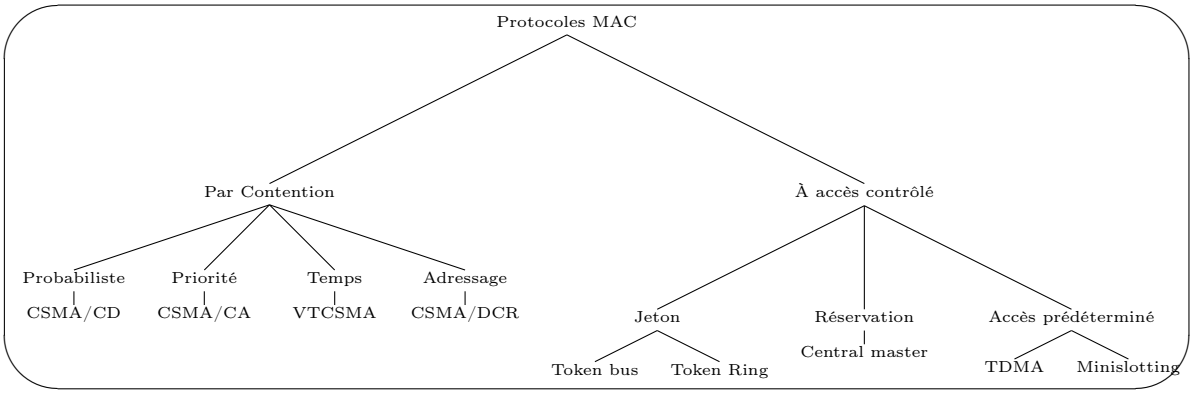


FIGURE 2.3 – Taxonomie des protocoles MAC

d'émettre un message pour vérifier sa disponibilité (liberté). Ce mécanisme s'appelle Carrier Sense, il n'empêche pas l'accès simultanés au bus ce qui cause des collisions de transmission. Ce protocole, après observation de la collision, intervient avec un mécanisme de contention pour remédier à ce conflit. Le principe est d'attribuer à chaque nœud un délai d'attente aléatoire dans un intervalle de temps borné (qui serait doublé en cas d'une deuxième tentative) avant de procéder à un nouvel accès au canal.

— Approche par priorité : se base sur l'attribution de priorité à chaque message ou chaque générateur de message. Lors d'un conflit, c'est l'unité la plus prioritaire qui retente une communication sur le médium.

2.2.3.1.2 CSMA/CA (CSMA with Collision avoidance) : (avec évitement de collision) le principe se base sur l'attribution des priorités aux messages afin d'autoriser ou non l'accès au medium. Avant de procéder à la transmission, chaque nœud utilise ses informations concernant l'état du réseau pour pouvoir calculer la probabilité d'entrer en collision à un instant donné et éviter les heures de pointe. Ce mécanisme est plutôt utilisé dans les communications de type « event triggered ». Lors d'un événement (génération d'un message), le processeur émetteur (qui possède l'accès au medium) détecte ce déclenchement suite à un changement de priorité des messages. Une fois que le bus est libéré, le processeur (générateur du message de plus haut priorité) tentera l'émission à son tour. Ce protocole supprime l'occurrence de collision.

2.2.3.1.3 CSMA/CR (Collision Resolution) : (avec résolution du conflit) Ce principe est particulièrement introduit dans les réseaux de type "Master-less", c'est à dire les réseaux sans gestion maître-esclave. L'idée est de permettre à tout nœud du réseau, d'envoyer son message sous la seule restriction : le bus doit être libre au moment de l'émission, s'il n'est pas libre, le nœud qui veut transmettre son message doit attendre que le bus se libère.

Au cas où deux nœuds voient le réseau libre, ils commencent à émettre leurs messages en même temps au point qu'un conflit de transmission se crée. Pour résoudre cette collision, l'idée est d'attribuer une priorité (poids) aux différents nœuds. La réalisation de cet arbitrage se fait grâce à un champ de bits dont la taille dépend de la norme du protocole. En cas de conflit, le nœud qui le détecte doit comparer ces bits un par un jusqu'à ce qu'un nœud gagne et reste le seul à utiliser le bus. Dans ce cas, les nœuds qui ont perdu l'arbitrage restent à l'écoute du bus, et cessent d'émettre. Ils ne réitèrent leur demande de transmission que lorsque le bus est de nouveau libre, comme l'illustre la figure 2.4.

Exemple : Pour l'automobile, en cas de collision, le déclenchement de l'airbag passera avant la gestion de l'injection du moteur.

Le réseau CAN (voir section 2.3), réseau très répandu dans le domaine de l'automobile intègre cette technique pour pouvoir transmettre les trames sur le medium correctement sans collision. En effet, cette technique se base sur le principe de ne transmettre aucun message tant que le support n'est pas libre. Afin d'éviter les collisions en chaîne, le processeur actif (émetteur) ne cesse d'émettre qui s'il détecte une différence de valeur entre son bit émis et le bit sur le support, i.e. un nœud qui émet un bit 1, s'arrête s'il observe sur le support un bit à 0. Sachant qu'une trame commence par une adresse unique et que ces adresses diffèrent au moins par un bit.

— Approche temporelle : c'est le concept de fenêtre de transmission. Le message ne peut être

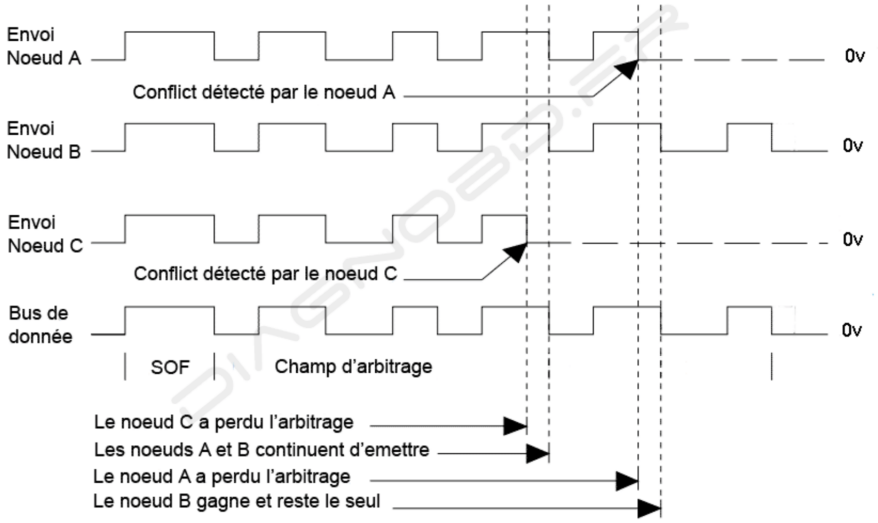


FIGURE 2.4 – Exemple d'arbitrage de type CR (Collision Resolution) pour le CSMA

transmis qu'à partir du moment où il est initié dans la fenêtre appropriée (Virtual Time CSMA [Molle and Kleinrock, 1985]).

2.2.3.1.4 VTCSMA (Virtual Time CSMA) : ce protocole se base sur la fenêtre de génération (il s'agit d'une horloge virtuelle locale). Après avoir marqué le silence sur le support, les nœuds du réseau initialisent une fenêtre avec la même valeur (qui est un paramètre du protocole). S'il y a une collision, cette fenêtre serait partagée en deux (procédure répétée jusqu'à détection d'un support libre).[Kurose *et al.*, 1984]

— Approche par adressage : est basée sur le principe de la dichotomie qui s'applique sur un arbre binaire. Cet arbre contient les équipements du réseau comme nœud. Lors d'une collision détectée, on parcourt la moitié de l'arbre afin d'autoriser la transmission au dernier demandeur d'accès de la partie sélectionnée.

2.2.3.1.5 CSMA/DCR (CSMA with Deterministic Collision Resolution) : cette méthode intègre un algorithme basé sur l'utilisation des arbres binaires. En absence de collisions, le protocole CSMA/DCR n'est pas différent de CSMA/CD. Par contre, sur occurrence d'une collision, le protocole procède différemment à CSMA/CD. Quant à CSMA/DCR, les processeurs se partagent en deux groupes selon leurs adresses (principe de dichotomie). L'objectif est d'identifier un sous arbre qui ne contient qu'un seul nœud actif. En effet, si le sous arbre sélectionné au début contient plus d'un nœud actif, il y a de fortes chances qu'une nouvelle collision soit détectée. Dans ce cas, le groupe se divise en 2 jusqu'à atteindre une seule feuille (processeur) qui peut accéder au medium pour transmettre tranquillement sa trame.

2.2.3.2 Les protocoles à accès contrôlé

Ces protocoles offrent un accès au médium libre de collision en se basant sur les approches suivantes : approche d'accès par jeton, approche d'accès par réservation et approche d'accès prédéterminé.

— Approche d'accès par jeton : se base sur la circulation d'un message "jeton" entre différents équipements et seul celui qui possède le jeton peut accéder au médium et transmettre son message.

2.2.3.2.1 Token ring et Token bus : cette technique est basée sur la circulation d'un jeton dans le réseau. Un jeton est une trame qui a un format spécial qui circule entre les nœuds selon l'ordre de l'anneau. Seule la machine qui est en possession du jeton peut émettre son message et occuper le canal pour un intervalle de temps donné. La nomination "token ring" (anneau physique) ou "token bus" (anneau logique) dépend de la topologie du réseau sinon le principe est le même.

— Approche d'accès prédéterminé : consiste à attribuer les droits d'accès aux stations de manière statique et prédéterminée. Le protocole le plus connu est le Time Division Multiple Access (TDMA). Il sert à partager le canal entre les différentes stations sous forme d'intervalles de temps appelés : time slot. L'accès au canal se fait d'une manière périodique dans l'intervalle de temps prédéterminé. Ces accès composent une séquence périodique appelée : cycle. Le problème rencontré avec cette méthode est la perte d'intervalles non occupés. Cet aspect sera pris en compte avec l'approche suivante.

— Approche d'accès par réservation : alterne la réservation avec la transmission afin d'éviter les pertes d'intervalles non occupés provoquées par le TDMA. En effet, à chaque intervalle de réservation un équipement explique sa volonté de transmettre des données selon un ordre défini.

2.2.3.2.3 Central master : Il s'agit d'un protocole centralisé. Les nœuds sont regroupés en deux catégories : un maître et des esclaves. Un esclave n'a le droit d'émettre que s'il est désigné par le maître en lui accordant un délai temporel. Cet esclave émetteur se comporte comme producteur tandis que les autres se positionnent en tant que consommateur. Cette méthode est largement déployée dans les réseaux de capteurs intelligents. En effet, les installation industrielles automatisées utilisent principalement cette méthode pour gérer la transmission (le réseau World Fip utilise cette technique).

2.2.4 Critères de sûreté de fonctionnement d'un réseau industriel

Après avoir fondamentalement bouleversé le domaine du contrôle-commande, l'émergence vers les SCR a amené le concepteur de machines à se poser de nouvelles questions vis-à-vis de la sécurité des ces systèmes : "Peut-on véhiculer des informations relatives à la sécurité à travers les réseaux de terrain ? Quel niveau de confiance peut-on accorder à ces systèmes de communication ? Doit-on recourir à des systèmes adaptés et/ou spécialisés pour la transmission d'informations relatives à la sécurité ?" Le concept général d'un réseau industriel dédié à la SdF est le suivant :

— côté émetteur : le message relatif à la sécurité est géré par la couche de safety avant d'être transmis selon le protocole du réseau standard,

— côté récepteur : le message est prélevé sur le médium du réseau standard puis reconnu, vérifié et validé (à travers les fonctions de sécurité qu'il contient) par la couche safety du réseau avant d'être transmis à l'application.

La communication safe utilise seulement les couches 1,2 et 7 du modèle ISO/OSI. Afin d'éviter tous changements dans ces couches, les mesures de sécurité sont implémentées dans une sur-couche "safety layer" qui se positionne au dessus de la couche 7. Cette couche de sécurité assure la transmission des données safe. Par conséquent, les mesures de sécurité peuvent être intégrées dans cette couche [Ciame *et al.*, 2009].

Cette couche safety est représentée par l'utilisation d'un logiciel développé spécifiquement pour garantir les exigences de SdF (niveau de SIL) appropriées à l'applicatif et qui concerne le système de communication. Cette couche de sécurité s'appuie sur la couche standard du réseau implémenté, non dédiée à la SdF.

Grâce à la technique décentralisée d'accès au medium, CAN garantit un temps de réaction court (exigence de réactivité pour les systèmes temps réel définie dans 2.2.2.1 à la page 25). CAN n'est pas capable d'assurer un fort débit pour des installations largement distribuées (sur de grandes distances). CAN peut être particulièrement adapté aux systèmes temps-réel localisés à intelligence distribuée et aux contraintes de fiabilité élevées.

Dans les installations temps réel trop critique qui exigent un niveau élevé de SdF, un réseau de type CAN peut mener à des problèmes de sécurité à cause de son non déterminisme (exigence de déterminisme définie précédemment dans 2.2.2.2, page 27). Par ailleurs, on ne pourra pas confier l'émission des messages de sécurité à un réseau de type CAN, il faudra naturellement donner une priorité parmi les plus élevées à ces messages, d'où la nécessité d'utiliser des protocoles déterministes.

2.3 Étude de l'existant de la communication dans les ascenseurs de Schindler, Sprinte, Sodimas

Le bus CAN a été créé par Bosh et soutenu par INTEL pour renforcer le niveau de fiabilité avec un faible coût dans le domaine de l'automobile. Il est standardisé par la norme ISO-11898.

Après son utilisation par plusieurs constructeurs (Mercedes, etc.), et après être adopté comme standard de bus de terrain par plusieurs associations des constructeurs, l'association CAN in Automation (CiA) se présente comme l'organisme fédérant les applicateurs de CAN. Cette association d'industriels est composée de sociétés et d'institutions dans le monde de l'automatisme. Elle a conçu au départ une spécification ouverte de la couche application CAN Application (CAL). Cette spécification fournit un groupe de services pour la gestion, la répartition et l'échange de données entre les nœuds. Un sous ensemble de CANOpen permet de produire des familles de profils de communication orientées métier performants pour des application industrielles temps-réel.

CAN est selon son objectif initial, fortement implanté dans l'industrie automobile pour réduire les faisceaux de câbles dans les véhicules. Il est venu résoudre le problème d'incapacité spatiale et économique pour l'échange des données des équipements électroniques embarqués toujours plus nombreux. Il n'est pas adapté à la transmission de données sur de grandes distances avec un fort débit. Celui-ci, est particulièrement adapté aux systèmes temps-réel localisé, à intelligence distribuée et aux contraintes de fiabilité élevées. Le débit d'un réseau CAN avec la topologie en bus (topolgie du bus CAN) pour une longueur maximale indicative de 40 m est égal à 1 Mbits/s (sur une paire torsadée) . Des longueurs de medium supérieurs (jusquà 1000 m) nécessitent une réduction de débit (passer à 50 Kbit/s).

Le bus CAN est de type producteur consommateur, comme détaillé précédemment (2.2.1 page 24). Il s'agit d'un réseau multi-maître de type contention (figure 2.3) (toute station maître peut initier une trame dès que le bus est libre) autorisant la production et la consommation

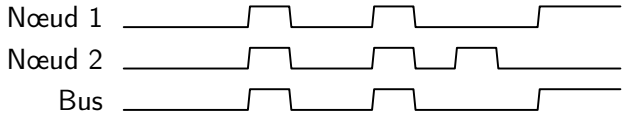
de l'information transmise. Il intègre la technique d'accès à inhibition de collision CSMA/CR (définie précédemment 2.2.3.1.3 page 30).

Toute anomalie détectée pendant l'arbitrage conduit le nœud émetteur à arrêter tout de suite sa transmission. Pour le CAN, les informations sont pondérées par des priorités différentes (un champ identifiant codé sur 11 bits (dans la version basique) se trouvant à l'entête de chaque trame explique cette priorité). L'arbitrage ne concerne que cet identifiant, i.e. chaque émetteur observe le bus et bascule en mode récepteur dès qu'il détecte un bit dominant (bit à 0) alors qu'il envoie un bit récessif (bit à 1). Chaque processeur a perdu l'arbitrage, tente à nouveau d'une manière automatique l'accès au medium (non destructive). De ce fait, la longueur du réseau (le temps de propagation) impacte le débit de transmission. En effet, CAN exige que le temps maximal de propagation des signaux soit très petit par rapport à la durée de transmission d'un bit (bit-time)[Ciame *et al.*, 1999].

Exemple :

— Nœud 1 : 0 01 00 10 00 11 gagne l'arbitrage

— Nœud 2 : 0 01 00 10 10 00 tente à nouveau après la transmission du maître 1.



Avec cette technique, la trame avec la priorité la plus élevée est retardée dans le pire des cas jusqu'à ce que la trame en cours de transmission (sur le medium) soit transmise en entier. Par contre, la trame de faible priorité risque de ne pas être transmise en cas de forte charge du réseau par des informations plus prioritaires. En effet, en utilisant la technique CSMA/CR, CAN gère la transmission des informations en fonction des priorités indépendamment de la charge du bus.

2.3.1 TTCAN

Le Time-Triggered Communication on CAN (TTCAN) introduit la notion d'une communication TT appliquée sur le protocole CAN. C'est le principe de sur-couche logicielle sur le CAN (i.e. au dessus de la couche supérieure). Le TTCAN est basé sur deux notions du temps : le temps local et le temps de cycle [Chabrol, 2006].

2.3.2 CANOpen

Le réseau CANOpen est basé sur le standard CAN pour la couche 1 et 2 du modèle Open Systems Interconnection (OSI). Le protocole CANOpen spécifie particulièrement la couche application en se basant sur le CAL. Ce protocole est défini sur les couches : physique, liaison de données et application (1,2 et 7) du modèle OSI. Depuis son apparition en 1995 suite au projet européen Esprit, le CANOpen a été normalisé dans EN 50325. Différents profils CANOpen réalisés par logiciel existent et concernent différents applicatifs : camions, trains, équipement médical, maritime, ascenseurs, etc. La couche application du protocole CANOpen supporte des canaux synchrones et asynchrones partagés. Le cycle de transmission synchrone est défini par la transmission cyclique d'une trame de synchronisation prioritaire (importance de la définition par l'utilisateur du champ de priorité). Il n'est pas possible de calculer un temps de réaction maxi, car CAN n'est pas déterministe sauf éventuellement pour la donnée ayant la priorité la plus haute. Ce qui pose un problème en SdF. En effet, le temps de réaction sur un réseau CAN dépend de la priorité de la trame. Ce temps peut être borné pour un message ayant la plus haute priorité, et ne peut pas l'être pour les autres messages. Le protocole CANOpen spécifie les

mécanismes de synchronisation au niveau de la couche application par la définition d'une trame de synchronisation hautement prioritaire d'une manière périodique. Cette technique permet de calculer le temps de cycle du réseau [Ciame *et al.*, 2009, Ciame *et al.*, 1999].

2.3.3 CANOpen-Lift

Dans les architectures des ascenseurs, les capteurs, le système de contrôle d'ascenseur et les actionneurs sont reliés par le bus CAN pour assurer le fonctionnement de la chaîne de sécurité. CANOpen fournit un framework qui contient la déclaration de différents composants du système d'ascenseur. Ce genre de frameworks donne au concepteur un accès complet aux variables du système de contrôle d'ascenseur. En effet, CiA ajuste et raffine le dictionnaire CANopen afin d'avoir un nouveau CANOpen approprié appelé CANOpen-Lift.

Cette spécification du CANOpen décrit l'utilisation du CANOpen dans un réseau de contrôle d'ascenseur. elle spécifie les interfaces de communication et les fonctionnalités de cette application au sein d'un système de contrôle d'ascenseur. Les spécifications applicatives du profil CANOpen décrivent les équipements virtuels (12 Virtual devices (VD)) du système d'ascenseur. Les contrôleurs virtuels (call controller, car door controller, car drive controller) exécutent les fonctions de contrôle dédiées à l'application. Dans cette application toutes les fonctions de contrôle peuvent être implémentées dans un seul appareil CANOpen (CANOpen device). Bien que dans les autres applications, les fonctions de contrôle sont implémentées obligatoirement dans différents équipements CANOpen. Ces unités virtuelles peuvent être mises chacune seule dans un équipement CANOpen ou combinées dans un ou plusieurs équipements CANOpen-Lift. Ce qui permet d'avoir une application à la fois simple et sophistiquée.

Fonctionnalités des contrôleurs (partie 2 Cia 417) : Le call-controller collecte toutes les données process temps-réel provenant des unités d'entrée (Input-panel-units) et les envoie dans un seul message vers les unités de sorties (Output-panel-units). Le car-door-controller reçoit les informations d'état des portes de toutes les car-door-units, et envoie une commande de porte dans un seul message vers toutes les car-door-units. Il est capable de collecter toutes les données process provenant de light-barrier-units, et les informations de la position de la cabine venant de car-position-unit.

Le car-drive-controller envoie la commande du moteur "drive commande" à l'unité correspondante "car-drive-units" et reçoit l'état du moteur de la cabine. Il reçoit aussi la position de la cabine et la mesure de charge de la cabine à partir de car-position-unit et load-measuring-unit.

Chaque équipement CANOpen conforme à la spécification de profil d'application doit partager les entrées du dictionnaire d'objet (de 6000h au 9FFFh : index des objets). Ces entrées sont communes pour tous les appareils et chaque dispositif met en œuvre les objets appropriés à ses fonctions et son rôle. On peut trouver des entrées du dictionnaire prédéfinies et réservées au CANOpen, comme on peut trouver des entrées, spécifiques à l'application (exemple application de contrôle d'ascenseur).

2.4 Convergence vers IP pour des solutions génériques

CAN est le bus de terrain utilisé actuellement par le système de contrôle d'ascenseur. Bien que sa technologie réponde intrinsèquement aux exigences temporelles et de disponibilité liées au système de contrôle d'ascenseur, ses performances ne sont plus développées depuis plusieurs années. Par conséquent, les performances n'ont pas suivi l'augmentation des besoins en réactivité de ses applications. De son côté, les performances du support Ethernet sont constamment

améliorées. À tel point qu'il est déjà devenu une alternative et est perçu comme le successeur des bus de terrain classiques pour le contrôle-commande de processus industriels. La section suivante va donc traiter des solutions industrielles à base d'Ethernet. L'utilisation de l'Ethernet dans les SCC va permettre de délivrer les nouveaux services : téléphonie, vidéo-surveillance, panneaux d'affichage dynamique.

2.4.1 Ethernet temps-réel

Aujourd'hui, Ethernet est devenu de plus en plus répandu avec des coûts de plus en plus faibles. Ethernet permet de connecter tout type de dispositif grâce à la rapidité et la facilité de son installation et son interopérabilité. En outre, Ethernet pourrait devenir une solution idéale pour l'automatisation. Néanmoins, il est connu qu'Ethernet classique n'est pas adopté pour les réseaux industriels à cause de la technique non déterministe d'accès au medium Carrier Sens Multiple Access with Collision Detection (CSMA/CD). Ce qui a poussé les chercheurs et les constructeurs industriels à réfléchir, à intégrer l'Ethernet dans la communication industrielle et avoir des réseaux à base d'Ethernet au lieu de bus de terrain classique. Pour atteindre cet objectif, l'IEC 61784 a mis en place un certain nombre de critères de performance pour les solutions de communication industrielle à base d'Ethernet. Les protocoles de communication, ainsi que les bus de terrain doivent répondre aux contraintes de communication industrielle (détaillé dans la section 2.2.1 page 22) : Le réseau doit être :

— robuste à l'environnement industriel (Couche physique),

— déterministe (garantir le rafraichissement des données dans le temps de cycle) (Couche Liaison de données),

— interopérable (pouvoir échanger des informations entre tous les types d'équipement industriel.) (couche Application).

2.4.2 Ethernet industriel

Le réseau Ethernet trouve son origine au début des années 1970 par le centre de recherche de Palo Alto (Parc) anciennement Xerox Palo Alto Research Center (Xerox) (`http://www.parc.com`). À la base, l'idée était d'exploiter la méthode d'accès CSMA/CD définie dans 2.2.3.1.1 en s'inspirant du réseau Aloha, mis en service à l'époque pour pouvoir transmettre des données par radio entre les îles de Hawaï. Le but était de réaliser un réseau qui relie les ordinateurs du campus de l'université de Hawaï. Ce premier réseau a été la première brique qui construisait le réseau Ethernet qui a été présenté avec les caractéristiques suivantes :

— débit de 2,94 Mb/s,

— capacité de connecter plus de 100 stations,

— distance maximale entre deux nœuds de 1 kilomètre.

Et depuis son apparition, Xerox et ses associés tel que Intel Corporation (intel) et Digital Equipment Corporation (DEC) ne cessent d'améliorer les performances du réseau Ethernet de base pour promouvoir Ethernet en tant que standard. Ethernet est aujourd'hui le protocole le plus répandu auprès du grand public. De plus, fort d'une disponibilité variée (nombreux fournisseurs), d'une flexibilité d'utilisation (support et topologie variée), de débit élevé (de 10 Mb/s à 10Gb/s) et d'une robustesse éprouvée, il est de plus en plus intégré dans des systèmes industriels conçus pour des applications critiques. Il se distingue par son protocole d'accès et la nature de son support. Les bus de terrain présentés dans le chapitre 2 sont considérés comme des solutions limitées par rapport aux performances offertes par l'Ethernet :

— Nombre important des nœuds terminaux.
— Topologie basique du réseau.
— Débit des solutions temps réel.
— Bande passante des non temsp réel.
— Précision du temps de synchronisation.

Avec le progrès technologique, les exigences liées aux SCC ont évolué avec le temps. En effet, le réseau de communication industrielle doit être capable de supporter un débit très élevé avec une grande quantité de données (téléphonie, voix sur IP, vidéo-surveillance). Les bus de terrain présentés dans la section 2.3 (i.e. CAN), ne sont pas conçus à l'origine pour répondre à ces exigences, ce qui les rend insuffisants pour les nouveaux SCC. Ethernet et les protocoles associés (IP, UDP, TCP) sont largement utilisés avec des faibles coûts. En outre Ethernet est disponible avec de très hauts débits de transmission (i.e. 1 Gb/s). Selon [Decotignie, 2005] l'Ethernet industriel doit répondre aux exigences suivantes :

— Capacité temps réel : l'Ethernet doit garantir un comportement temps réel (exigé par des applications industrielles) (délai de synchronisation entre les différents équipements, délai de transmission, gigue, bande passante).
— Capacité de supporter la sécurité fonctionnelle : l'Ethernet doit permettre de protéger les données contre les différentes défaillances qui peuvent toucher la communication afin de répondre aux niveau SIL de l'IEC 61508. Autrement, l'Ethernet doit garantir un échange efficace et fréquent des données.
— Capacité de déterminisme : l'Ethernet doit garantir un temps de communication déterministe et par conséquent remplacer le CSMA/CD, la principale cause du non déterminisme (cf. chapitre 2)

Considérant le modèle OSI classique d'un réseau de communication, la garantie temps réel résulte d'un effort réparti sur différentes couches.

— Niveau couche application : c'est le modèle d'interaction qui joue un rôle très important pour la garantie temps réel, i.e. le modèle client-serveur n'est plus adapté à un système temps réel, parce que l'aspect temps réel de ce modèle ne dépend pas que de la communication, il dépend aussi du comportement du serveur. Si pour une requête, le serveur ne répond pas au client, cette contrainte de temps réel ne sera plus respectée pourtant la communication est temps-réel. Pour cela, d'autres modèles d'interaction sont adaptés aux systèmes industriels (i.e. producteur-consommateur, publish-subscribe).
— Niveau couche liaison de données : Le protocole MAC joue un rôle très important à ce niveau pour garantir à la fois le temps réel et le déterminisme. Pour améliorer le comportement temporel de l'Ethernet, il existe trois approches possibles : en supprimant/évitant les collisions, en réduisant le nombre de collisions et en corrigeant les collisions.

Ethernet contient les spécifications de couches 1 et 2 (physique et liaison de données) du modèle de référence OSI. Selon [Alves *et al.*, 2000] plusieurs méthodes ont été proposées afin de garantir un aspect déterministe à la communication basée sur Ethernet. La solution la plus simple pour borner le temps d'accès est la technique TDMA.

2.5 Les approches Ethernet temps réel

L'Ethernet standardisé par l'IEEE dans la norme 802.3 et à l'international par la norme IEC8802-3, est un support développé à l'origine pour les applications bureautiques. Il couvre les

couches 1 et 2 du modèle OSI (Figure 2.5), c'est-à-dire la façon dont les données sont transmises électriquement sur le médium physique et les procédures de transmission de données entre nœuds. Depuis ces dernières années, l'Ethernet est devenu de plus en plus répandu dans l'automatisation.

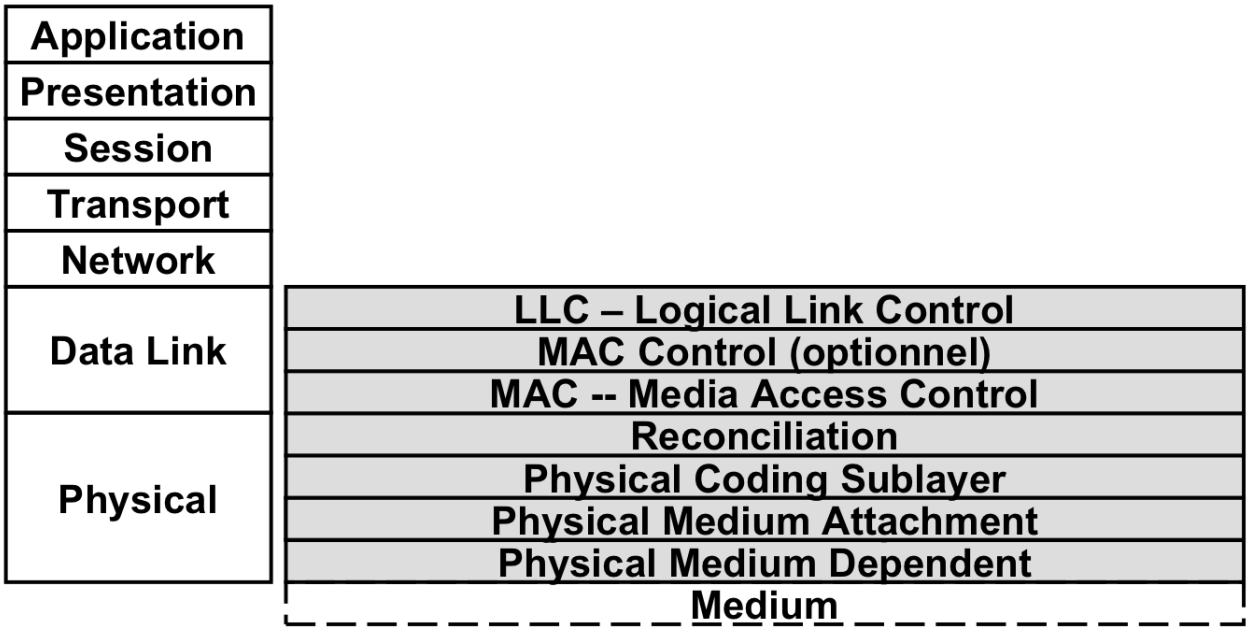


FIGURE 2.5 – Position d'Ethernet par rapport au modèle OSI.

Son utilisation a déjà dépassé le cadre de la bureautique pour certaines applications industrielles en remplacement des bus de terrain traditionnels tels que CAN, WorldFIP ou Profibus [Limal, 2009]. En effet, l'alternative Ethernet propose des caractéristiques attrayantes :

— une augmentation constante des débits possibles. À l'heure actuelle des couplages jusque 1Gbit/s sont proposés,

— une taille de trame possible jusqu'à 1526 octets (entête incluse),

— les coûts de développement sont supportés et amortis par les acteurs de la bureautique. Il en résulte une diminution du coût du support physique et éventuellement du couplage.

— par conséquent, l'offre est régulièrement enrichie par de nombreuses solutions basées sur du matériel standard, avec des possibilités de connexion simples ou durcies. Ces dernières sont développées afin d'être utilisées dans des environnements difficiles (par exemple en température ou humidité).

Ayant été supportés principalement par la bureautique, les choix technologiques qui ont conduit au développement d'Ethernet (et en particulier le développement du débit théorique possible) ne sont néanmoins pas toujours adaptés aux applications industrielles. En premier lieu, Ethernet n'est plus un bus. Des équipements d'interconnexion (commutateurs, concentrateurs) sont nécessaires. Ces équipements nécessitent une alimentation qui n'est pas fournie par Ethernet et sont susceptibles de tomber en panne. En plus du surcoût par rapport à un simple câble, les équipements d'interconnexion se révèlent donc être des éléments critiques de la disponibilité du réseau. Ensuite, comme cela a été expliqué par [Decotignie, 2005], Ethernet n'est pas déterministe. La méthode d'accès au médium CSMA/CD (cf. Section 2.2.3) est une méthode multi-maître qui

ne garantit ni le délai d'acheminement d'une trame sur le réseau, ni l'acheminement effectif de la trame. Dans [Thomesse, 2005], l'auteur explique que des propositions ont été faites très tôt pour bénéficier d'un Ethernet déterministe, mais qu'elles ne sont économiquement pas viables aujourd'hui. Par contre, la généralisation des commutateurs dérivés de la norme IEEE 802.3D a permis d'utiliser les mécanismes du Full-Duplex et ainsi de donner plus de déterminisme à l'acheminement des trames. Alors qu'un concentrateur répète toutes les trames reçues sur ses autres ports en Half-Duplex, c'est-à-dire que le signal ne peut circuler que dans un seul sens dans un lien, sous peine de collision, le Full-Duplex permet à un concentrateur d'émettre et de recevoir sur un port en même temps. Les éventuelles concurrences d'arrivée de trames seront résolues dans le commutateur par des mécanismes de files d'attente. Ainsi, le délai d'acheminement des trames peut être borné. À ce titre, notons que [Song *et al.*, 2002] propose des modélisations du comportement des différentes technologies de concentrateur. C'est pourquoi l'Ethernet commuté est présenté comme une solution de "réseau temps-réel et déterministe" [Jasperneite *et al.*, 2002] et des travaux tels que [Kalappa *et al.*, 2006], [Georges *et al.*, 2006] ont permis d'évaluer les contraintes temporelles et les conditions matérielles associées pour lesquelles Ethernet commuté peut être considéré comme réactif. Le commutateur filtre les messages selon les adresses destination et source pour les commuter sur le(s) lien(s) adéquat(s). La table de routage, donnant la relation destination-canal, est assimilée par un processus d'auto-apprentissage. De plus, les connectiques disposant de circuits séparés de transmission et de réception, la technique de détection de collision est désactivée et les communications sont réalisées en mode full-duplex : transmission et réception simultanée. Quand à savoir si tous les ports peuvent dialoguer simultanément sans perte de messages, cela dépend de la qualité du commutateur (non blocking switch). Au final, il en résulte que les échanges peuvent s'effectuer à débit normal et sans collision, et donc avec une bande passante sensiblement mieux occupée que dans un réseau partagé. Toutefois, les commutateurs (utilisant la technologie Store-And-Forward, seule technologie disponible aujourd'hui) introduisent :

— des délais pour l'acheminement des trames qui peuvent se révéler non négligeables en fonction de la contrainte temporelle de réactivité voulue. En effet, alors qu'il faut 0,56μs à un bit d'une trame Ethernet pour parcourir 100m de câble cuivre (0,5 μ s pour une fibre optique) et 1μs pour traverser un concentrateur, il lui faut de 12μs (pour une trame Ethernet de 64 octets) à 125μs (pour une trame Ethernet de 1526 Octets) en Fast-Ethernet (100Mb/s) pour traverser un commutateur Store-And-Forward (d'après les chiffres fournis par certains constructeurs d'équipements comme Hirschmann).

— une gigue (variation autour du délai moyen) importante qui résulte de la politique de mise en file d'attente implémentée.

— un risque de congestion d'un équipement pouvant nécessiter une contention de certains flux. Si les commutateurs industriels actuels sont dimensionnés pour supporter une charge importante, les couplages des nœuds terminaux peuvent être pris en défaut.

Par conséquent, s'il élargit l'espace des conditions de réactivité possible, Ethernet commuté n'apporte pas un gain significatif en terme de contrainte de réactivité par rapport aux bus de terrain traditionnels.

2.5.1 Classification

Introduire l'Ethernet temps réel dans l'industrie paraît une conséquence logique de l'introduction d'Ethernet dans "industrial automation". Les bénéfices de promouvoir Ethernet comme un standard industriel sont :

— La réduction des coûts d'installation du réseau.

— La réduction des problèmes d'interopérabilité.

Les chercheurs ne cessent de proposer des solutions pour les spécifications d'Ethernet, afin de répondre aux critères "temps-réel". Certains proposent des solutions pour la qualité de service, pour la synchronisation entre les dispositifs ou la modification du traitement de paquet [Felser and Sauter, 2004]. En se basant sur le temps de réponse de chaque solution, nous pouvons classer ces approches à base d'Ethernet en trois classes comme le montre la figure 2.6.

Le temps de réponse est le temps nécessaire pour transmettre un Protocol Data Unit (PDU) d'un nœud à un autre. Le temps de réponse est mesuré au niveau de la couche application.

— Classe 1 : pour le contrôle humain avec un temps de réponse environ 100ms. Ce temps est typique pour les interventions humaines dans l'observation des processus. Cette classe est garantie par les canaux classiques TCP/IP, c'est pour cela que l'on l'appelle cette solution une "Solution RTE au-dessus de TCP/IP".

— Classe 2 : pour le contrôle de processus avec un temps de réponse inférieur à 10ms. Cette contrainte est imposée par la plupart des outils de contrôle, comme les PLC, PC-based control. Pour atteindre ce comportement temporel, des mesures spéciales doivent être prises au niveau des équipements : 'utilisation d'équipements spéciaux, performants paraît indispensable afin de simplifier la pile protocolaire TCP/IP dans la phase temps réel et de garantir ainsi le temps de réponse nécessaire.

— Classe 3 : pour motion control avec un temps de réponse inférieur à 1 ms. La gigue ne doit pas dépasser 1 micro seconde, cette classe ne sera plus atteignable avec l'Ethernet classique (100 Mbps) que si le protocole d'accès au medium et la structure de matériels sont modifiés.

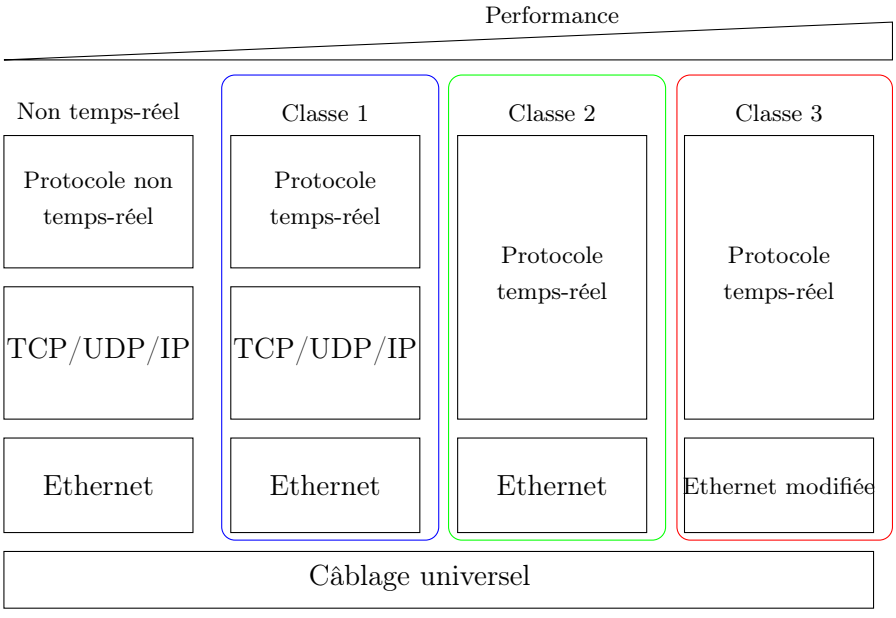


FIGURE 2.6 – Classification des solution RTE.

Cette classification se base à la fois sur les performances et sur les fonctionnalités ajoutées à Ethernet. La plupart des protocoles font face au problème de la couche MAC. Ils ajoutent une

sous couche de priorité/ordonnancement dans laquelle des mécanismes de type TDMA ou maître /esclave sont utilisés (voir figure 2.3, 29). Avec ces méthodes, le réseau transmet seulement un message à la fois, pour supprimer tout problème de collision.

Différentes solutions ont été proposées afin de faire face à ce problème. Chaque fournisseur fait sa propre promotion avec pour but la standardisation de son produit. Ces protocoles respectent la classification des différentes solutions RTE. Nous avons fait une étude détaillée des protocoles en nous basant sur les travaux [Prytz, 2008, Jasperneite *et al.*, 2007] qui portent sur la comparaison entre deux protocoles de la classe 3, l'EtherCAT de Beckhoff (type 12 dans le standard IEC 61158) et le Profinet IRT (Isochronous Real-Time) de Siemens (type 10 dans le standard IEC 61158).

2.5.2 EtherCAT

L'EtherCAT est un protocole de classe 3 qui modifie la couche Ethernet afin d'atteindre un temps de réponse inférieur à 1 ms. En théorie, EtherCAT utilise des cartes Ethernet standard. Cependant en pratique, un matériel spécifique (FPGA, ASIC) est utilisé pour réduire le temps de traversée de la trame. La topologie utilisée dans l'industrie est la topologie linéaire.

Les datagrammes EtherCAT sont encapsulés dans une seule trame Ethernet. Donc une trame Ethernet est composée d'une entête et de l'ensemble des datagrammes EtherCAT. Un datagramme est défini pour chaque esclave et contient le type de la commande ainsi que les données associées. La taille du datagramme est prédéfinie et elle est fixe pour tout esclave. Le nombre des datagrammes encapsulés dans une trame Ethernet classique est limité, vue la limitation de la taille d'une trame Ethernet comme le montre la figure 2.7.

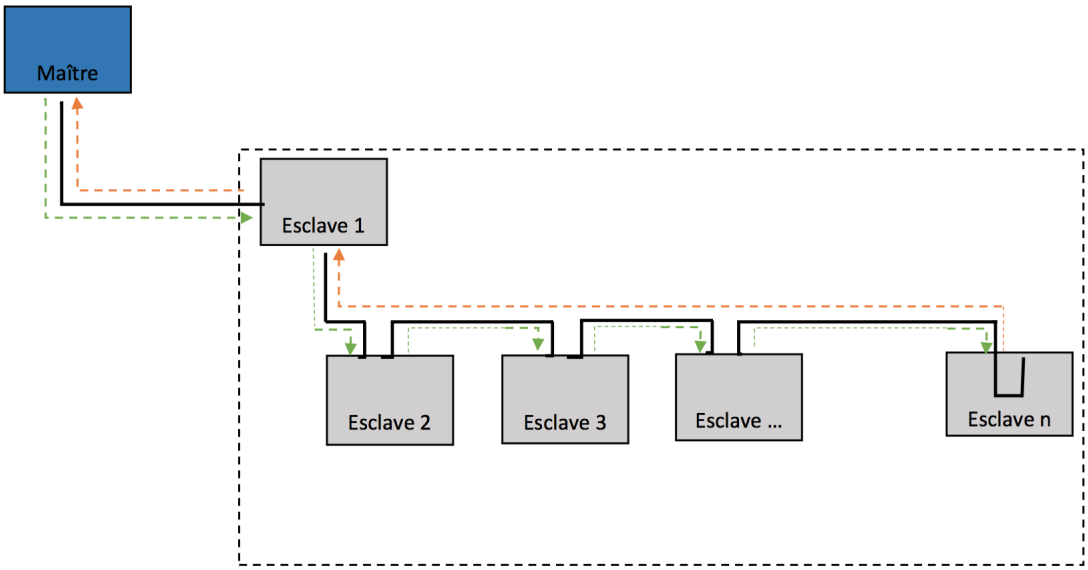


FIGURE 2.7 – Segment d'EtherCAT

On peut exprimer le nombre maximum d'esclave dans un seul segment EtherCAT en fonction de la taille de chaque datagramme come dans l'équation 2.1.

$$N_{max} = \frac{Taille\ Ethernet - Entete\ Ethercat}{Taille\ donne\ utile} \tag{2.1}$$

Le principe d'one-frame-many-slave est appliqué dans le segment EtherCAT. La trame Ethernet circule et passe par tous les esclaves du segment. Chaque esclave retransmet la trame afin de la faire passer à l'esclave suivant. Au cours de cette procédure un temps de latence s'ajoute et s'accumule, en plus du temps de propagation de la trame sur le support physique. Pour réduire cette latence, EtherCAT a recours au matériel spécifique afin de réduire le délai de forwarding de la trame dans le segment. Nous calculons le temps de cycle pour l'EtherCAT comme dans l'équation 2.2

$$T_{cycle\ pour\ EtherCAT} = (2n-1) \times latence + 2n \times delai\ de\ propagation$$
$$Avec\ n\ = nombre\ d\ esclave\ dans\ le\ segment \quad (2.2)$$

2.5.3 Ethernet PowerLink (EPL)

Le problème principal des protocoles Ethernet TCP/IP est le non déterminisme causé par l'utilisation de mécanisme CSMA/CD.

Pour une transmission déterministe de données, le protocole EPL améliore la couche liaison de données avec des mécanismes de scrutation et de découpage temporel pour permettre une transmission déterministe de données.

L'EPL est un protocole de communication temps réel déterministe de classe 2 (temps de réponse inférieur à 10 ms) purement logiciel et indépendant des fournisseurs se basant sur le standard Ethernet. Il est régi par Ethernet PowerLink Standardization Group (EPSG).

L'EPL intègre tous les mécanismes de CANOpen pour la couche applicative comme le montre la figure 2.8. L'utilisation des switches n'est pas recommandée dans les réseaux EPL puisque ces derniers augmentent les gigues, sachant que la gigue ne doit pas dépasser 1 μs afin de respecter les performances de la classe à laquelle il appartient.

L'EPL distingue deux domaines différents :

— le domaine temps réel pour la transmission de données critiques et

— un domaine non temps réel pour l'envoi de données non critiques.

Par manque de mécanisme de séparation du trafic, EPL nécessite dès le début une séparation du trafic sous forme de réseaux locaux ; un réseau pour le trafic temps réel, et un réseau pour le trafic non temps réel.

Le protocole EPL définit deux types de nœuds :

— un nœud qui gère la synchronisation du réseau ainsi que l'accès au médium des différents équipements connectés à ce réseau qui est le Managing Node (MN),

— tous les autres nœuds sont des Controlled Node (CN).

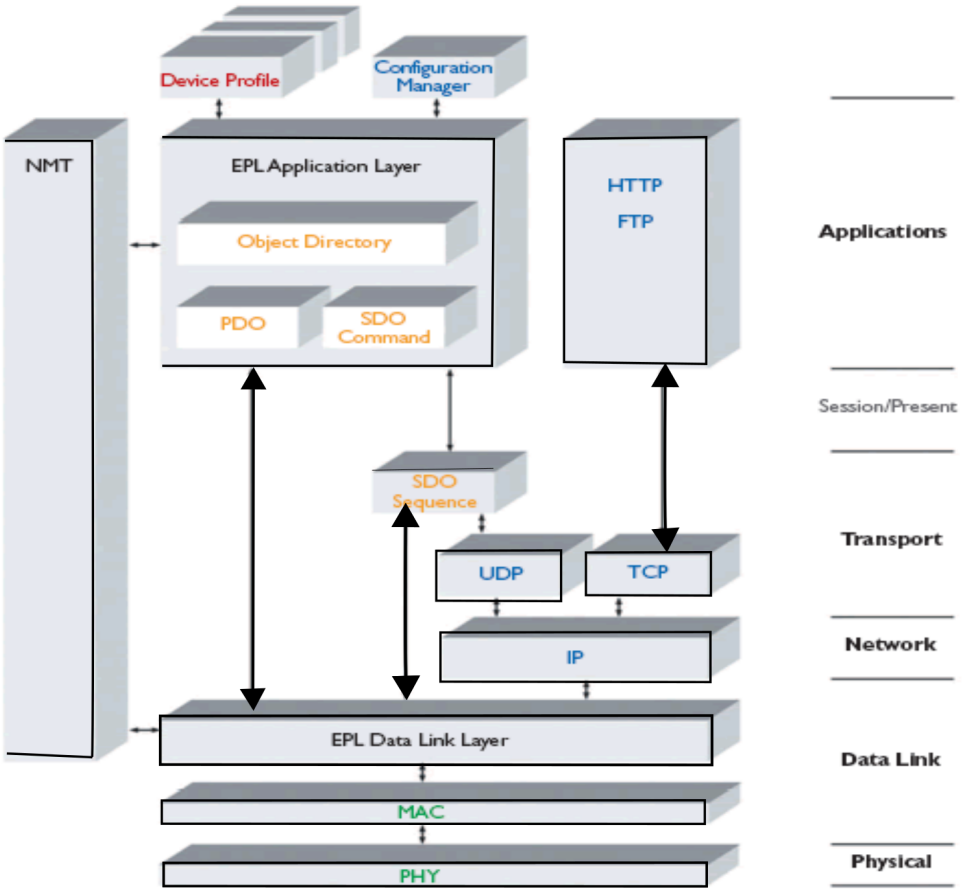


FIGURE 2.8 – Positionnement du Powerlink par rapport au modèle OSI et intégration des mécanismes du CANOpen.

Pour achever le déterminisme, EPL intègre le mécanisme de Polling avec la technique TDMA qui donne à chaque nœud l'accès exclusif au réseau pour une durée bien déterminée afin d'éviter tout type de collision. La figure 2.9 détaille la technique de division de temps avec EPL.

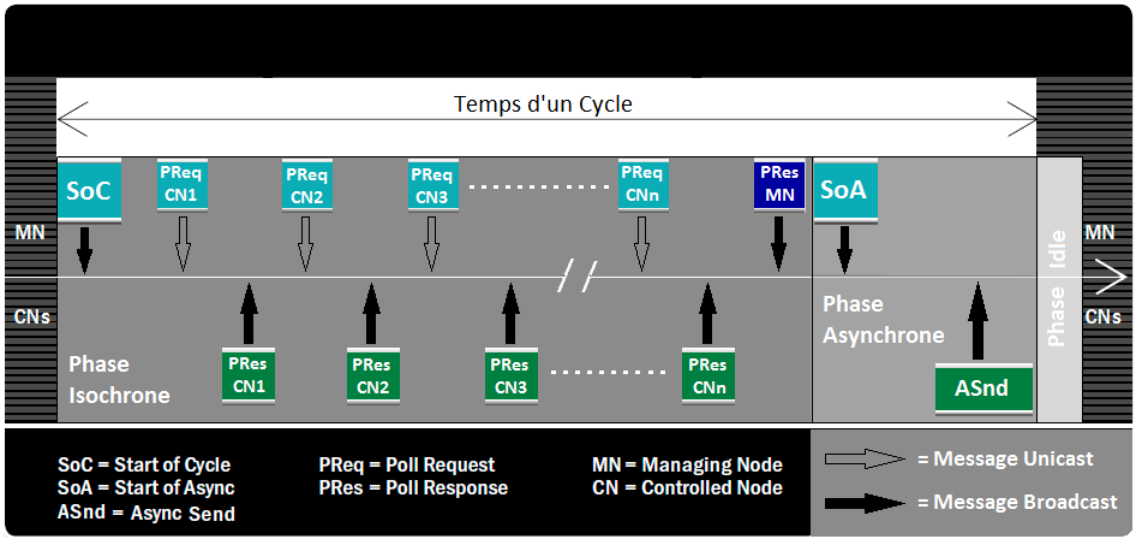


FIGURE 2.9 – Différents messages échangés lors d'un cycle EPL.

L'EPL opère d'une façon cyclique pour assurer la communication entre les différents nœuds du réseau. Il existe trois principales phases dans un cycle EPL comme le montre la figure 2.9.

— la phase isochrone,

— la phase asynchrone,

— et la phase idle.

2.5.3.1 Phase isochrone

Cette phase est dédiée pour le transfert de données critiques, l'annonce du début de cette phase se fait par un message « Start of Cycle » (SoC) envoyé en broadcast depuis le MN vers tous les CNs du réseau. Selon le mécanisme de Polling, le MN envoie à chaque CN du réseau un message « Poll Request » (PReq) spécifique en Unicast auquel ce dernier doit répondre, dans un délai bien déterminé (configurable), par un message « Poll Response » (PRes) en broadcast. L'accès au réseau est exclusif pour chaque nœud émetteur pour éviter les collisions. La fin de cette phase est annoncée par un message PRes envoyé par le MN en broadcast vers tous les nœuds CNs. Pendant cette phase le protocole TCP/IP n'est pas utilisé vu qu'elle exige des performances temps réel.

2.5.3.2 Phase asynchrone

Pendant cette phase il y a un échange de données non critiques, cette phase est généralement dédiée à la configuration et au paramétrage du réseau. Le début de cette phase est annoncée par un message « Start of Asynchronous » (SoA) envoyé en broadcast dans le réseau, pendant cette phase n'importe quel nœud peut envoyer un message « Asynchronous Send » (ASnd) contenant des informations non critiques généralement liées à la configuration du réseau ou bien la collecte d'informations sur les nœuds.

2.5.3.3 Phase idle

C'est une phase de repos qui séparent deux cycles consécutifs et pendant laquelle il n'y a pas échange de données [Doyle, 2004].

2.5.4 Ethernet\IP

L'Ethernet\Industrial Protocol (EIP) est un protocole ouvert de la couche application développé par ControlNet International (CI), Open DeviceNet Vendors Association (ODVA), Industrial Ethernet Association (IEA). Il est fondé sur les couches physique et liaisons de données de l'IEEE 802.3 et les couches TCP/UDP/IP. EIP utilise les objets définis dans le protocole Control and Information Protocol (CIP) comme le montre la figure 2.10.

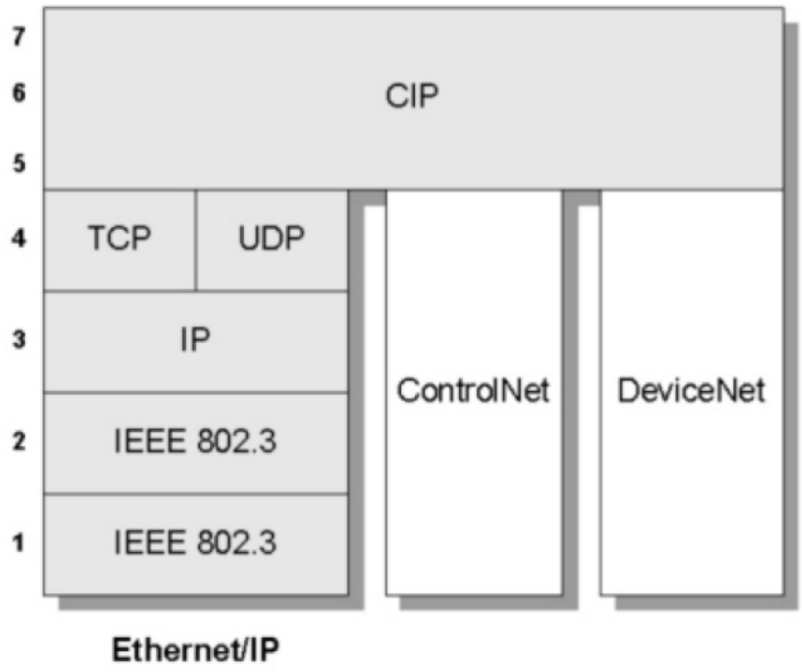


FIGURE 2.10 – La pile Ethernet\IP

2.5.5 Time-Trgiggered Ethernet (TT Ethernet)

Aujourd'hui on trouve Ethernet dans diverses applications qui demandent une haute criticité temporelle, par exemple :

— Dans les applications industrielles : EtherCAT, EPL, EIP, ProfiNet,

— dans les applications aérospatiales : ARINC, ASCB/D, Avionics Full DupleX switched Ethernet (AFDX)..

— dans d'autres application : AVB, DCB.

Toutes ces variantes d'Ethernet sont apparues pour garantir/améliorer certaines exigences/demandes du systèmes industriels en QoS, comme le déterminisme, la capacité de garantir un échange "safe" des données, la possibilité de fournir un seul support pour faire circuler les données "safe" et "non safe".

Parmi ces variantes, nous trouvons le Time-Trgiggered Ethernet (TT Ethernet), qui intègre le paradigme de communication time-triggered dans l'Ethernet. TT Ethernet est un protocole réseau déterministe, distingué par :

— une synchronisation distribuée tolérante aux fautes,

— un partitionnement temporel robuste de la bande passante et

— une communication synchrone avec une latence fixe et une très faible gigue.

La technologie TTEthernet de la compagnie TTTech (`http://www.tttech.com`) a été standardisée depuis 2011 par la Society of Automotive Engineers (SAE) sous SAE AS6802 pour son utilisation en automobile, aérospatiale, automatisation industrielle, etc.

TT Ethernet permet de supporter des flux de données de différentes criticités temporelles su sein du même réseau, c'est-à-dire la coexistence des applications critiques de contrôle-commande, des programmes de divertissement et d'autres applications standards de réseaux locaux, non critiques sur le même réseau bien au contraire des autres variantes d'Ethernet temps réel. En effet, TT Ethernet impose la classification des données en trois services de communication ayant des exigences temporelles différents :

— La classe du trafic TT est celui de plus haute criticité : faible gigue, latence limitée grâce au maintien d'une synchronisation globale entre différents nœuds du réseau et le respect d'un plan de communication globale partagé entre les différents nœuds du réseau afin de gérer/éviter les conflits entre les messages TT. Ce plan planifie les évènements concernant l'envoi et la réception de messages TT. Ce trafic est utilisé quand des faibles latences et gigues sont requises. Avec ce trafic, le routage employé est le mode statique qui rend le trafic complètement déterministe, en revanche cela nécessite une configuration hors ligne du réseau. L'adressage utilisé est orienté contenu, c'est-à-dire qu'en plus de l'adresse de destination, l'émetteur doit spécifier le contenu de la trame (la même valeur pour chaque message dans un cycle de synchronisation)

— La classe du trafic moyen ou avec une criticité moyenne (Rate Constrained (RC)). Cette classe garantit une bande passante de bout en bout avec un trafic asynchrone contrairement au trafic TT. Sa latence est beaucoup plus élevée que la latence du trafic TT. Cela est dû au temps d'attente des messages RC dans les files d'attentes. Par contre le temps de transmission reste toujours borné. Cette classe est basée sur la norme ARINC 664. Cette norme exige une réservation au préalable d'une bande passante suffisante pour les messages RC ce qui limite les délais et les déviations temporelles de ces messages. Cette classe est utilisée pour du trafic moins exigeant du point de vue déterminisme et temps-réel

— La classe du trafic le moins critique (le BE D'Ethernet) utilise le reste de la bande passante sans garantie d'envoi ou de réception. Il correspond au trafic d'Ethernet classique.

2.6 Conclusion

Ce chapitre a permis de décrire les exigences d'un SCC (i.e. déterminisme, réactivité, disponibilité). Pour répondre à ces exigences, nous avons identifié les propriétés du réseau industriel adapté. Par ailleurs, une partie a été consacrée à la description des caractéristiques d'une communication dans les SCC, à savoir les systèmes temps réel critique. Cela pourra justifier la description des protocoles d'accès au medium déployés dans ce type de systèmes. Dans la troisième partie de ce chapitre nous avons fait une étude de l'existant de la communication utilisée par les trois PME françaises spécialistes dans la construction des ascenseurs : Shindler, Sprinte et Sodimas.

À la fin de ce chapitre, et pour présenter les résultats de nos travaux, nous avons introduit la convergence de la communication industrielle vers IP pour des solutions génériques évidemment l'introduction de l'Ethernet temps réel dans les SCC en général, et dans les ascenseurs en particuliers. Dans cette partie nous avons présenté une classification des solutions RTE. Dans le chapitre suivant, nous allons nous servir de cette classification pour présenter, notre choix de protocoles avec une analyse d'évaluation de leurs performances temporelles et le développement d'une couche safety au-dessus de la couche application du protocole considéré.

Chapitre 3

Performances temporelles des solutions à base d'Ethernet temps réel

Sommaire

3.1 Introduction

L'Ethernet industriel doit conserver entièrement la capacité d'Ethernet classique. En effet, son utilisation pour la communication industrielle est basée sur l'utilisation de la couche physique standard du modèle OSI, qui permet de gagner en terme de prix du matériel et offrant ainsi un vaste choix d'équipement réseau supportant divers protocoles (i.e. HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Network Time Protocol (NTP), etc.). Dans ce chapitre, nous allons comparer les performances temporelles de deux solutions d'Ethernet industriel basées sur deux protocoles candidats dans le cadre de ce projet. Cette comparaison nécessite un contexte d'application commun. Ainsi, la spécification du scénario de communication est basée sur un seul contrôleur (i.e. Programmable logic controller (PLC), Personal computer (PC)), des capteurs et des actionneurs interconnectés entre eux via un réseau Ethernet. Du point de vue réseau, le contrôleur du système représente le maître qui gère les communications, celui-ci initie tous les dialogues avec les nœuds esclaves (i.e. capteurs et actionneurs).

3.2 Définition d'un temps de cycle minimum

Les applications de contrôle industriel sont généralement basées sur une ou plusieurs unités de contrôle. Ces contrôleurs définissent une période appelée "cycle de contrôle". En effet, le contrôleur est caractérisé par un temps de cycle dans son algorithme de contrôle. Ce temps de cycle de contrôle peut être divisé en trois parties comme le montre la figure 3.1 :

— Temps de récupération des données du capteur qui représente le délai d'acquisition d'un signal d'entrée (refresh time en anglais),

— Temps de traitement algorithmique dans l'unité de contrôle,

— Temps de mise à jour de l'actionneur qui représente le délai de mise à jour d'un signal de sortie (envoi des commandes aux actionneurs).

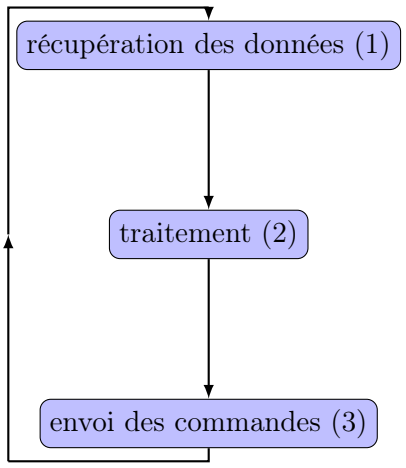


FIGURE 3.1 – Temps de cycle de contrôle

Les étapes (1) et (3) représentent respectivement les délais de transmission au contrôleur et le délai de transmission à un équipement de sortie. Ces délais décrivent les temps de communication dans un système de contrôle en réseau. Cette durée devrait être inférieure à la différence entre le temps de cycle de contrôle et le temps du traitement [Robert *et al.*, 2012]. Par conséquent, ce temps de communication est une mesure de performance important pour les systèmes de communication industrielle. Aujourd'hui, beaucoup de constructeurs de systèmes industriels préférent utiliser l'Ethernet industriel comme support de communication au vu des avantages qu'il présente par rapport au bus de terrain classique et de ses caractéristiques attrayantes (section 2.3, i.e. le bus CAN) :

— rapidité : Ethernet permet d'atteindre de hauts débits de transmission de données (de 10 Mbps à 1 Gbps),

— capacité : Ethernet offre une taille de trame qui peut atteindre 1526 octets,

— flexibilité : le bus Ethernet permet d'échanger, en plus des données nécessaires pour le contrôle industriel, tous types de données numériques comme l'indique le standard IEEE 802.3 (i.e. audio, vidéo, etc.),

— réduction du coût : Ethernet couvre les couches 1 et 2 du modèle OSI ce qui permet d'unifier les interfaces physiques qui assurent la transmission électrique des données sur le support (diminution du coût des interfaces et éventuellement du couplage),

— interopérabilité : la norme Ethernet permet de gagner en interopérabilité offrant la possibilité d'intégrer les applications industrielles basées sur du matériel standard.

Les solutions Ethernet temps réel sont décomposées en 3 classes : Selon [Jasperneite *et al.*, 2007] la classe 1 des solutions Ethernet temps réel est basée sur le standard natif d'Ethernet sans aucune modification (i.e. EIP, Modbus/TCP). La classe 2 est la solution qui utilise l'ordonnancement

basé sur les priorités définies dans IEEE 802.1D. La classe 3 offre une solution basée sur Ethernet qui intègre des nouveaux mécanismes d'ordonnancement dans des équipements physiques complémentaires (i.e. Application-Specific Integrated Circuit (ASIC) ou Field-Programmable Gate Array (FPGA)). Cette classe permet l'élimination de toutes collisions et simplifie l'estimation du temps de transmission [Prytz, 2008]. L'évaluation des performances temporelles de chaque protocole Ethernet temps réel est basée sur une métrique dite "temps de cycle", définie de la façon suivante dans la littérature. Selon [Jasperneite *et al.*, 2007] et [Robert *et al.*, 2012], *"un temps de cycle est défini comme le temps nécessaire pour échanger les données d'entrée/sortie entre le contrôleur, qui représente le maître du réseau, et les autres équipements du système, qui représentent les esclaves du réseau"*.

3.3 Performances temporelles d'une communication à base d'Ethernet temps réel : EPL Vs EtherCAT

3.3.1 Modèle de communication à base d'Ethernet PowerLink

Le protocole EPL améliore la couche liaison de données de l'Ethernet standard qui utilise le mécanisme CSMA/CD avec les mécanismes TDMA qui donnent à chaque nœud esclave CN l'accès exclusif au réseau pour une durée bien déterminée. Ainsi EPL définit un maître MN qui est en charge d'attribuer l'accès au réseau afin d'éviter tout type de collision afin de permettre une transmission de données déterministe.

3.3.1.1 Couche liaison de données d'Ethernet PowerLink

Comme le montre la figure 2.9 du chapitre précédent et sans appliquer aucun changement, l'EPL utilise la couche MAC de l'Ethernet. Par conséquent, la technique utilisée par les stations EPL pour l'accès au médium de transmission, est la technique CSMA/CD. Les spécifications d'EPL définissent deux modes de fonctionnement différents :

— Le mode EPL : ce mode opère en temps réel et d'une façon cyclique pour assurer la communication entre les différents nœuds du réseau. Dans ce mode, c'est la couche DLL de l'EPL qui est mise en fonction. En effet, dans ce mode, c'est la station maître (MN) qui gère l'accès au médium de transmission. Ce type de nœud (i.e. MN) est unique dans un réseau EPL. Un esclave (CN) ne peut jamais accéder au médium sans avoir l'autorisation d'accès de son MN. Cette technique centralisée d'accès au médium permet au réseau EPL d'éviter les collisions. Par conséquent, les mécanismes de résolution de collision aléatoire définis par CSMA/CD, qui sont la cause de non déterminisme pour une transmission Ethernet, ne sont plus employés, ce qui donne une communication déterministe sûre. Dans ce mode opératoire d'EPL, la communication se met en place principalement à travers des trames EPL, cependant, il existe des slots de temps dédiés aux échanges asynchrones sans passer par les trames EPL. La couche la plus utilisée pour le trafic asynchrone est la couche User datagram protocol (UDP), mais tout autre protocole peut être utilisé pour ce trafic.

— le mode basique d'Ethernet : ce mode offre une communication qui se fait à travers les canaux Transmission control protocol (TCP)/UDP/IP. Ce mode est utilisé lorsque le temps réel n'est pas exigé pour le transfert des données. Quand un réseau EPL fonctionne en mode Ethernet basique, la couche Data link layer (DLL) classique du standard IEEE802.3 est utilisée. Par conséquent, le mécanisme CSMA/CD est employé comme technique d'accès au médium de transmission, sans aucune modification. Dans ce cas la communication devient,

comme dans les réseaux Ethernet en général, une communication non déterministe et toutes les couches du modèle OSI peuvent être utilisées (TCP, UDP, etc.).

3.3.1.2 La trame EPL

Comme le montre la figure 3.2, les trames EPL sont encapsulées et transmises dans des champs de données d'une trame Ethernet définie dans le standard IEEE 802.3. La taille d'une trame Ethernet doit varier entre 64 octets et 1526 octets. Cette trame contient 3 champs : une entête fixe de 22 octets (y compris la préambule et le délimiteur de début), 4 octets réservés au CRC et un champs de données qui ne doit pas dépasser 1500 octets.

Le champs "Ethernet type" dans un réseau EPL doit contenir la valeur hexadécimale $88AB_h$.

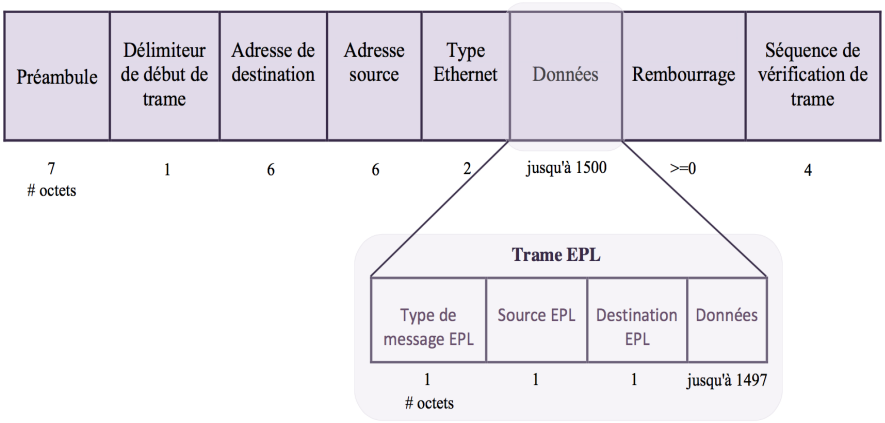


FIGURE 3.2 – Structure d'une trame EPL.

Une trame EPL contient quatre champs :

— message type (de 1 octet) : spécifie le type de la trame EPL (Start of Cyclic (SoC) ou PollRequest (PReq) ou PReq ou Start of Asynchronous (SoA) ou Asynchronous Send (ASnd), etc.)

— adresses source/destination (de 1 octet chacune)

— données échangées pour le reste.

3.3.1.3 Le cycle élémentaire d'un réseau EPL

Clairement dans les applications industrielles, nous nous sommes intéressés au mode EPL. Chaque cycle EPL est constitué de 4 différentes périodes comme le montre la figure 3.3.

Phase de démarrage : Synchronisation Lors de chaque démarrage d'un cycle EPL, le MN envoie en multicast d'Ethernet une trame de type "SoC" afin de synchroniser les CNs. La trame SoC est la seule trame périodique d'EPL, i.e c'est la seule trame générée et transmise périodiquement et indépendamment dans le réseau à chaque cycle. Toute autre trame EPL est déclenchée par un évènement (event-driven), i.e. les autres trames d'EPL sont générées et transmises dans le réseau pour répondre à un évènement particulier (i.e. réception d'une trame).

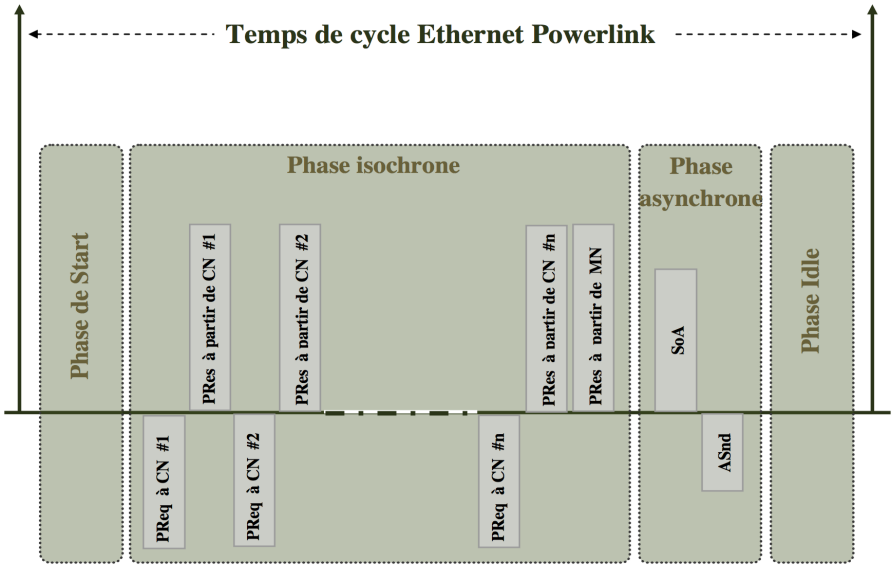


FIGURE 3.3 – Un cycle EPL.

Phase isochrone Cette période de la phase isochrone est dédiée aux échanges des données temps-réel via des slots de temps entre les stations d'EPL. En effet, après l'envoi de la trame SoC, l'échange cyclique des données temps réel se fait grâce à une interrogation séquentielle. Cette interrogation est assurée grâce aux trames "PReq" et "PollResponse (PRes)". Afin d'éviter les délais indésirables (potentiellement dangereux), l'EPL exige que l'interrogation de chaque CN doit être bornée par un intervalle de temps fixe. Si ce délai n'est pas respecté par un CN, le MN saute au CN suivant.

Phase asynchrone C'est une période dédiée aux échanges des données non temps réel entre les stations EPL. Une fois que la phase isochrone est achevée en envoyant une trame "SoA" à tous les CNs, le MN autorise le début de la phase asynchrone. En effet, ce maître envoie à un esclave, qui a manifesté sa volonté d'envoyer d'occuper la phase asynchrone, une trame SSoA afin de lui autoriser de transmettre toutes ses données non temps-réel dans une seule trame de type "ASnd.

Phase de repos Finalement la période de repos est un intervalle de temps entre la fin de la transmission asynchrone et le démarrage du cycle suivant. Durant cette période, toutes les stations EPL attendent simplement la trame SoC suivante.

Le temps de cycle EPL représente la période nécessaire pour la collection des données temps réel par les CNs et leurs transmissions dans chaque cycle au MN aussi bien que la transmission des données de contrôle temps réel envoyées par le MN aux CNs. En détail, la communication entre les stations EPL se produit en se basant sur un cycle. Le cycle EPL est géré par le MN et est répété d'une façon continue durant la phase de communication. La durée de ce temps de cycle est définie par l'utilisateur dans la phase de configuration off-line, qui reste constante tant que le réseau est mis en place, c'est à dire, le temps de cycle ne change pas durant la phase d'opération EPL. La durée du temps de cycle est un paramètre de configuration dans le nœud MN du réseau EPL. Désormais, nous notons le temps de cycle d'un réseau EPL par T_{cycle}. Ce paramètre représente la durée totale nécessaire d'une communication dans un réseau EPL, autrement dit, c'est la période durant laquelle la transmission des données entre le MN et les

CN doit être achevée. Il est important que ce T_{cycle} ne soit pas dépassé durant différentes phases d'opération du réseau. Le temps de cycle EPL est donné par l'expression 3.1 :

$$T_{cycle} = T_{sync} + T_{iso} + T_{asy} + T_{idle} \tag{3.1}$$

où T_{sync}, T_{iso}, T_{asy}, T_{idle} représentent la durée de la phase de démarrage, isochrone, asynchrone et du repos. La valeur T_{sync} est une valeur constante qui représente le temps nécessaire pour transmettre la trame SoC avec la marge nécessaire de safety qui garantit que toutes les stations sont synchronisées. Par conséquent, il est important de garder le temps de démarrage du cycle (T_{sync}) EPL précis et sans gigue (gigue réduite = $\leq 70s$).

3.3.1.4 Analyse de performance d'EPL

Afin d'analyser le comportement des configurations expérimentales et de configurer correctement le protocole de communication, nous avons fait une étude théorique du système de communication. L'objectif de cette analyse est de fournir une évaluation des performances qui peuvent être atteintes par le système. Pendant la phase de configuration du réseau EPL, l'utilisateur doit définir les valeurs de certains paramètres tels que le temps de cycle, le délai de réponse (timeout) aux messages de type "polling" et le délai d'une communication asynchrone. Afin de configurer correctement un réseau EPL, il est important de choisir soigneusement ces paramètres après une analyse de performance théorique. Dans cette étude théorique, nous prendrons en compte les temps nécessaires pour l'envoi d'une trame. Dans l'analyse théorique suivante, et selon [Knezic *et al.*, 2016], [Seno, 2011], [Seno *et al.*, 2009], [Cena *et al.*, 2009], [Seno and Vitturi, 2007], plusieurs paramètres réseau doivent être pris en considération pour assurer un comportement temps-réel du système. Ces paramètres caractérisent le système de communication mis en œuvre. Ils influencent le temps de cycle théorique dans un réseau EPL. Nous avons besoin d'étudier le pire cas des conditions idéales d'un scénario opérationnel sous les hypothèses suivantes :

— les trames Ethernet temps réel échangées de type SoC, PReq, PRes et SoA sont de taille minimum (i.e. 64 octets), sachant que cette taille est largement suffisante pour des applications de communication industrielle. En effet, la taille des données échangées est généralement faible (quelques octets pour représenter l'état d'un capteur, la commande d'un actionneur, un état de variable, etc.) et le protocole EPL n'ajoute que 3 octets. Néanmoins les trames asynchrones de type ASnd sont de taille 300 octets (exigence fonctionnelle imposée par les spécifications du protocole EPL),

— le réseau fonctionne sous condition d'absence totale d'erreurs de transmission, i.e. chaque trame transmise est correctement reçue par la/les destination/s pendant la phase d'analyse de performance.

Le cycle EPL est découpé en trois phases (i.e. isochrone, asynchrone et idle) comme le montre l'équation 3.1. Le T_{sync} décrit la période de démarrage (synchronisation des nœuds). Le T_{iso} représente la durée de la phase isochrone. Dans cette phase, nous trouvons la période d'interrogation de tous les CNs du réseau. La phase isochrone est limitée par les trames SoC et SoA. Le T_{asy} représente la durée de la phase asynchrone et dépend directement de la configuration du réseau (i.e. ajout/suppression d'un nœud, état d'un nœud, etc). Le T_{idle} représente la durée de la phase inactive (repos) du réseau. Dans notre analyse nous nous focaliserons sur la phase isochrone, en particulier, nous verrons l'impact de T_{iso} en fonction des autres paramètres pour énumérer toutes les configurations que nous pourrons mettre en place dans la partie expérimentation. Le diagramme de séquence illustré dans la figure 3.4 montre les messages échangés entre

les nœuds d'un réseau EPL composé d'un maître MN et de deux esclaves CN1 et CN2 dans un cycle EPL.

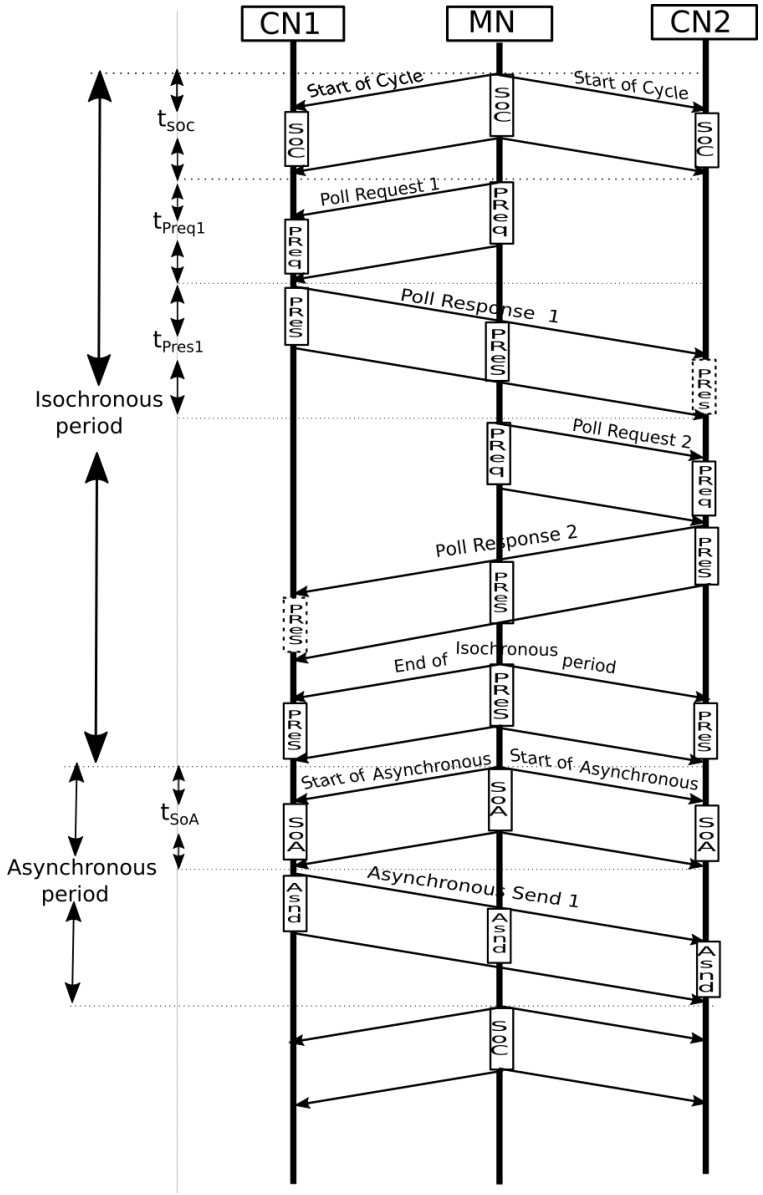


FIGURE 3.4 – Diagramme de séquence de la communication EPL avec un maître et deux esclaves.

3.3.1.4.1 La phase isochrone La phase isochrone est influencée par différents facteurs tels que le nombre M de CNs et la taille de données cycliques qui affectent le délai de transmission d'une trame. La durée de la phase isochrone est donnée par l'équation 3.2.

$$T_{iso} = T_{sync} + M \times T_{poll} + T_{PRes} \tag{3.2}$$

où T_{sync} est le temps de synchronisation et M est le nombre des CNs dans le réseau EPL comme le montre la figure 3.3. Les spécifications d'EPL mentionnent que 45 μs est une valeur

typique pour T_{sync}. T_{poll} est le temps d'interrogation de chaque CN. Durant la période de synchronisation, le MN informe tous les CNs du réseau qu'il est prêt à commencer un cycle de communication EPL en envoyant une trame SoC. C'est la seule trame générée d'une manière périodique dans un réseau EPL. Tandis que durant la période d'interrogation, le MN interroge séparément (one by one) chaque CN du réseau en lui envoyant une trame PReq qui nécessite une réponse de type PRes. La transmission de ces trames est déclenchée par la réception d'une autre trame, nous parlons d'event-driven frames. Pour commencer la phase de synchronisation, une trame de type SoC qui a un temps de transmission T_{SoC} est envoyée depuis le MN en multicast aux CNs du réseau pour les informer du début de cette phase. Selon ses spécifications, le protocole EPL exige un temps d'attente minimum avant l'envoi du prochain PReq. Une fois que tous les CNs ont reçu la trame SoC, la phase d'interrogation s'active et les CNs se mettent à l'attente des interrogations du MN. Une trame PReq, issue du MN à destination d'un CN particulier est transmise en unicast. La durée de cette opération est exprimée par T_{PReq}. Pour la phase de réponse, le CN envoie une trame PRes en multicast à travers le réseau. L'équation 3.2 est composée de T_{sync} et T_{poll}, qui sont détaillés dans l'équation 3.3 :

$$\begin{cases} T_{sync} &= T_{SoC} + T_W \\ &= 45\mu s \\ T_{poll} &= T_{PReq} + T_{PRes} \end{cases} \tag{3.3}$$

où T_{PReq}, T_{PRes} sont respectivement le temps de transmission des trames PReq et PRes. Dans le cas général, avec M CNs, l'expression 3.2 devient l'expression 3.4 :

$$\begin{aligned} T_{iso} &= T_{sync} + T_{PRes} + M \times (T_{PReq} + T_{PRes}) \\ &= T_{SoC} + T_W + T_{PRes} + M \times (T_{PReq} + T_{PRes}) \end{aligned} \tag{3.4}$$

3.3.1.4.2 La phase asynchrone Durant la période asynchrone, le MN détermine et identifie la station qui peut commencer la transmission asynchrone et diffuse cette information dans la trame SoA, sous une condition, pas plus d'un nœud est autorisé à envoyer des données asynchrones par cycle, sachant que la trame SoA marque le début de la communication asynchrone en échangeant des messages de type ASnd. En général, le nœud autorisé à envoyer ses données asynchrones peut les envoyer dans un format de trame Ethernet avec une taille inférieure ou égale à la taille de "maximum transmission unit (MTU)", qui précise la taille maximum d'une trame asynchrone d'EPL. La MTU des données asynchrones d'EPL varie entre 300 octets et 1500 octets. La valeur de T_{ASnd} dépend du besoin de la communication (i.e. la taille de données échangées varie de 300 octets à 1500 octets selon les spécifications d'EPL). 3.5.

$$T_{asy} = T_{SoA} + T_{ASnd} \tag{3.5}$$

3.3.2 Modèle de communication à base d'EtherCAT

EtherCAT, le deuxième protocole candidat dans le cadre de ce projet, a été développé par Beckhoff. Il implémente un protocole maître / esclave sur l'Ethernet. En principe, le maître EtherCAT et l'esclave peuvent être mis en œuvre à l'aide d'interfaces physique entièrement standards. Cependant, dans la pratique, un esclave EtherCAT est mis en œuvre avec un équipement spécial (i.e. FPGA, ASIC) pour faciliter les délais de transfert de paquets très courts dans les appareils esclaves (nous allons décrire cette partie en détail dans le chapitre suivant). Une trame

est envoyée par le maître vers les esclaves. Ces derniers peuvent lire et écrire des données à la volée (principe "on the fly"). Le protocole EtherCAT peut prendre en charge les topologies en ligne et en anneau. Vu que la topologie en ligne est utilisée principalement dans le cadre industriel, elle a été étudiée dans notre analyse. En effet, un anneau logique est défini entre les esclaves de sorte que lorsqu'une trame atteint le dernier esclave dans l'anneau elle est retournée au maître via tous les esclaves. Une seule trame traverse l'anneau en parcourant tous les esclaves. En outre, EtherCAT est capable de transporter tout type de trafic Ethernet standard de manière transparente (en théorie, il est également possible que le trafic EtherCAT et tout autre trafic Ethernet coexistent lorsque des esclaves matériels standards sont utilisés avec des dispositifs de commutation standard selon la norme IEEE 802.1D). Si la trame Ethernet est trop grande pour être transportée dans le temps disponible, dans un cadre EtherCAT, cette trame est fragmentée puis retournée à la réception. C'est une caractéristique importante qui ne limite pas le temps de cycle réalisable en raison du temps réservé uniquement au trafic asynchrone.

3.3.2.1 La trame EtherCAT

Le protocole EtherCAT utilise la trame Ethernet pour encapsuler ses datagrammes, comme illustré dans la figure 3.5. Généralement, un grand nombre d'esclaves peuvent être logés en utilisant un seul télégramme, ce qui optimise l'utilisation de la bande passante et réduit les taux d'interruption. Cette utilisation très efficace de la bande passante est activée par un concept appelé adressage logique. Un télégramme EtherCAT est encapsulé directement dans une trame Ethernet basique comme le montre la figure 3.5. Un télégramme EtherCAT est composée d'un entête spécifiant sa longueur ainsi qu'une liste de datagrammes dont le nombre dépend du nombre d'esclaves. Un datagramme est défini pour chaque esclave, et il contient le type de commande et les données associées. La taille totale de la trame EtherCAT peut être divisée en deux parties :

— une valeur constante (de 28 octets) égale à la somme d'entête du protocole Ethernet (26 octets), et l'en-tête de l'EtherCAT (2 octets),

— une valeur variable qui dépend du nombre d'esclaves M, de la quantité des données transportée et destinée à chaque esclave et de l'en-tête du télégramme (12 octets).

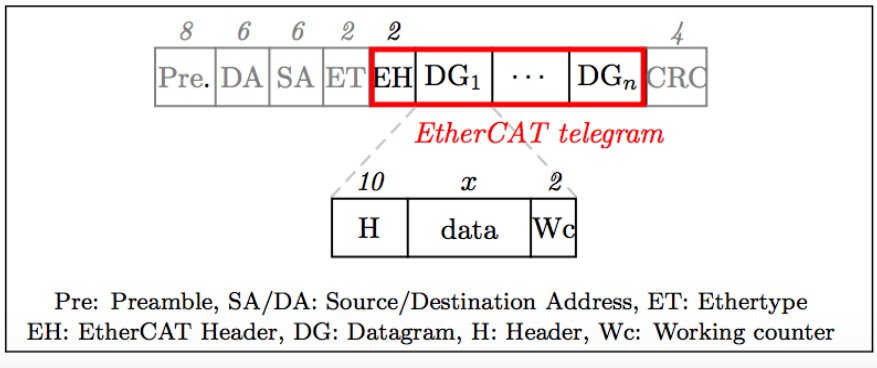


FIGURE 3.5 – Trame EtherCAT.

3.3.2.2 Le cycle d'un réseau EtherCAT

Le temps de cycle théorique résultant consiste en une somme pondérée des temps d'acheminement pour tous les télégrammes du maître et le délai de transfert lié à chaque esclave. Ce

temps de cycle EtherCAT peut également être défini comme la somme du délai de transmission de la trame maître et de la période d'inactivité du réseau comme illustré dans la figure 3.6.

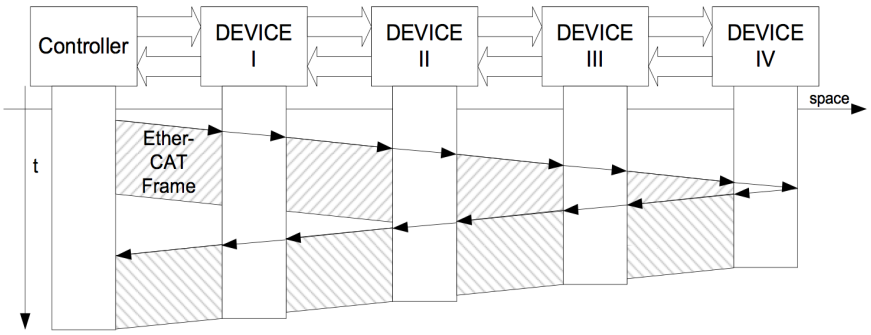


FIGURE 3.6 – Un cycle EtherCAT.

Dans cette étude, notre analyse de la performance d'EtherCAT a considéré un scénario de test hypothétique :

- la topologie physique du réseau est en ligne afin d'avoir une topologie logique de type anneau.
- l'étape d'initialisation est ignorée et seule la communication cyclique est étudiée,
- le maître envoie une seule image par cycle,
- la charge utile x est la même pour chaque esclave,
- la durée des opérations de lecture ou d'écriture correspond uniquement à la latence du périphérique réseau, qui est indépendante de la taille de la trame et est donc la même pour tous les esclaves,
- le délai de synchronisation d'EtherCAT est ignoré vu qu'il a été estimé à quelques nanosecondes.

En se basant sur la figure 3.6, l'équation 3.6 décrit le temps de cycle d'un réseau EtherCAT :

$$T_{cycle-EtherCAT} = (2n-1)T_{latence} + T_{transmission-EtherCAT} \tag{3.6}$$

La taille minimum d'une trame Ethernet contenant le télégramme EtherCAT ($Taille^{ETH}_{Ethernet}$) doit être supérieure ou égale à 64 octets (($Taille^{min}_{Ethernet}$) et inférieure ou égale à 1526 octets ($Taille^{max}_{Ethernet}$). Si la longueur du télégramme EtherCAT est inférieure à 38 octets, une quantité équivalent de remplissage est insérée dans la trame Ethernet pour assurer la taille minimale des données définie par le protocole Ethernet. Ce télégramme comprend déjà une en-tête de 2 octets, ce qui signifie qu'il n'y a aucune exigence de remplissage lorsque la longueur de la séquence de datagrammes est supérieure à 36 octets comme l'explique l'équation 3.7.

$$\begin{array}{lcccl} Taille^{min}_{Ethernet} & \leq & Taille^{ETH}_{Ethernet} & \leq & Taille^{max}_{Ethernet} \\ 64 & \leq & 28 + n \times (12 + x) & \leq & 1526 \end{array} \tag{3.7}$$

Le délai de transmission est le rapport entre la taille de la trame et la capacité de liaison (débit du réseau) comme le décrit l'équation 3.8.

$$\begin{aligned} T_{transmission-EtherCAT} &= Taille^{ETH}_{Ethernet} \times t_{tx} \\ &= (28 + max(36,\ n \cdot (12 + x))) \times t_{tx} \end{aligned} \tag{3.8}$$

où t_{tx}, comme décrit dans la section précédente, représente le temps nécessaire pour transmettre un octet dans un réseau en fonction de son débit (i.e. avec un débit de transmission = $100Mbps$, $t_{tx} = 0,08\ \mu s$). L'équation 3.6 devient :

$$\begin{aligned} T_{cycle-EtherCAT} &= (2n-1)T_{latence} + T_{transmission-EtherCAT} \\ &= (2n-1)T_{latence} + (28 + max(36,\ n \cdot (12 + x))) \times t_{tx} \end{aligned} \tag{3.9}$$

Sachant que dans l'hypothèse nous avons précisé que le maître ne peut transmettre qu'une seule trame Ethernet par cycle (afin d'éviter le découpage des trames qui cause la perte en performance). Il convient de noter que l'équation 3.6 considère une seule trame par cycle. Et vu que la taille de la charge utile Ethernet dépend directement du nombre d'esclaves et que la taille de la trame Ethernet ne peut pas dépasser 1 526 octets, l'équation 3.6 n'est valable que si le nombre des datagrammes (n) de longueur (x) qui peuvent êtres inclus dans une seule trame Ethernet est inférieur à un nombre maximum (n_{max}). Par conséquent, en nous basant sur l'équation 3.7, nous déduisons l'équation 3.10 (on suppose que $1 \leq x \leq 1486$) :

$$\begin{array}{lcll} Taille^{min}_{Ethernet} & \leq & Taille^{ETH}_{Ethernet} & \leq & Taille^{max}_{Ethernet} \\ 64 & \leq & 28 + n(12 + x) & \leq & 1526 \\ 36 & \leq & n(12 + x) & \leq & 1498 \\ & & n \leq \lfloor \frac{1498}{12+x} \rfloor & & \\ & & n_{max} = \lfloor \frac{1498}{12+x} \rfloor & & \end{array} \tag{3.10}$$

En général, le nombre d'esclaves sur un réseau EtherCAT peut être plus grand que n_{max} (supérieur à la capacité de la taille de la trame). Cela signifie que le contrôleur doit envoyer plus d'une trame dans un temps de cycle pour servir tous les esclaves. En fait, le nombre de trames Ethernet requis pour supporter n périphériques avec une charge utile constante x est donné par : $K = \lceil \frac{n}{n_{max}} \rceil$.

Par conséquent, l'équation 3.9 doit distinguer entre la dernière trame et les autres trames Ethernet déjà transmises dans le cycle avec un nombre de datagramme égal à n_{max} et donne au final :

$$\begin{aligned} T_{cycle-EtherCAT} &= (2n-1)T_{latence} + T_{transmission-EtherCAT} \\ &= (2n-1)T_{latence} + ((28k + (k-1)n_{max}(12 + x))) \times t_{tx} \\ &\quad + (max(36,\ (n - (k-1)n_{max}) \cdot (12 + x))) \times t_{tx} \end{aligned} \tag{3.11}$$

Le terme final de l'expression 3.11 a été utilisé pour différencier les cas où la dernière trame a généré un bourrage. Des résultats similaires sont donnés dans [Jasperneite *et al.*, 2007] et [Robert *et al.*, 2012] mais, contrairement à ce travail antérieur, l'expression 3.11 :

— prend en compte le mécanisme "on the fly" dans le cycle EtherCAT ; le principal avantage est l'aptitude d'un périphérique à commencer le transfert de la trame avant sa réception complète (contrairement, par exemple, au mode "store-and-froward"), ce qui réduit significativement le temps d'acheminement,

— considère l'utilisation des bits de bourrage, tel que définis par Ethernet,

— intègre le temps nécessaire pour transmettre les informations envoyées par les périphériques au contrôleur,

— considère les cas où le nombre d'esclaves et leur charge utile nécessitent l'utilisation de plusieurs trames.

3.3.3 Comparaison du temps de cycle pour les deux modèles : EPL et EtherCAT

Pour pouvoir calculer la variation du temps de cycle de deux protocoles étudiés, nous avons besoin de fixer les valeurs numériques des paramètres qui figurent dans les équations 3.12 pour l'EPL et 3.11 pour l'EtherCAT. Ces paramètres, comme le montre le tableau 3.1, décrivent le temps de cycle théorique de chaque protocole.

$$\begin{aligned} T_{cycle} &= T_{iso} + T_{asy} + T_{idle} \\ &= T_{sync} + T_{PRes} + M \times (T_{PReq} + T_{PRes}) + T_{SoA} + T_{ASnd} \\ &= T_{sync} + (2 \times M + 2) \times T_F + T_{ASnd} \\ &= T_{sync} + ((2 \times M + 2) \times b + MTU) \times t_{tx} \end{aligned} \tag{3.12}$$

Symbole	Valeur	Description
$Debit$	100 $Mbps$	Débit de transmission en full duplex.
t_{tx}	$0,08$ μs	Délai nécessaire pour transmettre un octet dans un réseau avec un débit de 100 Mbps.
MTU	300 $octets$	Taille maximum d'une trame asynchrone d'EPL.
b	64 $octets$	Taille d'une trame Ethernet pour un réseau EPL (la taille de la trame de type SoC, PReq, PRes et SoA).
x	16 $octets$	Payload d'un datagramme EtherCAT.
$T_{latence}$	$1,35$ μs	Temps de latence dû au passage de la trame par un esclave dans un réseau EtherCAT.
n_{max}	53	Nombre maximum de datagrammes dans un télégramme EtherCAT.
t_{tx}	$0,08$ μs	Temps nécessaire pour transmettre un octet avec un débit de 100 Mbps.
T_{sync}	45 μs	Temps d'attente pour le premier message P_{Req}.
T_{idle}	$0,96$ μs	Durée de la phase inactive.
M *ou* n	$1 - 239$ $esclaves$	Variation du nombre d'esclave dans un réseau à base d'Ethernet temps réel.

TABLE 3.1 – Valeurs numériques des paramètres du temps de cycle pour l'EPL et EtherCAT.

Délai de transmission d'une trame IEEE 802.3 c'est le temps nécessaire pour transmettre une trame qui respecte l'IEEE 802.3. Pour un débit de 100 Mbps ce délai varie entre $5,12\mu s$ (pour une trame de 64 octets) et $122,08\mu s$ (pour une trame de 1526 octets). Ce délai de transmission s'écrit :

$$T_F = b \times t_{tx} \tag{3.13}$$

où b est la taille en octet de trames (variant entre 64 et 1526 octets) et t_{tx} représente le temps nécessaire pour transmettre un octet dans un réseau en fonction de son débit (i.e. avec un débit de transmission $= 100Mbps$, $t_{tx} = 0,08\ \mu s$). Parmi les hypothèses de notre réseau, nous considérons que la taille minimum des trames est de 64 octets, ce qui donne un délai de transmission de $5,12\ \mu s$.

Délai de transmission d'une trame ASnd c'est le temps nécessaire pour la transmission d'une trame dans la phase asynchrone. Ce délai varie en fonction de la taille de la trame qui varie de 300 octets à 1500 octets.

Première synchronisation Dans la spécification d'EPL, la valeur typique T_{sync} est de 45 μs (T_W inclus).

Durée de la phase inactive c'est le temps d'inactivé du réseau après l'achèvement d'un cycle EPL.

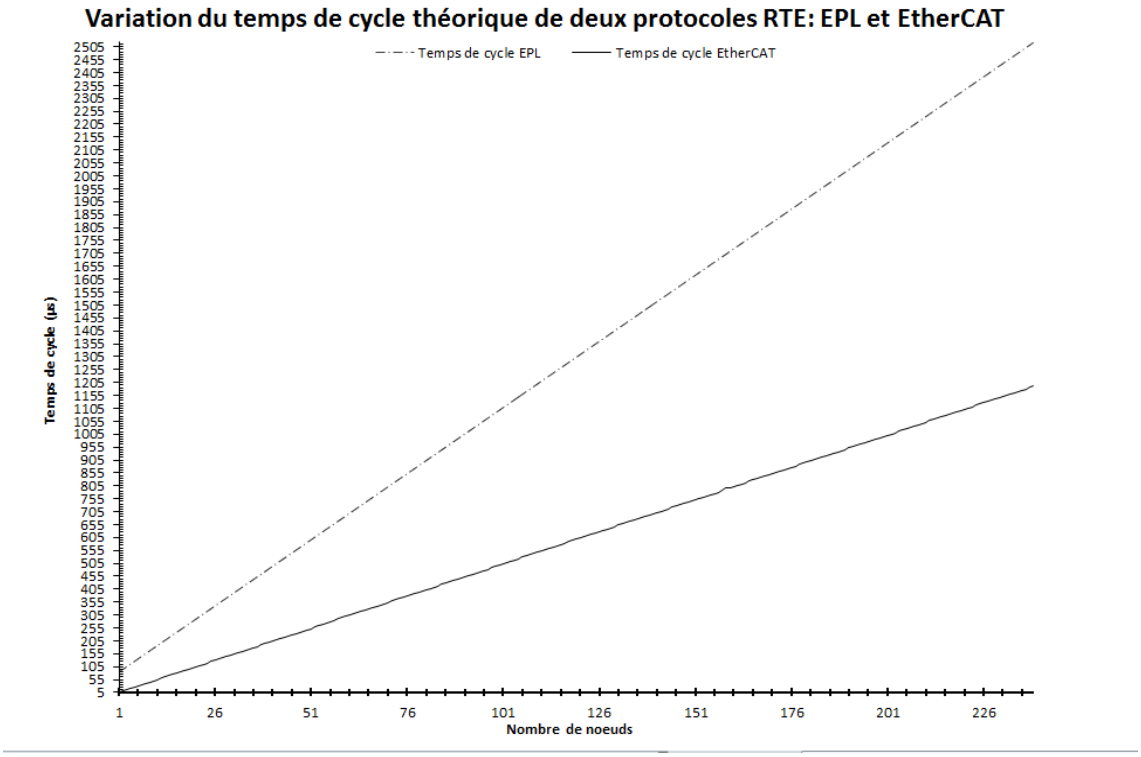


FIGURE 3.7 – Variation du temps de cycle théorique d'EPL et EtherCAT

En respectant les hypothèses du réseau décrites dans la partie 1.3.1.4, nous avons obtenu les résultats de la figure 3.7 en faisant varier le nombre d'esclaves de 1 à 239. Cette valeur théorique représente le nombre maximum d'esclaves permis dans un réseau EPL. Néanmoins, dans nos expérimentations, nous allons borner cette valeur à 100 nœuds. Nous constatons que la différence de performance entre les deux protocoles est proportionnelle à la variation du nombre d'esclaves. Par conséquent, une différence importante est visible lorsque le nombre d'esclaves est supérieur à n_{max}. Le temps de cycle d'EPL augmente d'une façon linéaire avec une pente très grande par rapport à l'EtherCAT. Cela est dû au nombre important de trames transmises dans le réseau pour interroger les esclaves. Cette particularité d'EPL (une trame pour chaque esclave)

montre son inconvénient pour un réseau industriel avec un nombre important d'esclaves. Par contre, grâce au découpage de la trame Ethernet, l'EtherCAT montre un meilleur résultat quand le nombre d'esclaves augmente. Malgré le mécanisme de fragmentation utilisé par l'EtherCAT, les performances de ce protocole restent plus avantageuses par rapport à l'EPL. Pour gagner en terme de temps de cycle d'EPL, nous pouvons creuser la piste d'augmenter la bande passante pour passer à un débit de 1 Gbps.

Je tiens à préciser que cette analyse est théorique, c'est-à-dire que nous nous sommes basés sur les spécifications théoriques de chaque protocole. Nous allons voir dans le chapitre 4 qu'en pratique, ce temps de cycle n'est pas le même qu'en théorie. Réellement il existe d'autres paramètres qui impactent le calcul du temps de cycle et qui sont liés directement aux caractéristiques physiques de l'équipement utilisé.

3.4 Conclusion

Dans ce chapitre, nous nous sommes servis de deux modèles de communication à base d'Ethernet ; l'EPL et l'EtherCAT pour garantir la communication dans le SCC. En effet, le modèle à base d'EPL fournit une transition douce pour la mise en œuvre de la chaîne de sécurité de l'ascenseur qui est devenue numérique. Avant de faire le déploiement de ce protocole sur des cibles physiques, nous avons identifié ses différentes phases qui constituent son temps de cycle. Cette étude théorique nous a permis de borner le temps de cycle avant sa mise en œuvre sur un banc d'essai expérimental qui sera décrit dans le chapitre 4. Ensuite, nous nous sommes concentrés sur l'évaluation théorique des performances du deuxième modèle à base d'Ethernet qui est l'EtherCAT. Cette évaluation est basée sur le calcul de son temps de cycle en fonction de différents paramètres. Pour garantir la safety de ces deux modèles, nous allons définir dans le chapitre 4 le concept de "black channel" selon l'IEC 61508. Afin de développer une couche safety basée sur ce concept, nous devrons identifier les différentes erreurs que nous pouvons avoir dans une communication industrielle. La prochaine partie sera consacrée à la description détaillée de notre SCC cible, l'ascenseur. En effet, nous allons présenter notre contribution par rapport à l'existant en précisant les conditions de sécurité imposées par la PESSRAL et la façon dont réagit notre modèle de communication proposé. Pour faire face à ces problèmes nous allons proposer les mesures de sécurité nécessaires pour passer par la suite à leur mise en œuvre dans la couche haute, indépendamment du modèle déployé (i.e. EPL ou EtherCAT).

Chapitre 4

Application : chaîne de sécurité d'ascenseur

Sommaire

4.1 Introduction

Dans ce chapitre nous décrivons l'architecture adaptée au système GESA. Ce modèle, proposé pour le démonstrateur D2 du projet ADN4SE, se base sur la réalisation des fonctions de protection sur des composants électroniques. La communication entre ces composants se fait grâce au modèle présenté dans le chapitre 3. Nous commençons ce chapitre par une comparaison entre l'architecture telle qu'elle est reliée actuellement sur la majorité des ascenseurs (systèmes électromécaniques) et celle que nous proposons pour la numérisation de la chaîne de sécurité qui permet de répondre aux exigences de la norme PESSRAL. Notre architecture va se baser sur un PES qui gère la sécurité du système. La deuxième section de ce chapitre sera consacrée à évaluer deux protocoles candidats pour ce nouveau système. Dans cette section nous allons tester la réalisation de chaque protocole afin de pouvoir les comparer. La métrique de comparaison reste

toujours le temps de cycle pour ce type de réseau. Nous introduisons les mesures de sécurité que nous avons développées dans la couche safety au dessus de la couche application des protocoles étudiés dans la deuxième section. Dans cette section du chapitre, nous prenons en considération les conditions de sécurité imposées par la PESSRAL pour le déplacement de l'ascenseur grâce aux scénarii de test. Cette partie va nous permettre de valider notre approche de communication à base d'Ethernet temps-réel et sécurisé, en respectant les contraintes fonctionnelles et temporelles imposées.

4.2 Contexte industriel : Vers une sécurité électronique du système de contrôle d'ascenseur

Dans le but de faire évoluer le système de contrôle d'ascenseur nous allons décrire le modèle existant afin de pouvoir identifier ses limites par rapport aux exigences d'évolution, qui ne sont pas négligeables. En effet, cette évolution n'apparait pas en tant que choix, mais plutôt une nécessité recommandée voire imposée par des normes de SdF telle que la norme PESSRAL. 'Cette dérivée', propre aux ascenseurs, permet de remplacer les systèmes de sécurité électromécanique de l'ascenseur par des systèmes électroniques programmables capables de gérer la sécurité du système de contrôle d'ascenseur lors de son déplacement. Nous rappelons que notre objectif est de proposer et valider un modèle de communication sécurisé à base d'Ethernet temps-réel dans de nouveaux systèmes de contrôle d'ascenseur. Dans cette partie, nous allons placer les travaux du chapitre 3 dans leur contexte industriel afin de pouvoir proposer une solution de sécurité adéquate aux exigences du système.

4.2.1 Modèle existant de l'ascenseur

Le transport des usagers dans un ascenseur est assuré par le déplacement de la cabine d'un étage à un autre que nous prendrons comme cas d'analyse. Le mouvement est assuré par un moteur entrainant un câble. Ce mouvement est commandé soit à partir de la cabine, soit à partir du palier, soit à partir de la machinerie. L'arrêt de la cabine hors zone de palier est assuré par le moteur et/ou un frein rattaché au moteur suite à l'ouverture de la chaîne de sécurité. Actuellement cette chaîne de sécurité repose sur des dispositifs électromécaniques branchés en série qui autorisent le déplacement de la cabine comme le montre la figure 4.1. Pour déplacer la cabine d'un étage à un autre, le centre de contrôle d'ascenseur procède à la vérification des conditions de sécurité du système. Avant d'autoriser ou non l'activation du moteur, le contrôleur vérifie l'état des portes au niveau des étages. Cette vérification conditionne l'alimentation électrique du moteur à travers les éléments de la chaîne de sécurité. Actuellement, cette chaîne est composée d'interrupteurs branchés en série qui délivrent ou non une tension sur une entrée dédiée au moteur afin d'autoriser le déplacement de la cabine dans des conditions de sécurité. Après vérification par le contrôleur que tous les interrupteurs de la chaîne de sécurité sont fermés (condition de sécurité vérifiée), le contrôleur commande le pilote moteur pour faire déplacer la cabine. Le déplacement de la cabine n'est possible que lorsque tous les éléments placés en série de la chaîne sont fermés.

Dans le cadre du démonstrateur D2, la composition matérielle de l'ascenseur étudié a été simplifiée par rapport à un ascenseur réel tel que nous le décrivons dans la figure 4.2. Seuls les composants nécessaires aux fonctions de base (mouvement de la cabine, ouverture/fermeture des portes, ouverture/fermeture de la chaîne de sécurité, etc.) sont considérés. Ce périmètre fonctionnel a été réduit afin d'être compatible aux parties correspondantes à notre tâche dans

le cadre du projet ADN4SE. Les échanges entre les différents composants se font actuellement à travers les bus de terrain : l'un des bus les plus utilisés par notre partenaire Sprinte dans la construction des systèmes de contrôle des ascenseurs est le réseau "CAN". Dans un souci de réduction des coûts (d'installation, de maintenance, de certification, etc.) et de modernisation (gain en fiabilité, robustesse, évolution, capacité à détecter et identifier les défauts, etc.), la société Sprinte est notre partenaire industriel dans le démonstrateur D2. La société souhaite faire évoluer les fonctions de sécurité non plus par des moyens électromécaniques mais via un système électronique programmable (PESSRAL) en assurant (en améliorant) le même niveau de sécurité. L'originalité de ce projet réside dans l'usage d'un système électronique appelé "GESA" qui permet d'intégrer un système d'exploitation temps réel (appelé Krono-OS) développé par la société "Krono-safe" qui est le principal leader du projet ADN4SE. Ce système GESA doit offrir une communication sécurisée et temps-réel pour assurer la SdF d'un nouveau système de contrôle d'ascenseur. Nous décrirons ce nouveau système dans la section suivante pour que nous puissions nous positionner par rapport à son architecture afin de l'intégrer.

4.2.2 Description du nouveau système de sécurité communicant pour l'ascenseur

Le système GESA est un système électronique programmable (PES en anglais) imposé par la norme PESSRAL pour remplacer l'ancien modèle précédemment développé sur des systèmes électromécaniques. Il permet de gérer la SdF dans le système de contrôle d'ascenseur en gardant un niveau de sécurité élevé. Selon la norme PESSRAL, le système GESA doit contenir :

— des interfaces d'entrée (i.e. convertisseur numérique-analogique),

— des équipements d'entrée (i.e. capteurs),

— un système de communication,

— une partie électronique programmable,

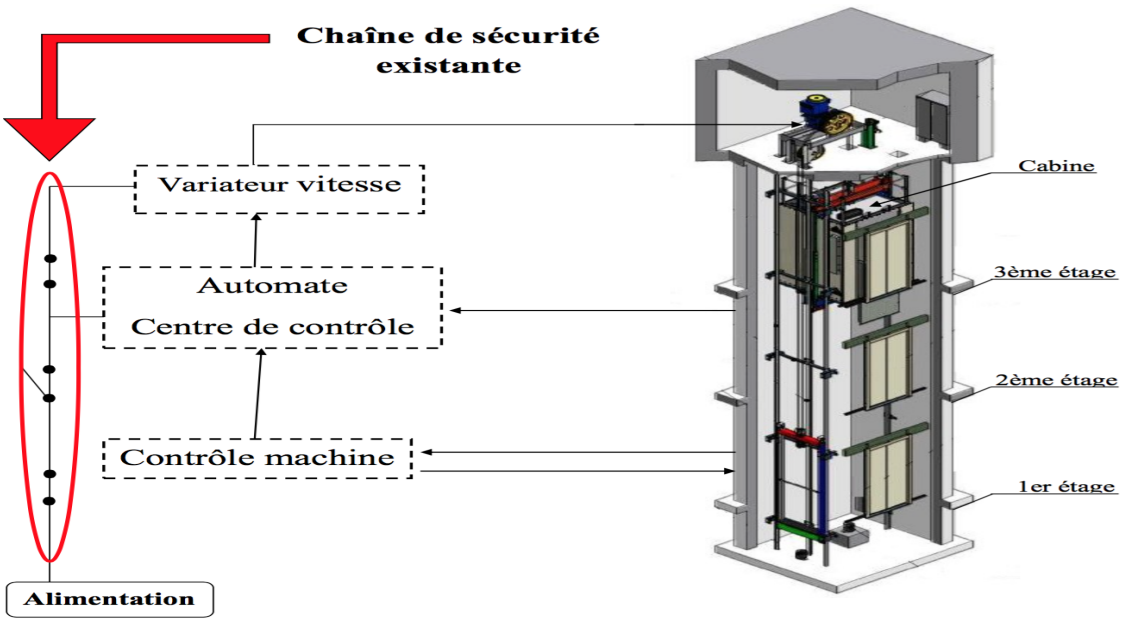


FIGURE 4.1 – Démonstrateur du modèle existant d'un système de contrôle d'ascenseur.

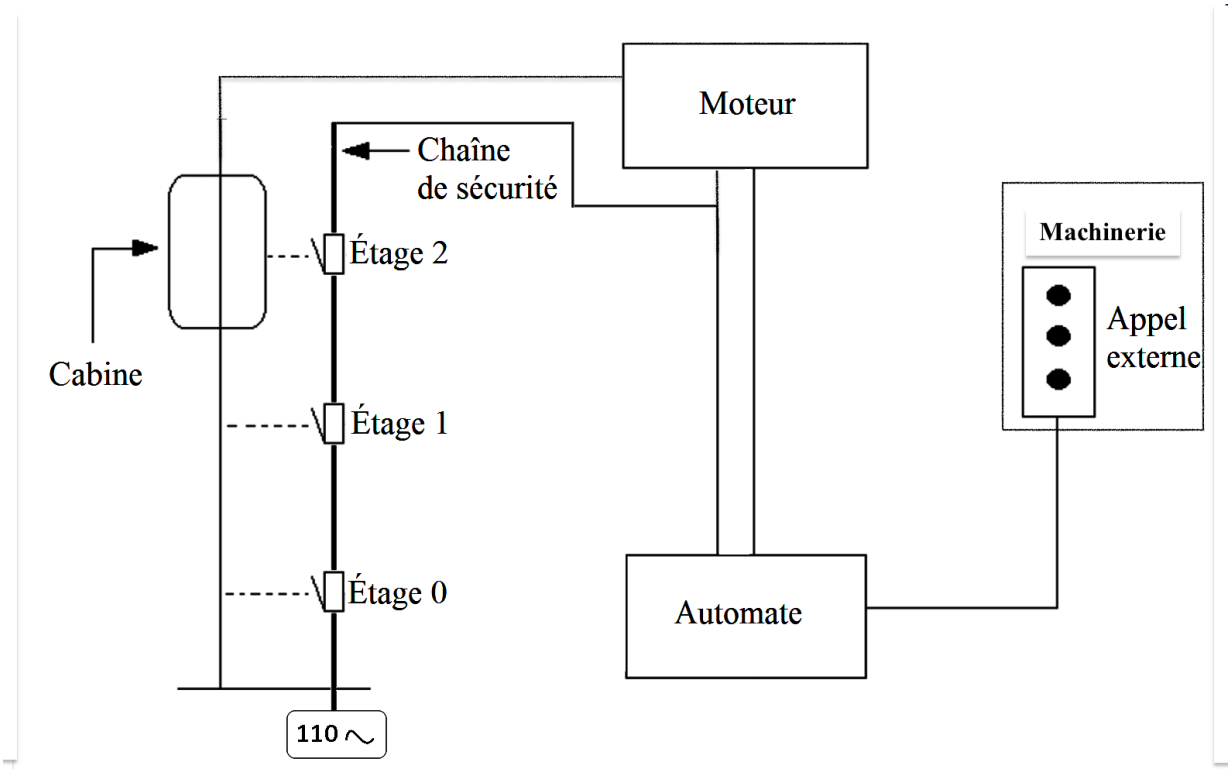


FIGURE 4.2 – Description technique réduite des composants principaux du modèle existant d'un système de contrôle d'ascenseur.

— des interfaces de sortie (i.e. convertisseur numérique-analogique) et
— un équipement de sortie (i.e. actionneurs).

Le système GESA doit vérifier toutes les conditions de sécurité afin d'autoriser le déplacement de la cabine de l'ascenseur. Par conséquent, il doit intégrer l'ensemble de fonctions critiques tirées de la norme PESSRAL ainsi que des fonctions non critiques. Comme l'illustre la figure 4.3 ce système est composé essentiellement de deux sous systèmes ; un système de CSM qui assure la sécurité en machinerie ainsi qu'aux paliers et le système de CSC qui prend en charge la sécurité en cabine.

Chacun de ces sous systèmes doit intégrer deux micro contrôleurs afin de satisfaire mutuellement aux exigences relatives au niveau de sûreté exigé. En effet, la communication entre les deux sous systèmes CSC et CSM doit être capable de garantir l'échange des fonctions critiques en temps réel pour assurer la surveillance réciproque des deux sous-systèmes (CSC et CSM). En effet, cette surveillance est basée principalement sur l'échange des signes de vie. Cet échange temps réel permet de garantir la synchronisation des nœuds du réseau de chaque sous-système. Ceci est dans le but de répondre à l'exigence de réactivité du système en général, et de sécurité en particulier. En effet, le système de signe de vie permet d'agir sur les contacteurs en cas de défaillance d'un composant du système. Cette exigence a un aspect normatif dirigé par la PESSRAL et l'IEC 61508 dans sa partie 7. En particulier, l'annexe 2.5 de la PESSRAL décrit explicitement le rôle de la composante communication. En effet, cette interface doit assurer :

— la surveillance mutuelle de CSC et CSM,
— la surveillance du codeur absolu et l'état du variateur de vitesse,

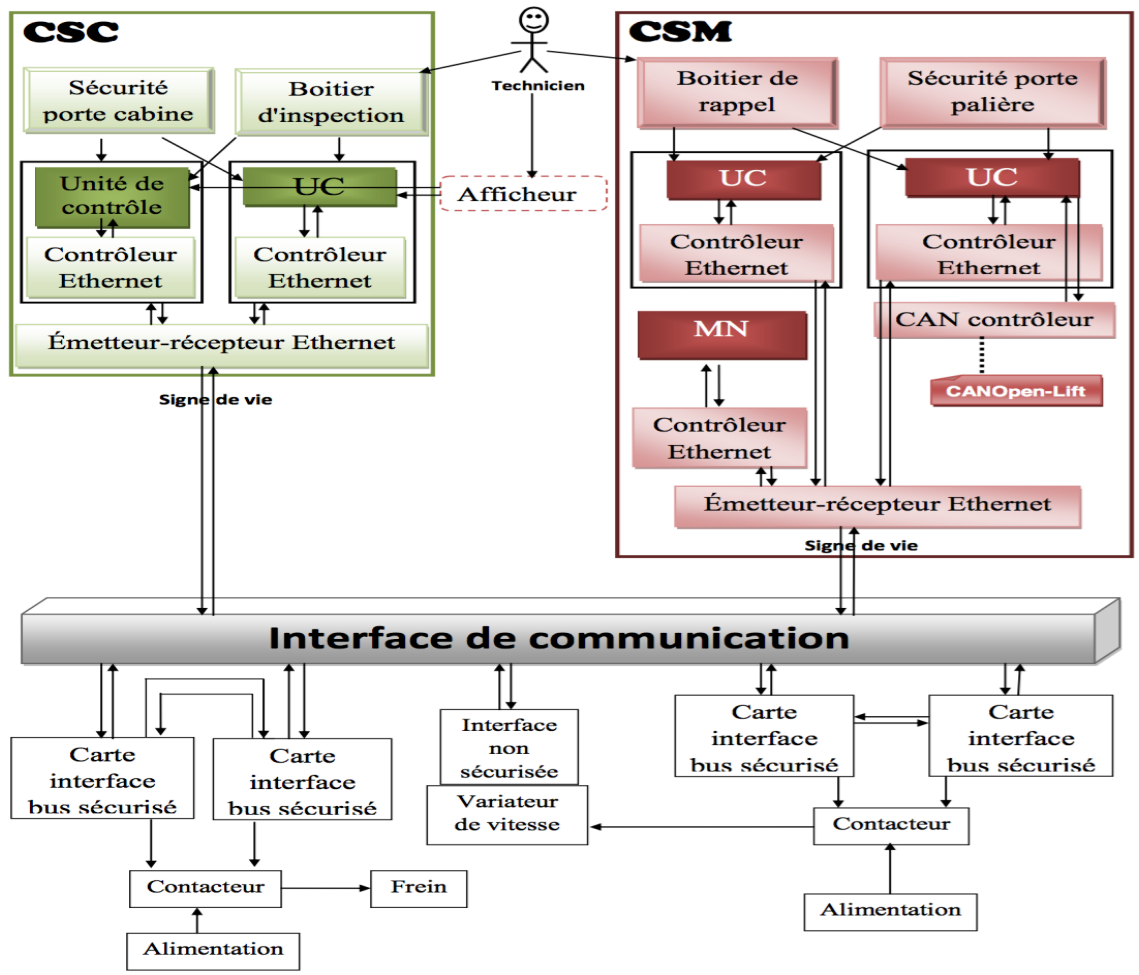


FIGURE 4.3 – Architecture matérielle dédiée à la sécurité de l'ascenseur

— l'envoi et réception des signes de vie,

— l'envoi et réception des messages de comparaison et

— la coexistence des messages critiques et non critiques sur la même interface.

Afin de concevoir le système GESA, nous avons besoin d'utiliser des cartes électroniques dans chacun des sous systèmes. L'architecture proposée par nos partenaires du démonstrateur D2 est illustrée dans la figure 4.3 : elle est composée ;

— d'une carte électronique en cabine (i.e.CSC) sur laquelle sont branchées les sécurités de la cabine qui doivent réagir en cas d'incident,

— d'une carte sécurité machine (i.e. CSM) qui devra arrêter le variateur du moteur en cas d'incident et

— d'un système de communication entre les deux cartes qui permet d'échanger des données fonctionnelles (position de la cabine, état du moteur, etc) et des données de sécurité (des informations sur l'état de la chaîne de sécurité, les messages d'actions en cas d'incident, etc.).

Par conséquent le système de communication doit montrer ses capacités temps réel pour pouvoir assurer un échange des messages à la fois temps réel et déterministe. Les données échangées

ont une forme spécifique liée à la nature du système de contrôle d'ascenseur. Elles doivent respecter le format du protocole CANOpen-Lift (déjà décrit dans la section 2.3). Nous allons proposer un modèle de communication entre ces deux sous-systèmes en développant tout d'abord, le protocole EPL (qui hérite du CAN) et enfin vérifier la compatibilité logicielle (format des données échangées) et matérielle (coût du matériel nécessaire) du protocole EtherCAT avec le système GESA.

4.3 Intégration de la communication à base d'Ethernet temps réel dans la chaîne de sécurité de l'ascenseur

Pour faciliter le contrôle, l'automatisation et la numérisation de la chaîne de sécurité actuelle, nous proposons un modèle basé sur un réseau industriel comme le décrit la figure 4.4.

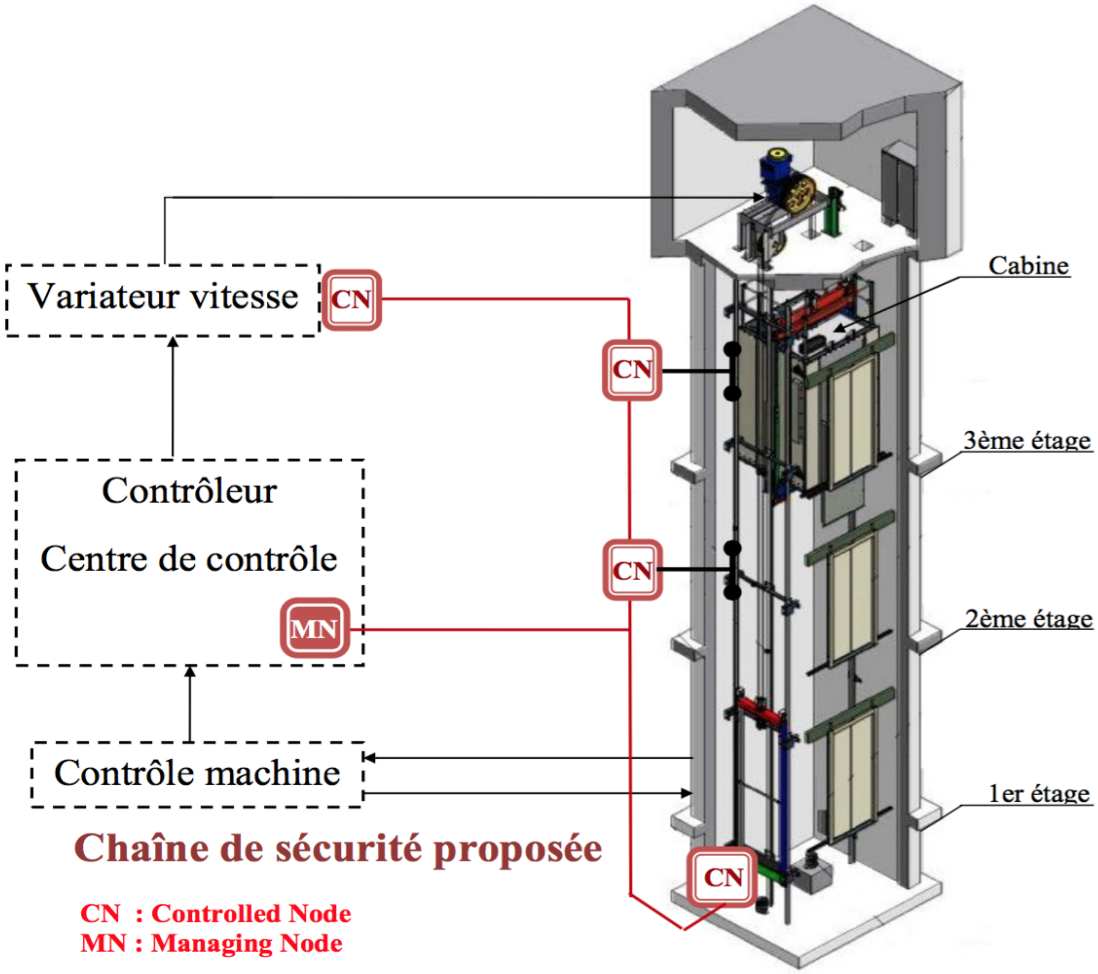


FIGURE 4.4 – Démonstrateur du nouveau modèle.

En effet, chaque interrupteur de la chaîne est représenté par un nœud esclave (CN). L'ensemble des CNs est piloté par un nœud maître (MN). Ce nœud maître collecte les informations sur les interrupteurs après avoir interrogé les nœuds esclaves du réseau. Ensuite le MN valide les commandes au contrôleur qui commandera le moteur pour déplacer la cabine. Les nœuds

vont communiquer à travers un protocole déterministe basé sur l'Ethernet temps réel et obéissant aux exigences physiques (équipements utilisés) et logicielles (utilisation des données de type CANOpen-Lift dans le système GESA) comme l'explique l'architecture matérielle dédiée à la sécurité de l'ascenseur illustrée dans la figure 4.3. Nous proposons en premier lieu le protocole EPL. L'utilisation du réseau EPL est adéquate au modèle actuel du système de contrôle de l'ascenseur. En effet, l'EPL facilite l'intégration du système de communication dans le système GESA grâce à son héritage du CAN. Pour interroger les nœuds CN nous aurons besoin d'un maître MN dans le contrôleur qui gère la vérification d'état des interrupteurs de la chaîne de sécurité comme le montre la figure 4.4. Par rapport à l'architecture des couches pour les systèmes industriels (i.e Computer Integrated Manufacturing (CIM) en anglais), notre système de contrôle communicant se positionne au niveau des deux premières couches comme le montre la figure 4.5. En effet, les nœuds CN de notre système seront placés au niveau des interrupteurs qui représentent la couche 1 (Capteurs, Actionneurs). Ils doivent collecter les informations qui décrivent l'état des interrupteurs. Ces informations seront transmises au maître MN qui fait partie du centre de contrôle et représente donc la couche 2 (Contrôle et Commande) de l'architecture CIM. Comme le montre la figure 4.3 le nœud EPL permet d'interfacer le CSM avec des équipements compatibles pour des fonctions critiques et non critiques.

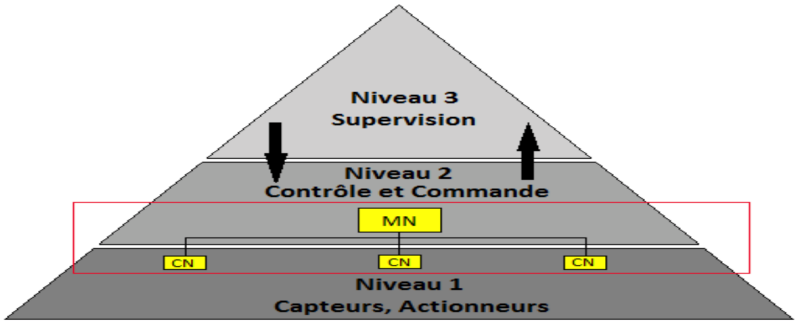


FIGURE 4.5 – Le nouveau système de contrôle d'ascenseur dans l'architecture CIM.

La partie de contrôle et commande du système de contrôle d'ascenseur devient capable de :

— collecter des informations sur la chaîne de sécurité grâce aux nœuds esclaves (CN),
— prendre la décision sur le déplacement de la cabine,
— commander l'automate depuis le nœud maître (MN),
— donner les ordres au moteur pour déplacer la cabine.

4.3.1 Réalisation d'un banc d'essai à base d'EPL

Dans cette section, nous détaillons le développement d'un banc d'essai expérimental dédié à la chaîne de sécurité de l'ascenseur. Ce banc d'essai, comme le montre la figure 4.6, est basé sur l'utilisation de maquettes électroniques qui seront intégrées dans le système GESA. Dans ce système nous allons déterminer le temps de cycle pratique dans le réseau EPL et le comparer avec sa valeur théorique calculée dans le chapitre précédent. Cette comparaison permet d'évaluer à la fois le comportement réel du protocole dans un contexte opérationnel et sa capacité à répondre aux contraintes temporelles normatives. Ce banc d'essai représente un environnement complet d'application pour le protocole EPL qui permet d'évaluer le comportement réel du protocole EPL avec des conditions de fonctionnement similaires à celles qu'il pourrait rencontrer lors de

son utilisation industrielle tel que les latences des nœuds, les délais des équipements du réseau, etc. Dans notre application, nous avons utilisé un ordinateur (PC) en tant que maître du réseau (MN), une carte STM32 qui représente un nœud esclave (CN1) et une carte Raspberry pour jouer le rôle d'un deuxième esclave (CN2) dans le réseau. L'ensemble des nœuds du réseau est interconnecté à travers un concentrateur Ethernet. Nous avons rajouté au réseau une sonde afin d'analyser le trafic échangé. En respectant le standard Ethernet, deux termes (IFG et $Gigue$) s'ajoutent au temps de cycle d'EPL déjà calculé dans le chapitre 3. Nous rajoutons aux hypothèses du chapitre 3, les caractéristiques des équipements utilisés :

— tous les nœuds CNs sont connectés à l'aide d'un câble Ethernet RJ45 (cat. 5e) de même longueur et

— tous les CNs ont le même temps de réponse (latence).

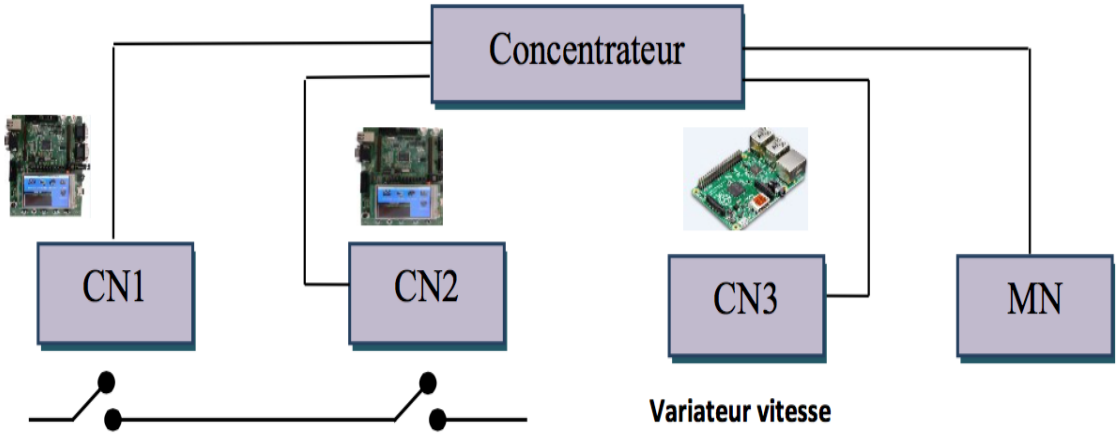


FIGURE 4.6 – Système de contrôle en réseau EPL pour l'expérimentation.

L'équation 4.1 décrit le temps de cycle en ajoutant ces deux termes. Sa valeur minimum dépend essentiellement de la taille des messages et du nombre des esclaves dans le réseau. Les différents termes qui figurent dans l'équation 4.1 sont décrits dans le tableau 4.1.

$$T_{cycle}^{EPL} = t_{tx} \times (L_{SoC} + L_{SoA} + (2M+1) \times L_{iso} + L_{asy}) + 2(M+2) \times IFG + Gigue \quad (4.1)$$

Nom	Valeur		Description
	Valeur min.	Valeur max.	
L_{iso}	64 *octets*	1526 *octets*	Longueur des données isochrones (L_{PReq} ou L_{PRes}).
L_{ASnd}	300 *octets*	1526 *octets*	Longueur des données asynchrones (L_{ASnd}).
L_{SoC}	64 *octets*		Longueur de la trame SoC.
L_{SoA}	64 *octets*		Longueur de la trame SoA.
t_{tx}	$0,08\ \mu s$		Temps nécessaire pour transmettre un octet (100 $Mbps$).
IFG	$0,96\ \mu s$		Inter-frame (Inter-frame Gap).
$Gigue$	$0,04\ \mu s$		Variation autour du délai moyen de traversée.

TABLE 4.1 – Variation des valeurs numériques des paramètres dans un réseau EPL.

En ajoutant les deux termes (IFG et $Gigue$) au temps du cycle d'EPL nous constatons une différence, comme l'illustre la figure 4.7 entre les résultats théoriques du chapitre 3 qui sont basés uniquement sur la taille de trames échangées et ces premiers résultats. En effet, l'étude théorique étant basée essentiellement sur la taille des messages échangés n'a pas traité les délais propres aux réseaux. En utilisant le standard Ethernet, un délai d'attente (de 12 octets) entre les trames est imposé pour garantir une transmission fiable. L'accumulation de ce délai dans un cycle EPL est la cause principale de cette différence constatée sur la figure 4.7. Cependant, nous ne devons pas négliger les délais dus à l'équipement d'interconnexion choisi. Ces équipements imposent des contraintes temporelles liées à leurs capacités physiques de traitement des trames. Néanmoins, le choix de concentrateur s'impose pour deux principales raisons :

— pour acheminer une trame, le concentrateur répète les données sur tous ses ports sans avoir besoin de générer un délai important pour l'acheminement, alors que le commutateur utilise une file d'attente pour pouvoir n'émettre la trame qu'à sa destination. Ce qui explique l'avantage que présente le concentrateur dans un réseau EPL par rapport au commutateur.

— le commutateur nécessite une variation autour du délai moyen (gigue) pour gérer la file d'attente implémentée. Cette gigue ne peut pas être négligée dans le cadre des applications industrielles (une gigue de plusieurs μs). Par contre, le concentrateur, n'impose aucune politique de mise en file d'attente. Par conséquent il en résulte une réduction importante de la gigue. En effet le concentrateur réduit cette variation (40 ns) autour du délai moyen de traversée.

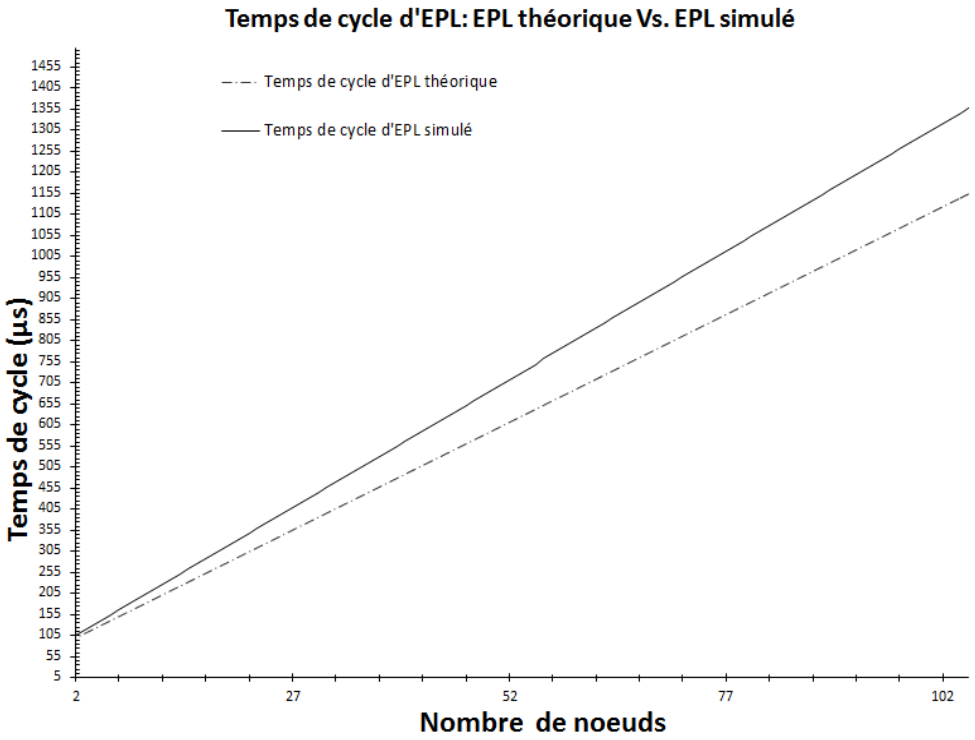


FIGURE 4.7 – Comparaison du temps de cycle minimum théorique d'EPL avec celui simulé.

4.3.1.1 La phase isochrone

Comme l'illustre la figure 4.8, pour commencer la phase de synchronisation, une trame de type SoC avec un temps de transmission T_{SoC} est envoyée depuis le MN pour arriver au concentrateur afin d'être diffusée en multicast aux CNs du réseau pour les informer du début de cette phase.

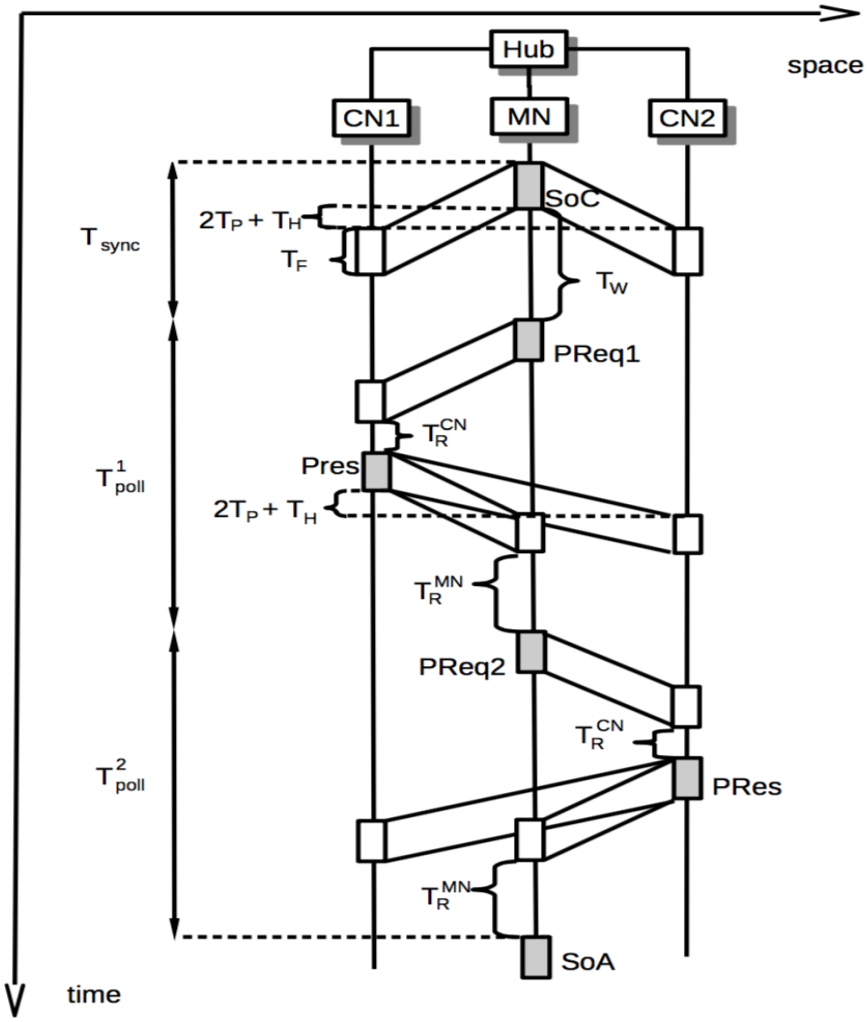


FIGURE 4.8 – Diagramme espace-temps du cycle EPL avec un maître et deux esclaves.

Cette procédure nécessite un délai minimum de $M \times T_p + T_H$, où M est le nombre de CNs dans le réseau EPL, T_p et T_H sont respectivement le délai de propagation dû au câble et le délai d'un concentrateur y compris la gigue. Selon ses spécifications, le protocole EPL exige un temps d'attente minimum avant l'envoi du prochain PReq. Une fois que tous les CNs ont reçu la trame SoC, la phase d'interrogation s'active et les CNs se mettent à l'attente des interrogations du MN. Une trame PReq, issue du MN à destination d'un CN particulier est transmise (en unicast). La durée de cette opération est exprimée par $T_{PReq} + M \times T_p + T_H$. En outre le CN ajoute un temps d'attente T_R^{CN} qui est dû à la latence du nœud.

$$\begin{cases} T_{cycle} &= T_{sync} + T_{iso} + T_{asy} + T_{idle} \\ T_{sync} &= T_{SoC} + M \times T_{Propagation\ SoC} + T_H + T_W \\ T_{iso} &= \sum (T_{PReq} + T_{Propagation\ PReq} + T_{Reaction\ PReq} + T_{PRes} \\ & \quad + T_{Propagation\ PRes} + T_{Reaction\ PRes} + T_H) + T_{PRes} + T_{Propagation\ PRes} \\ T_{asy} &= T_{SoA} + T_{Propagation\ SoA} + T_{Reaction\ SoA} + T_{ASnd} \\ & \quad + T_{Propagation\ ASnd} \\ T_{idle} &= 0,96\mu s \end{cases} \tag{4.2}$$

Pour la phase de réponse, la procédure est similaire. Le CN envoie une trame PResp en multicast à travers le réseau via le concentrateur. Il est important de souligner que durant la diffusion en multicast de la trame PResp, le temps total pour cette opération est le même que calculé pour la PReq. En effet, l'opération de multicast s'est déroulée en parallèle. Par conséquent, il n'y a plus de délai à ajouter. Encore une fois, dès réception de PRes, le MN introduit un temps d'attente T_R^{MN} dû à la latence de l'équipement. Selon les observations précédentes, détaillées dans le diagramme espace-temps de la figure 4.8, l'équation 4.3 décrit le T_{sync} et le T_{poll}.

$$\begin{cases} T_{sync} &= T_{SoC} + M \times T_{Propagation\ SoC} + T_H + T_W = 45\mu s \\ T_{poll} &= T_{PReq} + T_{PRes} + T_{delai} \\ &= T_{PReq} + T_R^{CN} + T_{PRes} + T_R^{MN} + ((M+1) \times T_P + T_H) \\ &= (T_{PReq} + T_{PRes}) + (T_R^{CN} + T_R^{MN}) + ((M+1) \times T_P + T_H) \end{cases} \tag{4.3}$$

où T_{PReq}, T_{PRes} sont respectivement le temps de transmission des trames PReq et PRes et T_{delai} est la somme des délais liés à cette transmission. Dans notre exemple (un MN et deux CNs), le T_{delai} est composé du délai de propagation lié au câble T_P, délai de répétition lié au concentrateur T_H, le temps d'attente exigé pour que tous les CNs puissent recevoir la trame SoC et les deux latences T_R^{CN}, T_R^{MN}. Le tableau 4.2 définit les paramètres d'un scénario opérationnel.

Dans le cas général, avec M CNs, l'expression de T_{iso} devient :

$$\begin{aligned} T_{iso} &= T_{sync} + M \times T_{poll} + T_{PRes} + (M \times T_P + T_H) \\ &= T_{sync} + M \cdot [T_{PReq} + T_{PRes} + T_R^{CN} + T_R^{MN} + (M+1) \cdot T_P + T_H] + T_{PRes} + M \cdot T_P + T_H \end{aligned} \tag{4.4}$$

Nous avons gardé les mêmes valeurs numériques de certains paramètres (de la spécification du protocole) que nous avons identifiés dans le chapitre 3.

Symbole	Valeur	Description
T_F	$F \times t_{tx}$	Délai de transmission d'une trame IEEE 802.3 de taille F (100 Mbps).
T_H	$0,68\ \mu s$	Délai d'un concentrateur y compris la gigue.
T_{sync}	$45\ \mu s$	Temps d'attente pour le premier message P_{Req}.
T_P	$10\ ns$	Délai de propagation dû au câble.
T_R^{MN}, T_R^{CN}	de 2 μs à 50 μs	Latence d'un nœud EPL ou délai de réactivité (selon les caractéristiques de l'équipement ; 8 μs pour les nœuds B and R et 16 μs pour les PC).
M	de 2 à 100	Nombre d'esclaves dans un réseau EPL.

TABLE 4.2 – Valeurs des paramètres réels dans notre réseau EPL

Délai de transmission d'une trame IEEE 802.3 C'est le temps nécessaire pour transmettre une trame qui respecte l'IEEE 802.3. Pour un débit de 100 Mbps ce délai varie entre $5,12\mu s$ (pour une trame de 64 octets) et $122,08\mu s$ (pour une trame de 1526 octets). Ce délai de transmission s'écrit :

$$T_F = F \times t_{tx} \tag{4.5}$$

avec F la taille en octet de trames (variant entre 64 octets et 1526 octets) et t_{tx} le temps nécessaire pour transmettre un octet dans un réseau en fonction de son débit (i.e. avec un débit de transmission $= 100Mbps$, $t_{tx} = 0,08\ \mu s$).

Délai généré par le concentrateur Idéalement, lorsqu'une trame Ethernet arrive dans l'un des ports du concentrateur, elle est diffusée aux autres ports en même temps que son arrivée. Mais en réalité, les composants électriques du concentrateur génèrent un retard entre la réception de la trame et sa diffusion aux autres ports. Ainsi les concentrateurs ont un effet direct sur le temps de cycle EPL.

Première synchronisation (T_{sync}) Dans la spécification d'EPL, la valeur typique T_{sync} est de $45\ \mu s$ (T_W inclus).

Latence d'un nœud EPL C'est le temps nécessaire d'un nœud pour réagir à une entrée. En effet, c'est le temps requis pour changer l'état de sortie en fonction d'une entrée. Dans un réseau EPL, le nœud CN_i reçoit la trame $PReq$ du maître de réseau MN. Le CN prend un temps T_R^{CN} avant d'envoyer sa réponse via une trame $PRes$. De la même façon, le MN, avant d'interroger le prochain CN_{i+1} via une trame $PReq$ requiert un temps de mise à jour T_R^{MN}. En d'autre terme c'est le temps de réaction d'un nœud.

4.3.1.2 La phase asynchrone

La figure 4.9 montre un diagramme espace-temps de la phase asynchrone de notre réseau EPL. L'expression T_{asy} de la phase asynchrone est calculée à partir de l'équation 4.6.

$$T_{asy} = T_{SoA} + T_{ASnd} + 2 \cdot (T_H + M \times T_P) + T_R^{CN} \tag{4.6}$$

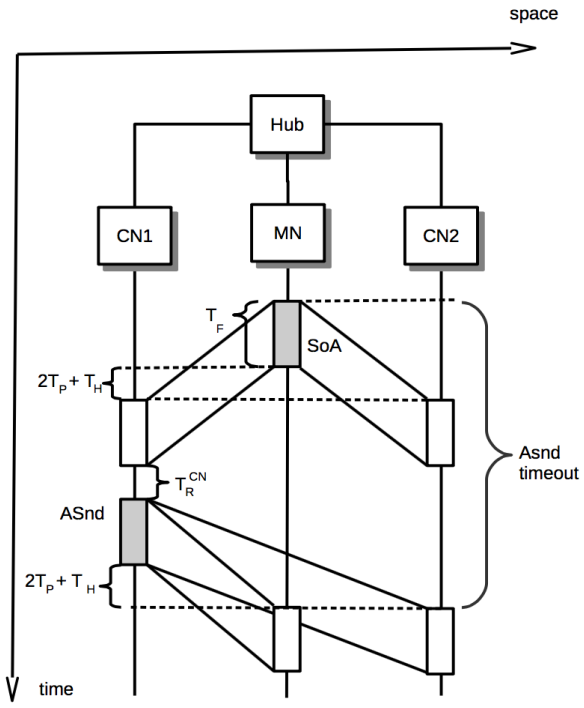


FIGURE 4.9 – Diagramme espace-temps de la phase asynchrone d'une communication EPL.

4.3.1.3 Résultats

Avec l'hypothèse que tous les délais $T_{SoA}, T_{PRes}, T_{PReq}$ sont égaux, l'expression 4.7 devient :

$$\begin{cases} T_{cycle} &= T_{sync} + T_{iso} + T_{asy} + T_{idle} \\ T_{cycle} &= T_{sync} + M \cdot [T_{PReq} + T_{PRes} + T_R^{CN} + T_R^{MN} + (M+1) \cdot T_P + T_H] \\ & \quad + T_{PRes} + M \cdot T_P + T_H + T_{SoA} + T_{ASnd} + 2 \cdot (T_H + M \times T_P) + T_R^{CN} + T_{idle} \\ T_{cycle} &= T_{sync} + T_{idle} + t_{tx} \cdot [(2M+1)L_{min} + L_{ASnd}] + (M+1) \cdot T_R^{CN} + M \cdot T_R^{MN} \\ & \quad + (2M+1) \cdot T_H + M \cdot (M+4) \cdot T_P \\ T_{sync} &= 45\ \mu s \\ T_{idle} &= 0,96\ \mu s \end{cases} \quad (4.7)$$

Comme le montre la figure 4.10, le choix des équipements demeure primordial dans un contexte industriel contraint en temps réel. Théoriquement, le constructeur d'EPL annonce qu'un cycle EPL peut atteindre des valeurs inférieures à 1 ms(de l'ordre de 400 μs [Kaczmarczyk *et al.*, 2011]). Cependant, en pratique, l'EPL ne peut pas atteindre ces performances temporelles idéales si les caractéristiques physiques ne le permettent pas même en cas d'une configuration détaillée et précise des paramètres (voir annexe A). Les valeurs théoriques peuvent être atteintes en utilisant un matériel spécifique (i.e. FPGA ou ASIC) aux niveaux des nœuds esclaves et du maître permettant un traitement rapide de données. En effet, notre mise en œuvre n'a pas pu atteindre cette valeur. L'augmentation du temps de cycle de l'implémentation du réseau est due à la limite des équipements utilisés ainsi qu'au nombre des nœuds utilisés.

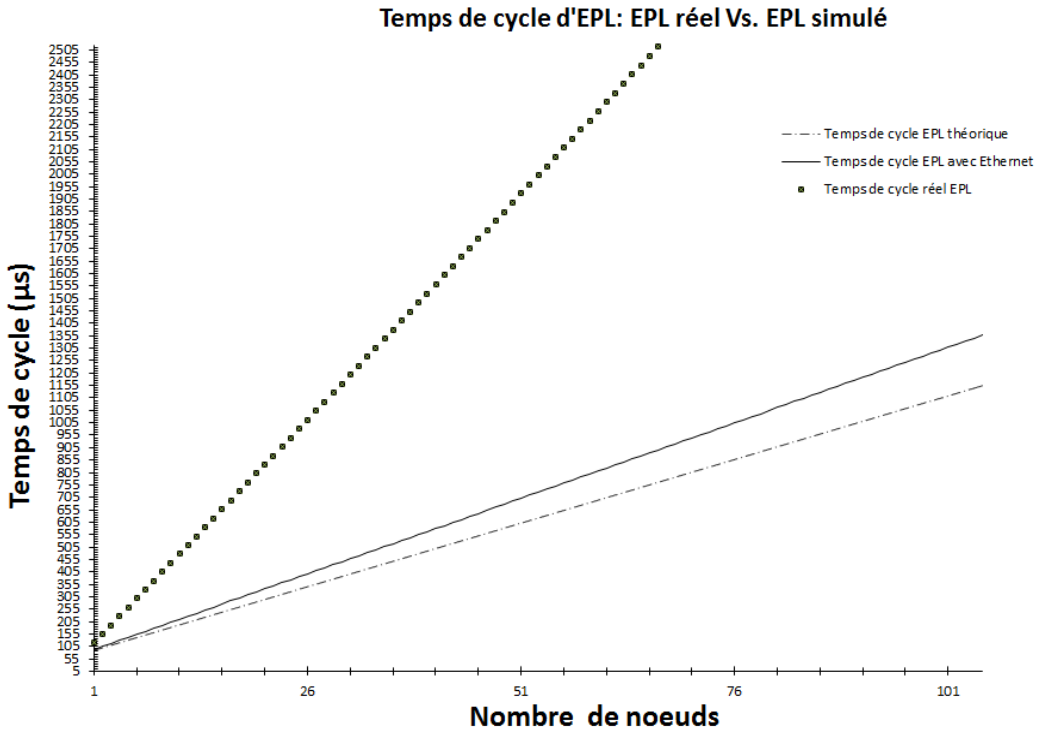


FIGURE 4.10 – Comparaison du temps de cycle minimum d'EPL en faisant varier le nombre d'esclaves dans le réseau.

4.3.2 Simulation du réseau EtherCAT

L'étude d'EtherCAT du chapitre précédent s'est basée principalement sur ses spécifications fonctionnelles sans prendre en considération la capacité de chaque nœud (principalement le délai) ni la topologie du réseau déployé. D'ailleurs, théoriquement, le protocole EtherCAT peut être implémenté dans plusieurs topologies (linéaire, anneau, étoile, arbre, etc.) en utilisant des commutateurs Ethernet standard. Étant donné que la performance du réseau est en partie dépendante de sa topologie, il est nécessaire d'utiliser des architectures réalistes et efficaces dans les calculs. Par ailleurs, un réseau EtherCAT se base sur le principe de "daisy chain" comme le montre la figure 4.11 qui illustre que la topologie en ligne est très efficace pour le réseau. En se basant sur ce principe, le maître du réseau doit être connecté à un ensemble d'esclaves qui sont interconnectés entre eux, d'où la nécessité de rajouter un port Ethernet [Eth, 2017]. Grâce à ses deux ports Ethernet, un nœud EtherCAT devient capable d'assurer le fonctionnement en mode "on the fly" pour ne pas générer un retard dans la communication.

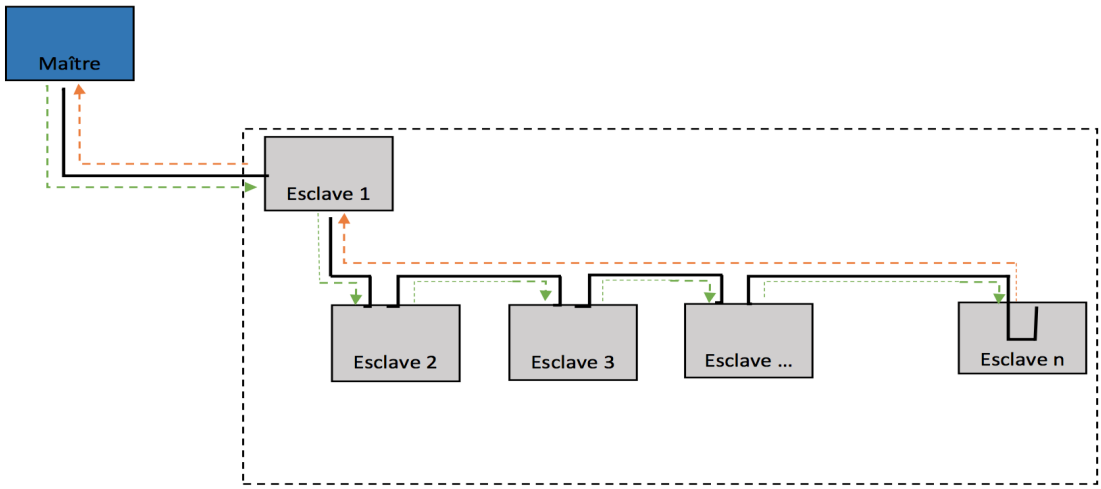


FIGURE 4.11 – Topologie physique et logique d'un réseau EtherCAT.

Le principe de base de fonctionnement d'EtherCAT est que tous les nœuds du réseau peuvent lire et écrire leurs données sur la même trame dans une courte période de temps. En effet, tous les datagrammes EtherCAT sont générés et envoyés par le maître dans une trame Ethernet en utilisant une transmission full duplex. Dès réception de cette trame, chaque esclave peut lire/écrire ses données à la volée dans le datagramme qui lui est attribué. Ces datagrammes, regroupés dans une trame, sont retournés au maître à la fin de chaque segment du réseau (dernier esclave du réseau). EtherCAT utilise des trames Ethernet standards telles que définies par IEEE 802.3 avec des longueurs de trames totales allant de 64 à 1526 octets. Lorsque le nombre de trames dans un cycle est supérieur à un, l'écart entre les trames successives de 12 octets doit également être pris en compte (IFG). La durée des opérations de lecture/insertion des données correspond à la latence d'un nœud. Elle est indépendante de la taille de trame. L'[Eth, 2017] exige que le retard constant du nœud doit être inférieur à 500 ns. Pour atténuer ce retard, L'[Eth, 2017] recommande d'utiliser un matériel spécifique (FPGA ou ASIC) pour des topologies complexes. Il convient de noter qu'en utilisant uniquement des esclaves EtherCAT standards, la topologie logique possible est en pratique une topologie anneau. Pour pouvoir évaluer les performances réelles d'EtherCAT, nous devons identifier les paramètres dont dépend son temps du cycle. Avant de nous engager dans l'achat des équipements recommandés par [Eth, 2017] tel que la FPGA ou l'ASIC qui ont un coût important, nous avons choisi de simuler le comportement d'EtherCAT grâce à OMNet++ [Omn, 2016] dans sa version 4.6 en utilisant le paquetage inet 3.0 [Varga, 2005] comme le montre la figure 4.12. Dans la simulation d'EtherCAT, nous avons étudié en détails ces paramètres afin de pouvoir évaluer les performances "réelles" des deux protocoles candidats EPL et EtherCAT dans le démonstrateur D2. En effet, nous avons pu identifier les paramètres du tableau 4.3.

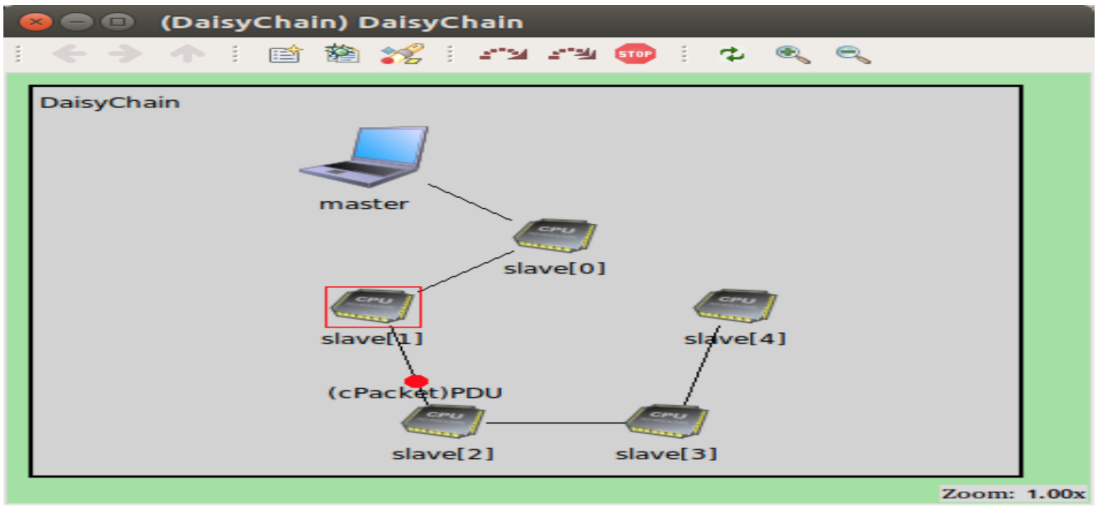


FIGURE 4.12 – Simulation du réseau EtherCAT sous OMNet++ version 4.6 sous un environnement Linux.

Symbole	Valeur	Description
t_{tx}	$0,08\ \mu s$	Temps nécessaire pour transmettre un octet dans un réseau de $100Mbps$.
T_F^{Master}	$F \times t_{tx}$	Délai de transmission de la trame au niveau du maître avec un débit $100Mbps$.
T_{RXTX}^{max}	$500\ ns$	Délai maximum de (Rx + Tx).
T_P	$10\ ns$	Délai de propagation dû au câble.
T_F^{Slave}	$450\ ns$	Délai de transmission du datagramme à l'intérieur de l'esclave.
T_F	$123,04$	Délai de transmission d'une trame de taille maximale avec l'IFG (12 octets)
T_{delai}^{ETH}	$960\ ns$	Retard total introduit par un esclave EtherCAT.

TABLE 4.3 – Valeurs des paramètres simulés d'un réseau EtherCAT.

La valeur du délai d'attente d'un esclave EtherCAT est d'environ 450 ns (la somme du délai dans les deux sens). En ajoutant cette valeur, nous retrouvons les délais physiques (RX et TX) qui varient quelque peu entre les différentes architectures, selon les constructeurs, mais la somme des retards de réception et de transmission d'un port physique ne doit en aucun cas dépasser les 500 ns comme mentionné précédemment (valeur exigée par [Eth, 2017]). En outre, il existe un délai de propagation du câble qui est généralement inférieur à 50 ns qui dépend de la nature du câble Ethernet utilisé (i.e. Cat. 5) et de la distance entre les nœuds (i.e. 10 m).

Toutes les abréviations utilisées dans les équations sont expliquées dans le tableau 4.3.

Lorsque une seule trame est capable de supporter les datagrammes EtherCAT de tous les esclaves du réseau, la formule du temps de cycle est décrite dans l'équation 4.8, où n est le nombre d'esclaves dans le réseau, n_{max} est le nombre maximum de datagrammes que peut contenir une seule trame Ethernet, sous condition que $n \leq n_{max}$.

$$\begin{aligned} T_{cycle}^{ETH} &= T_{cycle-EtherCAT}^{Theo} + T_P + n \times (T_P + T_{RXTX}^{max} + T_F^{Slave}) \\ &= (2n-1)T_{latence} + ((28k + (k-1)n_{max}(12+x))) \times t_{tx} \\ &\quad +(max(36,\ (n-(k-1)n_{max}) \cdot (12+x))) \cdot t_{tx} + T_P + n(T_P \\ &\quad +T_{RXTX}^{max} + T_F^{Slave}) \end{aligned} \tag{4.8}$$

Dans le cas général où plusieurs trames peuvent échanger dans un cycle EtherCAT ($n > n_{max}$.), nous obtenons l'équation 4.9.

$$\begin{aligned} T_{cycle-EtherCAT} &= T_P + n \cdot T_{delai}^{ETH} + T_{transmission-EtherCAT} \\ &= T_P + n(T_P + T_{RXTX}^{max} + T_F^{Slave}) + (k-1) \cdot T_F \\ &\quad +max(36,\ (n-(k-1) \cdot n_{max}) \cdot (12+x)) \times t_{tx} \end{aligned} \tag{4.9}$$

Le résultat de la simulation d'EtherCAT est illustré dans la figure 4.13. Nous constatons que le délai ajouté ne change pas trop le comportement du réseau, malgré l'augmentation de quelques micro secondes du temps de cycle d'EtherCAT.

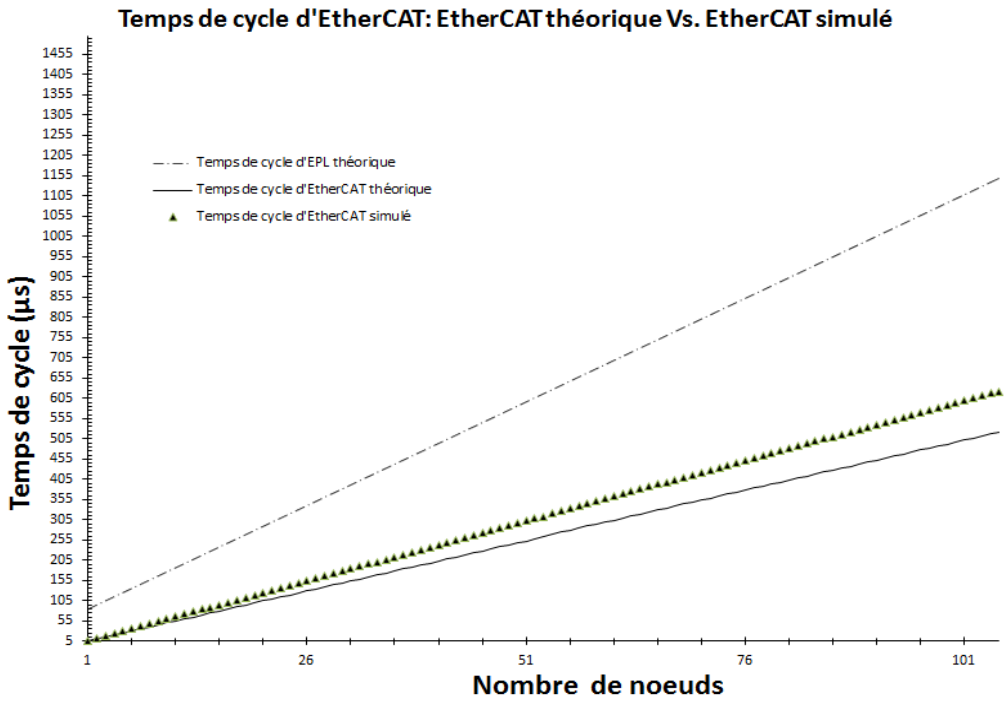


FIGURE 4.13 – Comparaison du temps de cycle minimum d'EtherCAT en fonction du nombre d'esclaves dans le réseau.

4.3.3 Comparaison d'EPL et d'EtherCAT

Après avoir évalué chaque protocole dans le contexte industriel, nous pouvons faire la comparaison des résultats obtenus. En effet la figure 4.14 montre clairement que l'EPL n'est pas très

performant quand il s'agit d'une communication temps réel critique dans un système très réactif.

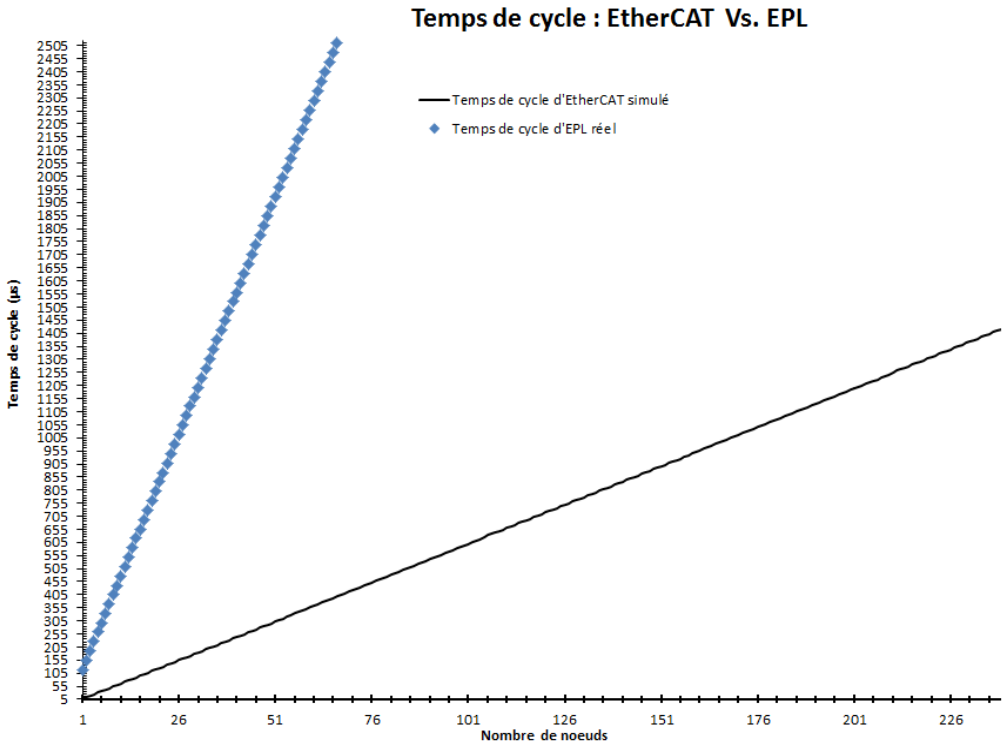


FIGURE 4.14 – Comparaison du temps de cycle minimum d'EtherCAT avec celui d'EPL en fonction du nombre d'esclaves dans le réseau.

L'EtherCAT a montré des performances temporelles plus intéressants que EPL et pourtant, c'est l'EPL qui a été retenu pour le système de contrôle d'ascenseur puisque son modèle a été validé par l'outil krono-OS. Le démonstrateur D2 du projet ADN4SE a pu atteindre ses objectifs grâce au système GESA qui intègre une communication déterministe et temps réel basée sur l'EPL.

4.4 Mesures de sécurité dans une communication Ethernet temps réel

Certes, les performances (i.e. temps de réponse, temps de cycle, etc.) de la communication industrielle, en particulier celles basées sur l'Ethernet temps réel sont nécessaires, mais un autre critère important doit être pris en considération dans ce type de communication, il s'agit de la "safety" des données échangées dans ces protocoles. Les réseaux déterministes à base d'Ethernet temps-réel sont loin d'être "safe". Ils garantissent une synchronisation parfaite entre les équipements et répondent aux exigences temps réel du système mais pas à celle liées à la "safety". En effet, plusieurs variables affectent le transfert de données liées à la "safety" dans les applications Ethernet (i.e. les switchs, les hubs, etc). En outre, chaque application utilise de nombreuses piles logicielles internes. Cet environnement ne tient pas compte des données liées à la "safety". Une solution existante pour la safety des données dans les systèmes critiques est par exemple la méthode de redondance physique pour la communication avec l'utilisation de deux micro-contrôleurs de chaque côté dans le réseau comme le montre la figure 4.15.

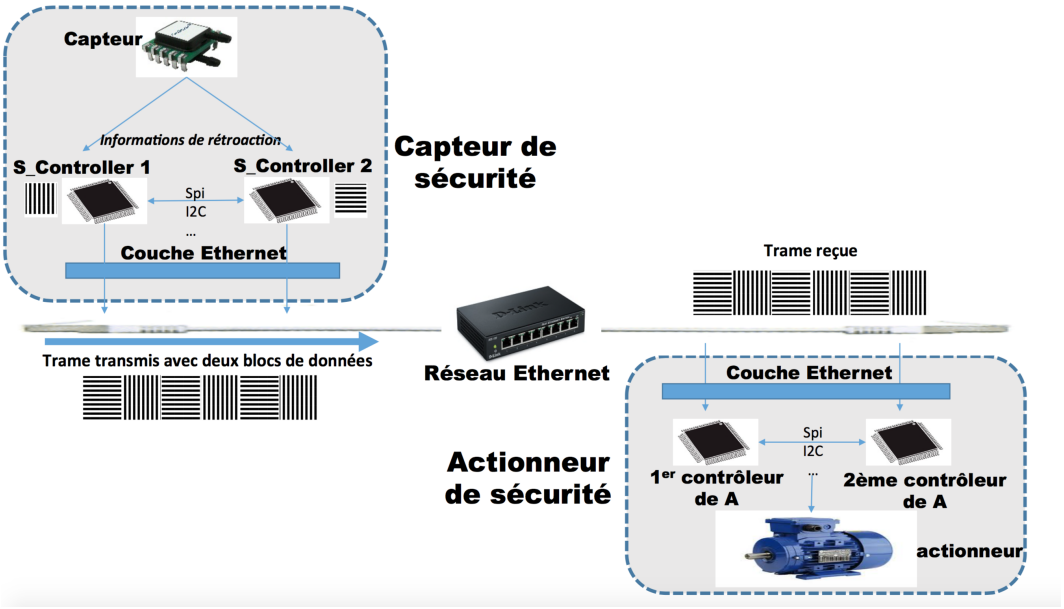


FIGURE 4.15 – Génération des données critiques via la redondance physique

Les deux micro-contrôleurs S_Controller 1 et S_Controller 2 dans le nœud "safety sensor" déterminent chacun la valeur du capteur d'une façon totalement indépendante afin de calculer les fonctions de chaque capteur. Le premier contrôleur (S_Controller 1) génère les données avec l'entête, numéro consécutif (consecutive number) et le nom de variables avant la transmission. Le second contrôleur (S_Controller 2) inverse toutes les données avant la transmission. Cependant, une autre option s'ajoute par la génération de toutes les données par le premier contrôleur avant d'être vérifiées par le deuxième micro-contrôleur. Une fois que la génération des données est achevée par le nœud "safety sensor", l'Ethernet encapsule les deux structures de données dans une seule trame. Ensuite le récepteur implémente le même processus dans le sens inverse. En effet, à travers la couche Ethernet, il isole les deux blocs de données et les transfère chacun à un de ces contrôleurs (A_Controller) dans le "safety actuator". Chaque contrôleur vérifie et compare le contenu de ces données. Du point de vue des performance du réseau, le processus de séparation des données nécessite un délai temporel supplémentaire qui n'est pas toléré dans les systèmes hyper-critiques avec un coût économique élevé dû à la duplication du matériel.

4.4.1 Mécanismes de SdF pour la communication numérique

Pour la communication numérique dans les systèmes critiques, la tolérance aux fautes, décrite dans la section 1 d'après [Laprie *et al.*, 1996], qui est un moyen pour assurer la SdF des systèmes est basée essentiellement sur la redondance qui consiste à dupliquer le composant (duplication logicielle, matérielle données, hybride, etc.). La notion de redondance est une notion classique et bien connue dans les systèmes tolérants aux fautes. Dans la littérature, [Dubrova, 2013] et [Knight, 2012] nous trouvons différents types de redondance déployés dans les architectures classiques de système critique.

— Redondance symétrique/asymétrique : Deux classes peuvent être distinguées :
  — La redondance symétrique consiste à dupliquer d'une façon identique le composant original (i.e. deux processeurs, deux micro-contrôleurs, etc.).

  - La redondance asymétrique qui exige une copie complètement différente du composant original (i.e. deux micro-contrôleurs différents en conception et construction, etc.).
- redondance statique/ dynamique/ hybride
  - la redondance statique selon [Knight, 2012] : il s'agit de la redondance active selon [Dubrova, 2013], consiste à faire fonctionner les composants et leurs copies (duplications) simultanément, ce qui nécessite une tâche de comparaison/ vérification afin de comparer les données de chaque composant ce qui permet de détecter les anomalies au cas où les données comparées du composant original et du composant copie sont différentes. Cette redondance active permet de garder le système en fonctionnement en cas de dysfonctionnement d'un composant, ce qui permet d'assurer la disponibilité du système
  - la redondance dynamique, ou passive selon [Dubrova, 2013] : Ce type de redondance impose une condition (évènement) pour faire fonctionner/initier la partie dupliquée dans le système. En effet, la mise en fonctionnement du composant dupliqué est déclenché par un évènement particulier (défaillance du composant principal). Par exemple, la duplication des composants, sans les mettre en tension jusqu'à la défaillance du composant principal.

    Avantage : elle est moins coûteuse en consommation d'énergie pour les composants physiques. Elle entraine moins d'usure sur des composants redondants.

    Inconvénient : le niveau de disponibilité doit être plus élevé.
- redondance temporelle/spatiale
  - la redondance temporelle consiste à dupliquer dans le temps l'exécution de la même tâche par le même composant. Exemple, un nœud répète la transmission d'un message plusieurs fois dans le temps. Selon [Marques *et al.*, 2012], cette famille de redondance est couramment déployé dans les systèmes critiques afin d'assurer l'intégralité de communication. En effet, la transmission du même message est fixé à des instants distincts. Cette transmission répétitive peut être périodique ou évènementielle (l'occurrence d'un évènement particulier qui sollicite l'envoi).
  - redondance spatiale c'est la redondance basique, il s'agit de la duplication d'un même composant. Par contre ce composant peut être logiciel, de données ou matériel (i.e. capteur, actionneur, calculateur, etc).

    Redondance matérielle nécessite la duplication du composant matériel au moins une fois. Cette famille permet d'améliorer le niveau de disponibilité du système en cas de défaillance. Son point faible c'est qu'elle est coûteuse et qu'elle complique l'architecture du système ce qui n'est pas approprié au système embarqué. La redondance matérielle peut être active, passive ou hybride.

    Redondance logicielle En cas de défaillance d'un composant logiciel, cette famille de redondance améliore la SdF du système. Cette redondance augmente le temps de conception, du test. Elle contient deux familles :

    - Redondance logicielle à version unique : duplication de la même version unique du logiciel afin de détecter ou recouvrir et ne pas laisser propager les erreurs.
    - Redondance logicielle multi-versions est basée sur la duplication d'un logiciel avec des copies de différentes versions. Les mécanismes de diversification sont différents langages de programmation, différents compilateurs, différentes équipes de développement, différences dans la conception.

Redondance de données (ou en valeur) Il s'agit de l'ajout d'une partie de données aux données principales. Ces extra-données, appelées aussi "bits de contrôle", sont regroupées dans des blocs séparés des données utiles, comme elles peuvent être insérées dans les données utiles. Ces bits servent à détecter et parfois corriger les erreurs. Ils sont générés par des algorithmes, i.e. code détecteur/correcteur d'erreur, code de parité, code cyclic redundancy check (CRC), les codes de Haming selon [Dorow, 2003].

En se basant sur cette classification, on peut présenter quelques architectures qui intègrent les différents principes de redondances inspirés de la littérature [Dubrova, 2013, Knight, 2012], [Knight, 2012] et [Dorow, 2003]. Il n'est bien sûr pas possible de décrire toutes les architectures existantes.

— Architecture à redondance double se base sur la duplication identique d'un composant en ajoutant un comparateur capable de détecter l'erreur suite à une différence dans les données des composants.

— Architecture à redondance double commutée se base sur le switching entre deux composants dupliqués qui ne fonctionnent pas en parallèle. Le commutateur s'active quand il y a une détection d'erreur, grâce aux techniques de détection supplémentaire intégrée dans le composant principal.

— Architecture N-modulaire c'est la version de la redondance double mais N fois avec N supérieur à 2. Cette architecture intègre un module de vote qui considère la sortie majoritaire correcte en cas de possibilité d'erreur. Cette architecture est souvent utilisée dans les systèmes qui nécessite un haut niveau de fiabilité comme dans les calculateurs des systèmes de commande de vol.

— Architecture hybride basée sur l'architecture N-modulaire avec une redondance dynamique c'est l'analogique de l'architecture double commutée appliquée à N copies. En effet, un système de vote, intégré dans ce type d'architecture déclenche l'activation de K copies (k inférieur à N). Ce type d'architecture est utilisée dans les systèmes critiques à longue durée (domaine de l'espace).

Dans cette partie, nous nous sommes focalisés sur les travaux de [Zammali, 2016]. Ces différents types de redondance sont rarement employés seuls et souvent combinés.

Exemple : la transmission de messages peut être répétée (redondance temporelle) via des canaux différents (redondance spatiale matérielle). Dans le cadre de notre travail, nous avons déployé une couche safety capable de détecter les différentes défaillances dans la communication, comme alternative à la solution de duplication physique et matérielle. Afin de satisfaire aux besoins liées à la safety dans le réseau, nous avons besoin d'identifier les erreurs qu'on peut avoir dans ce type de communication. Dans cette partie nous allons décrire une analyse de deux normes IEC 61508 et IEC 61784 du point de vue du réseau afin de concevoir ou d'implémenter des mesures de safety dans le système de communication. Ces mécanismes seront implémentés dans une extra-couche au dessus de la couche application d'un protocole RTE. La couche déployée permet de détecter les défaillances suivantes :

— Séquence erronée associée à la réception d'un message dû à une erreur, faute ou une interférence qui peuvent falsifier l'ordre chronologique de l'information.

— Corruption des données pouvant se produire à cause d'une erreur dans le médium de transmission ou d'une interférence de message mais ce n'est pas facile pour le récepteur de détecter des informations non valides dedans.

— Répétition involontaire : Il s'agit de la répétition d'un ancien message non actualisé au mauvais moment.

— Perte : au moment où un message n'est pas reçu ou n'est pas bien acquitté ce qui cause un manque d'information.

— Délai inacceptable : les messages peuvent arriver en retard de façon qu'ils dépassent la fenêtre fixée pour leur arrivée, i.e. due aux erreurs dans le medium de transmission, congestion dans la transmission, interférences, d'une façon que les informations sont retardées ou refusées côté récepteur (i.e. le message n'est pas transmis dans le temps de tolérance aux fautes).

— Insertion : c'est quand une entité inappropriée ou inconnue tente d'insérer un message (i.e. violation de droit d'accès à la participation de bus).

— Déguisement d'un message non critique sous la forme d'une message safe. les données sont insérées sous label critique et passent comme safe d'une source valide et donc vue comme participant safe alors qu'en vérité ce nœud doit envoyer des informations non critiques. Le nœud émetteur va se prendre pour un nœud safe permanent or ce n'est pas le cas puisque il peut envoyer des données non critiques.

— Adressage : Un message safe permanent est envoyé au faux nœud safe alors que ce dernier va considérer la réception comme correcte.

Pour faire face à certaines défaillances [Novak *et al.*, 2007] propose quelques mesures de sécurité pour :

— Corruption des données : utilisation d'un CRC ou une duplication de données avec comparaison.

— Perte d'un message : utilisation d'un watchdog.

— Insertion d'un message : l'utilisation du "time-stamp" et les adresses safe.

— Délai, répétition et séquence erronée : utilisation d'un time-stamp.

— Déguisement du message (unsafe Vs safe) : l'utilisation d'une entête spécifique à travers un modèle d'adressage spécifique.

4.4.2 Concept de la couche safety

Selon l'IEC 61508, il existe deux approches de safety totalement différentes :

— Le canal blanc (white channel) : consiste à vérifier l'intégrité de chaque couche du modèle OSI et s'intéresse aussi au hardware impliqué dans la transmission des données comme le montre la figure 4.16.a.

— Le canal noir (black channel) : ce concept, basé sur une couche safety, se positionne au dessus de la couche application du modèle OSI. Cette couche se base sur les fonctions de safety appropriées à la transmission des données comme l'indique la figure 4.16.b. En effet, elle fait une abstraction pour toutes les couches sous-jacentes du modèle OSI indépendamment de la manière de concevoir le système de transmission (sans avoir besoin de connaitre des détails/spécification de la façon dont les couches sous-jacentes sont construites).

Dans le cadre de ce travail, le système de transmission entre deux nœuds safes (communication safe) est basé sur le principe de canal noir (en analogie avec la boîte noire). Une communication safe dans notre réseau déployé nécessite :

— L'information doit être correcte au bon endroit et dans le bon ordre.

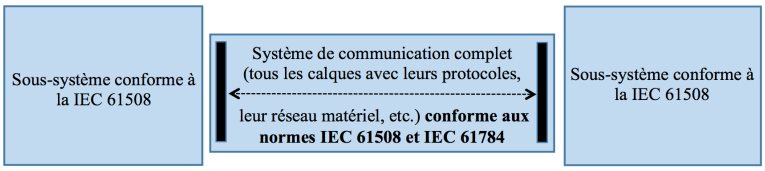


Figure a: Principe du canal blanc

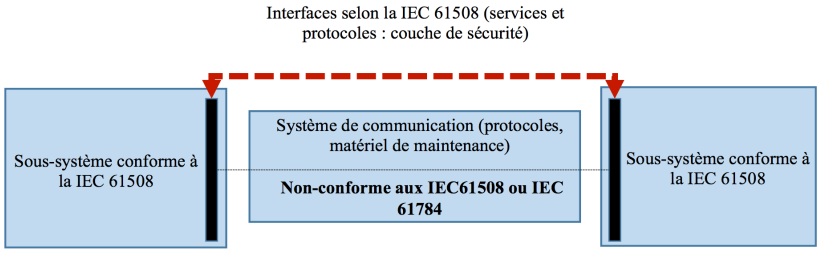


Figure b : Principe du canal noir

FIGURE 4.16 – Les deux concepts white Vs. black channel selon l'IEC 61508

— L'information doit être disponible au bon moment.

— Les communications safe et non safe qui circulent sur le même canal doivent être indépendantes en permettant l'utilisation de nœuds safes et standards.

4.4.3 Mise en œuvre de la couche safety et de ses mesures

Afin de remédier aux erreurs décrites auparavant, notre stack safety implémente les mesures nécessaires afin de détecter et d'éviter de passer aux états unsafe (sachant que, selon l'IEC 61508, un état safe est une situation qui peut générer un risque élevé pour le système). Pour enrichir les mesures de safety proposées par [Novak and Tamandl, 2007], nous ajoutons quelques techniques supplémentaires qui respectent l'IEC 61508 et l'IEC 61784.

— Numéro de séquence : Un numéro de séquence est intégré dans un message échangé entre la source et la destination. Cette mesure permet de détecter la retransmission, la perte, l'insertion d'une séquence erronée.

— Time-stamp : dans la plupart des cas, le contenu d'un message n'est valide que pendant un temps particulier. Le time-stamp peut être un temps ou bien un temps et une date inclus dans le message par l'émetteur. Cette mesure permet de détecter la répétition, la séquence erronée, un délai erroné.

— temps d'attente (time expectation) : Durant la transmission, le message vérifie le délai entre deux messages reçus consécutifs qui dépassent une valeur pré-déterminée. Dans ce cas, il faut supposer qu'il y a une erreur. L'erreur détectée est l'erreur du délai de transmission.

— Authentification lors de la connexion : les messages peuvent avoir un unique identificateur pour la source et/ou destination qui décrit l'adresse logique du participant safe et permanent. Cette mesure permet de détecter toute tentative d'insertion d'un message par un émetteur non authentifié.

— Un message de retour ou un acquittement : la destination renvoie un message de retour (confirmation) à la source afin de lui confirmer la réception du message original. Ce message

de retour doit être traité par la couche safety. Ça permet de détecter la perte, l'insertion, les données corrompues, le masquerading des données.

— La redondance avec vérification croisée : Dans les applications safety, les données de safety peuvent être envoyées de deux façons : soit regroupées dans un seul message ou bien séparées en deux messages en utilisant soit une mesure d'intégrité identique ou différente, indépendamment des couches sous jacentes du réseau. Si une différence est détectée, une erreur doit être considérée dans la phase de transmission. Dans les unités de traitement de la source ou de la destination les erreurs qui peuvent être détectées sont la répétition, la perte, l'insertion, la séquence erronée, la corruption des données.

— L'assurance d'intégrité des données : Les applications liées à la safety ne doivent pas faire confiance aux méthodes d'assurance d'intégrité des données si elles ne sont pas conçues d'un point de vue sécurité fonctionnelle. Par la suite, les données redondantes (dupliquées) sont incluses dans un message afin de détecter la corruption des données après vérification. Dans [Storey, 1996] et [Zammali, 2016], les auteurs définissent l'intégrité de données comme la capacité du système à détecter/corriger les données erronées durant l'échange via un réseau.

— Distinction entre les messages du point de vue safety : Il faut distinguer deux types de messages : safety related messages (SR) et non-safety related messages (NSR).

— Protection des données.

Dans nos travaux, [Soury *et al.*, 2015a], nous avons analysé la classification des solutions basées sur l'Ethernet temps réel afin de choisir l'approche qui correspond à notre système (i.e. le système de contrôle d'ascenseur doit être conforme à l'IEC 61508). Nous nous sommes intéressés au système de communication et à la safety de ce système dans notre banc d'essai composé de cartes industrielles. Le système de communication définit les techniques d'accès au medium, les mécanismes de transmission, etc. Pour les applications critiques qui demande de la safety, le protocole de communication doit répondre aux exigences de safety de certaines normes génériques comme l'IEC 61508 grâce à la sécurité fonctionnelle assurée par les méthodes et les mesures qui permettent de détecter les erreurs de transmission.

4.4.4 Réalisation d'un système de communication sécurisée

Pour garantir le niveau de sécurité imposé par la PESSRAL, nous avons proposé un ensemble de fonctions de sécurité, décrites dans la partie précédente, qui sont implémentées dans une surcouche, dite "couche safety" au dessus de la couche application en nous basant sur le concept du canal noir, comme nous l'avons décrit dans la partie 4.4.1. La surcouche des fonctions de sécurité doit se faire au dessus de la couche application du protocole RTE indépendamment du protocole utilisé dans les couches basses (i.e. EPL ou EtherCAT) comme le montre la figure 4.17. Cette surcouche adoptée, bien qu'elle duplique le temps de cycle initial du réseau, permet de diminuer la probabilité d'erreur afin d'atteindre une spécification de niveau SIL3. La duplication du temps de cycle revient à la redondance des données appliquée par le concept canal noir implémenté par la couche safety.

Conformément à la norme PESSRAL, le système GESA doit définir une variable de temps pour modéliser le temps de réaction éventuellement le temps de cycle du système de communication. Il s'agit de l'intervalle de temps entre la détection d'une faute ou d'un état non sûr du système et l'établissement d'un état sûr du système. Cette variable, va servir à évaluer la capacité du système à répondre aux exigences de la PESSRAL. Par exemple, le système GESA doit être capable d'évaluer l'intervalle de temps entre la détection d'une défaillance système (mémoire, réseau, etc.) (voir annexe B) ou la détection du dépassement de la vitesse maximale de la cabine

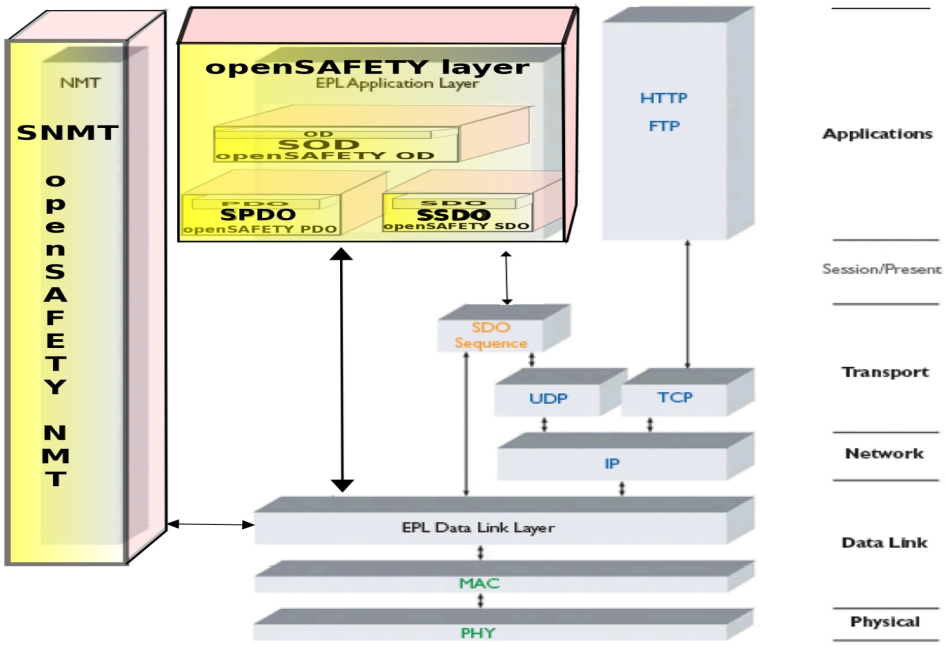


FIGURE 4.17 – Ajout de la couche safety au niveau de la couche application d'un protocole RTE

et l'ouverture de la chaîne de sécurité pour déclencher les freins.

Pour pouvoir identifier ces exigences temporelles, une analyse de la norme PESSRAL dans [Soury *et al.*, 2015b] nous a permis de spécifier chaque exigence imposée par cette norme. En effet, une exigence PESSRAL est composée d'une exigence racine à laquelle sont associées des exigences de raffinement comme le montre la figure 4.18.

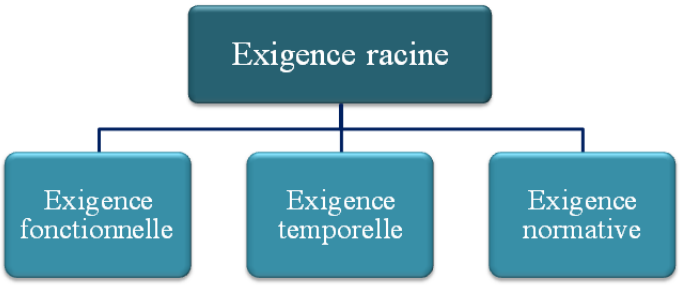


FIGURE 4.18 – Définition d'une exigence.

— Exigence racine : Nous prenons la protection contre la vitesse excessive de la cabine en montée
— Exigence fonctionnelle : un ascenseur doit être pourvu d'un dispositif de protection contre la vitesse excessive de la cabine en montée. Le dispositif, comprenant des organes de contrôle

et de réduction de la vitesse, doit détecter un mouvement incontrôlé de la cabine en montée à une vitesse d'au minimum 115% de la vitesse nominale. Le dispositif doit agir sur : la cabine, ou le contrepoids, ou le système de câbles (de suspension ou de compensation) ou la poulie de traction.

— Exigence temporelle : temps de réaction du système $\leq 100ms$

Cette décomposition des exigences rend possible la validation et la vérification des composants du système GESA à l'aide de l'outil Krono-OS. Les contraintes physiques ont été réduites dans le sens où le démonstrateur ne sera pas placé dans un environnement réel de fonctionnement. Afin d'assurer le niveau de sûreté de la communication sur l'ensemble de la chaîne de sécurité avec un seul canal de communication, nous devons pour cela utiliser un réseau de communication certifié SIL3.

4.4.5 Scénarii de tests

Dans un but de simplification, le réseau doit véhiculer des messages critiques et non critiques. L'intégration de la pile protocolaire sous Krono-OS ne doit pas constituer un trop gros effort d'intégration. En effet, dans le cadre de la PESSRAL, la chaîne de sécurité se traduit par la présence des variables d'état dans le système Kron-OS. Par conséquent, toutes les actions critiques (mouvement de la cabine, fermeture des portes, etc.) sont conditionnées par l'état de ces variables. Avec ce concept, toute action sur le système en général, et sur la chaîne de sécurité en particulier peut être limitée et validée par une durée qui permet d'ouvrir/fermer la chaîne de sécurité, l'élément-clé pour le mouvement de la cabine. La transmission des informations (état de sécurité de la cabine, position de la cabine en remontée, sécurité de la machinerie, état de palière, etc.) doit se faire via une liaison sécurisée pour activer le nouveau système électronique de sécurité. Ce nouveau système doit pouvoir agir, via une communication sécurisée, sur le système de freinage de la cabine afin de l'arrêter mécaniquement et sur la commande du variateur. Les contraintes temporelles peuvent être représentées sous la forme suivante. Le sous système CSC possède un certain nombre de propriétés temporelles. Ces propriétés (valeurs) sont évaluées par des équations associées à des contraintes. Suite à une action sur l'ascenseur (exemple : dépassement de la vitesse maximale, figure 4.19), le temps de réaction du système à l'intervalle de temps entre la détection d'une faute ou d'un état non sûr du système (détection de l'excès de vitesse) et l'établissement d'un état sûr du système (activation du système de freinage) ce qui correspond à $\Delta = T_7 - T_0 = 100\ ms$. Les contraintes sur le système GESA imposées par la PESSRAL s'appliquent à l'intervalle $\Delta_{GESA} = T_4 - T_3$. Cet intervalle de temps ne doit en aucun cas dépasser 50 ms. Ce qui est le cas comme le montre la figure 4.19 (tirée du document interne du projet) obtenue grâce à l'outil Krono-OS. Avec les performances que nous avons pu évaluer

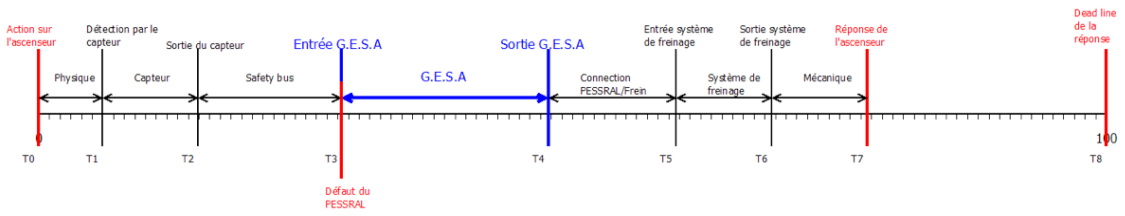


FIGURE 4.19 – Découpage temporel du temps de réaction du système GESA [Document interne Krono-safe].

sur ce protocole, nous avons réussi à valider notre approche d'EPL sécurisé avec nos partenaires

du projet. Un ensemble de tests ont été exécutés sur le système GESA intégrant le modèle de communication sécurisé à base d'EPL. Ces tests se sont basés sur l'évaluation de la réaction du système quand une fonction de sécurité est appelée. Ils se sont déroulés sur l'outil Krono-OS. Par exemple, les deux fonctions suivantes (fonction de sécurité du mouvement incontrôlé de la cabine et fonction de protection contre la vitesse excessive de la cabine en montée) ont été testées avec succès.

Fonction de sécurité du mouvement incontrôlé de la cabine : Si la cabine se déplace sans aucun ordre de commande (les portes palières ne sont pas en position verrouillée et les portes de cabine ne sont pas en position fermée), il faut l'arrêter avant de dépasser 1m20 du palier. Les ascenseurs doivent être équipés d'un dispositif permettant de stopper tout mouvement incontrôlé de la cabine lorsque cette dernière s'éloigne de l'étage. Ce mouvement résulte d'une défaillance d'un élément quelconque de la machine ou du système de commande de l'entraînement dont dépend la sûreté des mouvements de la cabine, à l'exception de toute défaillance des câbles ou chaînes de suspension et des poulies de traction ou tambours ou pignons de la machine. L'auto-surveillance fait l'objet d'un examen de type. En effet, le "safety car" contrôle en permanence sa position. Tout déplacement de la cabine avec les portes cabines ouvertes enclenche l'arrêt de la commande de la bobine parking. Afin de ne pas enclencher le parachute de façon inopinée, une "zone non enclenchement" empêche l'enclenchement de la sécurité sur un simple étirement des câbles. En cas d'"isonivelage", l'enclenchement s'effectuera en sortie de "zone de porte". Sous une contrainte temporelle, le système doit réagir dans un temps de réaction = 100 ms (à 1m/s un ascenseur parcourt 1 cm en 100 ms). Nous avons un temps de réaction maximum de 40 ms avant de déclencher le frein (temps nécessaire pour la synchronisation des modules et mise à jour des variables).

Fonction de protection contre la vitesse excessive de la cabine en montée : Un ascenseur doit être pourvu d'un dispositif de protection contre la vitesse excessive de la cabine en montée. Le dispositif, comprenant des organes de contrôle et de réduction de la vitesse, doit détecter un mouvement incontrôlé de la cabine en montée à une vitesse d'au minimum 115 % de la vitesse nominale. Le dispositif doit agir sur la cabine, le contrepoids, le système de câbles (de suspension ou de compensation) et la poulie de traction. Le temps de réaction système = 100 ms. Le système de communication sécurisé utilisé dans le GESA doit respecter les conditions de sécurité exigées par la norme PESSRAL avec des conditions d'arrêt inférieures à 0,48 secondes (pour une vitesse de cabine de 2,5m/s).

4.5 Conclusion

Ce chapitre a été consacré à la comparaison des performances du système de communication ainsi qu'à sa validation après avoir déployé les protocoles sur les cibles physiques. La première partie du chapitre a exposé le modèle existant de système de contrôle d'ascenseur en identifiant les limites de ce système. Pour suivre l'évolution dans le domaine de la SdF, la PESSRAL exige de remplacer les systèmes de sécurité électromécaniques par des systèmes électroniques programmables qui s'appelle GESA. Nous avons décrit l'architecture de ce système en mettant en exergue sa capacité d'intégrer un système de communication temps réel et sécurisé. Dans ce cadre, nous avons implémenté les deux modèles qui ont été testés théoriquement dans le chapitre 3. Le contexte industriel nous impose d'avoir une réalisation moins coûteuse et plus performant en temps réel. Nous avons réussi à implémenter, tester et évaluer le protocole EPL sur un banc d'essai. Malgré les valeurs optimistes que nous avons pu obtenir d'un réseau EtherCAT, nous nous

posons des questions sur le coût économique d'une implémentation réelle du protocole EtherCAT. En réalité c'est un protocole exigeant et coûteux en termes de réalisation. Par conséquent, nous nous sommes basés sur son comportement simulé. À la fin du chapitre, nous avons expliqué en résumé comment s'est déroulé la phase de validation du protocole EPL. En effet, bien qu'il ne soit pas le meilleur en terme de performances temporelles par rapport à l'EtherCAT, il répond bien aux contraintes du système de contrôle d'ascenseur. C'est donc le protocole EPL avec la couche safety proposée qui ont été retenus pour le démonstrateur de ce projet.

Conclusion

L'objectif des travaux de thèse présentés dans ce mémoire est de proposer un modèle de communication numérique et sécurisée d'une application industrielle, répondant aux exigences SIL 3 de la norme IEC 61508. Ces exigences sont surtout d'ordre fonctionnel, c'est à dire relatives à la sûreté de fonctionnement du système, et ce, du fait du risque qu'implique l'occurrence de défaillances dans ce contexte, des évènements catastrophiques, voire la perte de vies humaines. Du fait même de ce risque, des normes et des standards de certification ont été mis en place afin de dicter les exigences de sûreté à respecter par les systèmes. À cela s'ajoute le fait que ces normes de certification ont même, dans certains domaines, une autorité légale (autorité de certification) qui autorise ou non, au final, la mise en service du système en évaluant son niveau de sûreté. Dans la première partie de la thèse, nous avons introduit une architecture numérique commandée pour assurer la sécurité de l'application électromécanique, cet aspect a nécessité une analyse approfondie de la sécurité fonctionnelle dans les systèmes de contrôle commande. En outre, nous avons identifié les différents standards de la sûreté de fonctionnement qui décrivent les exigences de sécurité liées à chaque domaine et recommandent les mesures nécessaires. Cette analyse passe par un démonstrateur D2 qui a pour objectif global de concevoir et développer les fonctions de sécurité d'un ascenseur par des systèmes électroniques. Ces derniers utilisent un noyau temps réel déterministe avec un objectif de certification vis-à-vis de la norme PESSRAL.

Dans un premier lieu, nous avons décidé d'intégrer une communication numérique sur l'ensemble du système cible, ce point a nécessité la prise en considération des résultats de la première partie de notre analyse. Autrement dit, l'approche proposée doit répondre aux exigences de SdF qui sont à la fois normatives (spécifiées par des normes comme l'IEC 61508) et légales (imposées par des lois et des règlements). Cette proposition a servi à passer des systèmes électromécaniques aux systèmes communicants plus intelligents et donc de suivre l'évolution technologique des systèmes embarqués. À cet effet, nous avons commencé par l'étude des réseaux de communication industrielle qui repose actuellement sur des bus dédiés dont nous avons évalué la capacité à répondre aux exigences de la SdF.

Ensuite, nous avons réduit notre état de l'art au bus CAN. Cette réduction trouve tout son intérêt dans le cadre de notre projet relatif à l'ascenseur. Elle permet d'adoucir la transition du système lors de son évolution et de faciliter la tâche d'intégration avec les partenaires du projet. En effet, les ascenseurs actuels intègrent un profil de la famille CAN qui leur est dédié : le CANOpen-Lift. Dans cette partie nous avons présenté un cas de transition de la transmission analogique du système de contrôle commande à la transmission numérique. Pour améliorer cette architecture, nous avons identifié les exigences fonctionnelles du système d'ascenseur en nous basant sur les travaux de Limal ([Limal *et al.*, 2007, Limal, 2009]), concernant le déterminisme, la réactivité et la disponibilité. À travers l'étude de ces exigences, nous sommes parvenus à définir les propriétés du réseau d'un système temps réel critique. Ainsi, nous avons décrit les critères de sûreté de fonctionnement liés aux réseaux industriels et étudié les caractéristiques des réseaux déployés actuellement dans les ascenseurs français. Ces réseaux sont largement utilisés par les

trois entreprises françaises de construction d'ascenseur ; Sprinte, Sodimas, Schindler. Cependant ils ont montré leurs limites à l'égard des nouvelles exigences de SdF et des besoins d'évolution. Notre étude s'est particulièrement intéressée aux réseaux de type CAN vu les fonctionnalités propres aux systèmes de contrôle d'ascenseur que propose son bus. Par la suite, nous avons introduit notre contribution par une description des réseaux à base d'Ethernet pour répondre aux exigences fonctionnelles des systèmes industriels. Ce choix peut être justifié par l'évolution accélérée des réseaux de terrain vers la technologie IP.

Nous avons, ensuite, présenté les approches Ethernet temps-réel en nous basant sur le facteur temps de cycle qui a servi à les organiser en 3 classes. Cette classification nous a permis de réduire les protocoles candidats, puisqu'elle est basée sur la capacité du standard Ethernet classique à répondre aux exigences de déterminisme et de réactivité. En ajoutant l'héritage du bus CAN, notre choix s'est porté vers le protocole EPL. Il s'agit, donc, d'une intégration du bus CAN sur le standard Ethernet. Ce point a rendu notre approche capable de répondre aux exigences fonctionnelles propres aux systèmes embarqués temps réel.

Après, nous avons décrit les spécifications de deux protocoles à base d'Ethernet (EPL et EtherCAT). La première partie a été consacrée à l'EPL dont nous avons analysé les différents paramètres de temps de cycle. Cette analyse a rendu la vérification fonctionnelle théorique d'EPL possible grâce aux équations qui décrivent son comportement temporel. En fait, les spécifications de ce comportement définissent une phase asynchrone obligatoire pour chaque cycle d'EPL. Cette phase asynchrone représente le facteur principal qui affecte et augmente directement ce cycle. Malgré sa gérance par le nœud maître du réseau, elle reste une phase gourmande en terme de temps. Pour faire face à cet inconvénient, nous avons décidé d'agir sur les timeouts qui sont, heureusement, configurables lors de la phase pré-opérationnelle du protocole. Par ailleurs, la phase qui assure l'échange des données de contrôle est la phase isochrone. Durant cette phase, le maître, collecteur des données, interroge ses esclaves à travers des messages séparés séquentiels. Ceci dans le but de faire communiquer l'état de chaque nœud esclave. Grâce à cette étude quantitative du temps de cycle, nous avons pu réaliser des expérimentations sur des cibles physiques de type STM32. Elles nous ont permis de vérifier les capacités fonctionnelles et temporelles du protocole EPL. La même démarche d'évaluation a été suivie pour le protocole EtherCAT. Théoriquement nous avons réussi à identifier les différents paramètres qui affectent directement le temps de cycle de la communication. Cette analyse permets de borner le temps de cycle du protocole dès la phase de configuration, afin de satisfaire l'exigence temporelle du système. Elle permet ainsi de prendre en considération le temps de réaction du système en tant que contrainte normative majeure. Par ailleurs, pour renforcer la "safety" dans la communication offerte par l'un de ces deux protocoles, nous avons proposé d'introduire le concept du canal noir sur le modèle de communication. Dans ce contexte, l'utilisation du standard Ethernet a rendu la tâche plus facile. Il s'agit de faire une abstraction des couches sous-jacentes et de rajouter une extra-couche au niveau applicatif du modèle OSI. Cette couche contient toutes les mesures de sécurité qui répondent aux exigences de la norme IEC 61508 liées à la communication. Son étude théorique a permis de limiter l'ensemble de fonctions de sécurité développés sur le modèle de communication utilisé (indépendamment du protocole). Mais l'avantage majeur de cette couche safety reste sa capacité à s'adapter à tout protocole reposant sur l'Ethernet dans les couches sous-jacentes (EPL ou EtherCAT). Et ce, du fait qu'elle réalise ses fonctions de sécurité au niveau très haut, le niveau applicatif. Dans la dernière partie, nous avons présenté le contexte industriel du projet qui constitue un cas de transition d'un système de contrôle d'ascenseur. Cette transition s'intéresse essentiellement aux systèmes de sécurité de l'ascenseur. Il s'agit de faire remplacer l'ancien système de sécurité basé sur des composants électromécaniques par un système électronique programmable en assurant le même niveau de sécurité. Ce nouveau système GESA (défini dans la PESSRAL) contient deux

composantes principales : un système de contrôle sécurité machine (CSM) qui assure la sécurité en machinerie et un système de contrôle sécurité cabine (CSC). Le système GESA intègre un système de communication qui assure la surveillance mutuelle des deux sous système CSC et CSM. Les données échangées dans le système GESA doivent respecter le format CANOpen-Lift. Dans ce nouveau système de sécurité, nous avons conçu l'intégration d'un système de communication temps réel dans le GESA. Le protocole EPL hérite du protocole CAN toutes ses spécifications et les implémente sur le standard Ethernet. Il constitue, ainsi, le protocole potentiel pour la mise en œuvre des interfaces de communication Ethernet temps réel. Nous avons trouvé une différence importante entre le temps de cycle théorique, déjà calculé, et les valeurs trouvées lors de sa mise en œuvre sur les cibles STM32. Après avoir pris contact avec le concepteur du protocole EPL (B&R), il s'est avéré que cette incompatibilité entre les valeurs revient aux capacités des systèmes embarqués utilisés lors de l'évaluation temporelle du protocole. En effet, des latences liées aux propriétés physiques des composants qui varient en fonction de constructeurs s'ajoutent au temps de cycle mesuré. Pour renforcer la sécurité du système de communication, nous avons implémenté la couche safety. Cependant, la mise en œuvre de ces mesures a nécessité la duplication du temps de cycle déjà calculé (EPL et EtherCAT) tout en respectant les exigences normatives. Malgré la duplication des performances temporelles, notre modèle de communication sécurisée répond convenablement aux exigences normatives. Cette surcouche qui exige une duplication du temps de cycle, justifie notre analyse raffinée du comportement temporel de chaque protocole. Le principe était de minimiser le temps de cycle d'une communication non sécurisée pour pouvoir proposer un modèle de communication intégrant les fonctions de sécurité. Ce modèle de communication sécurisée a été accepté et validé par nos partenaires du projet. En effet, l'outil Krono-OS, conçu par krono-safe, le porteur de projet, a permis de faciliter la tâche de validation de ce choix et ce, grâce aux scénarii décrits par les spécialistes d'ascenseur.

La généricité de ces modèles fait que leur mise en œuvre est possible sur d'autres applications. En effet, une évaluation des deux modèles de communication proposés, sur des cibles physiques, serait de grand intérêt.

Nous étions motivés pour évaluer le protocole EtherCAT sur un autre démonstrateur (D3) du projet ADN4SE. Ce démonstrateur avait pour objectif de concevoir un robot collaboratif intégrant la communication temps réel. Les exigences temporelles du robot nous apparaissent plus strictes que celles de l'ascenseur et donc nous aurions aimé continuer à tester les fonctionnalités et évaluer réellement les performances du protocole EtherCAT sur le démonstrateur du robot. Malheureusement la divergence au niveau du choix physique entre les partenaires n'a pas permis d'aller jusqu'à la réalisation. En effet, avant notre arrivée, les partenaires du démonstrateur se sont engagés dans une solution matérielle et logicielle payante (EtherCAT classe A). Par ailleurs, nous n'avons pas pu suivre la direction prise par le CEA, premier responsable du démonstrateur robotique, car il avait déjà fait son choix matériel avec d'autres partenaires avant notre collaboration. Mais le concept de sécurité que nous avons proposé, basé sur le principe du canal noir, a bien trouvé sa place auprès de l'entreprise ISYBOT, concepteur et fabricant des bras robotiques collaboratifs (une spin-off de CEA LIST issue du projet ADN4SE).

En termes de perspectives, nous pouvons envisager une collaboration avec l'entreprise afin de vérifier ce concept et le valider à travers le "model cheking" [Thiele *et al.*, 2016]. Une deuxième perspective serait de s'appuyer sur les propriétés de l'outil krono-OS, classifié sûr de fonctionnement par construction, pour une intégration complète de l'EtherCAT classe B (version gratuite mais modeste en terme de performances) afin d'atteindre les mêmes performances de la classe A (payante) sans l'ajout de composants physiques. Cette perspective sera étudiée prochainement pour vérifier la possibilité de classer l'EtherCAT classe B avec l'outil krono-OS dans la classe 3 des solutions RTE.

Pour évaluer la généricité du concept de sécurité proposé dans ce manuscrit de thèse, nous comptons évaluer notre approche de sécurité sur une communication de type "time-triggered Ethernet". En effet, le protocole "TTEthernet" (Time-Triggered Ethernet en anglais) semble être une piste à creuser après son apparition dans la littérature [Boyer *et al.*, 2016]. En outre, récemment des technologies basées sur l'Ethernet sont venues pour répondre à la fois aux exigences de déterminisme et de tolérance aux fautes dans les réseaux industriels [Wollschlaeger *et al.*, 2017], [Gavrilut *et al.*, 2017], [Bush *et al.*, 2017]. Ces technologies, dites "TSN" (Time-Sensitive Networking en anglais) qui sont adaptées aux applications industrielles critiques devront être comparées à notre approche.

Annexe A

Configuration d'un réseau EPL

Dictionnaire d'objets pour l'EPL

Au niveau de la couche applicative, EPL utilise le mécanisme de dictionnaire d'objets pour assurer l'échange de données entre les différents nœuds du réseau. Ce mécanisme est hérité du protocole CANOpen utilisé avec CAN. Un dictionnaire d'objets est une structure virtuelle permettant l'accès aux données offertes par un nœud et pouvant transiter dans le réseau, à savoir des données critiques, des paramètres de configuration ou bien des informations du périphérique, ces données sont accessibles en mode lecture/écriture. Le dictionnaire d'objets est le cœur du nœud. Ce dernier doit avoir un et un seul dictionnaire d'objets. Il s'agit de l'interface de communication et d'échange de données entre les éléments du réseau. Afin de pouvoir envoyer un objet, l'application doit obligatoirement passer par le dictionnaire d'objets pour récupérer les données à transmettre et qui seront par la suite passées vers les couches inférieures du protocole EPL pour assurer l'envoi. La configuration des paramètres de la pile protocolaire est la clé pour permettre au réseau de transmettre des informations conformément au contrat d'utilisation. En outre, ces paramètres sont configurables "off-line". Ils sont accessibles via l'index du dictionnaire d'objets.

Par exemple, pour contrôler le comportement temporel du trafic réseau EPL, nous stockons notre configuration dans l'index d'objet 1006_h. Cet objet définit l'intervalle du temps de cycle en micro seconde. L'objet $1F9C_h$ affecte à un nœud un créneau isochrone particulier et l'objet $1F8B_h$ définit la taille de données de charge utile PReq en octet pour chaque nœud configuré. L'objet $1F8D_h$ configure les données de charge utile PRes.

Le dictionnaire d'objets structure les entrées selon des plages d'index ordonnées et bien déterminées. Chaque objet du dictionnaire est adressé par un index de 16 bit (de 0000_h jusqu'à $FFFF_h$). Ce dictionnaire d'objet peut contenir au maximum 65536 (2^{16}) entrées. Le profil de communication avec l'indice entre 1000_h et $1FFF_h$ contient les paramètres spécifiques à la communication EPL. Ces entrées sont communes pour tous les nœuds. L'EPL propose trois types d'objets qui sont :

- les Service Data Object (SDO) qui sont utilisés pour transférer des données non critiques. Généralement elles sont liées à la configuration du réseau pendant la phase asynchrone et
- les Network Management (NMT) : qui sont les objets échangés pendant la phase asynchrone pour transférer des informations concernant l'état des nœuds du réseau,
- les process data object (PDO) : qui sont utilisés pour le transfert de données critiques en temps réel pendant la phase isochrone. Il existe deux types de PDO :
  - les Transmitted Process Data Objects (TPDO) qui sont configurés au niveau du nœud

émetteur et stockés dans les indices d'objet de 1800_h à $1AFF_h$. Bien que, dans le nœud récepteur, il y a les Received Process Data Objects (RPDO),

— il y ait les objets de données de processus reçus RPDO stockés dans l'indice d'objet de 1400_h à $16FF_h$ dans le nœud récepteur.

Afin de pouvoir assurer l'échange des PDOs pendant la phase isochrone entre le MN et les CNs du réseau, il est obligatoire de préciser les types des PDOs (TPDO ou RPDO) ainsi que leurs indices dans les dictionnaires d'objets, cette configuration est appelée "mapping" de PDOs et elle définit les objets qui vont être échangés pendant la phase isochrone. Pour réduire le temps de cycle, les utilisateurs doivent accorder beaucoup plus d'attention à la définition de ces indices d'objets (taille de la charge utile, délai d'attente de trame, etc.) lors de déploiement du protocole.

Annexe B

Les fonctions de protection critiques et analyse des modes de défaillance

Pour pouvoir répondre aux exigences de la PESSRAL, nous définissons les états de sécurité de l'ascenseur lorsque :

— le système est en mode nominal, l'ensemble des composants (matériel et logiciel) du système se trouvent dans un état de fonctionnement nominal c'est à dire non dégradé.

— le système est en mode sécurisé. Ce mode correspond à une coupure de l'alimentation du moteur et au déclenchement des freins. Cet état fait suite à une défaillance d'un des composants du système déjà décrite précédemment.

La figure B.1 représente une vue globale de l'architecture d'un système électronique de sécurité pour l'ascenseur, le système GESA.

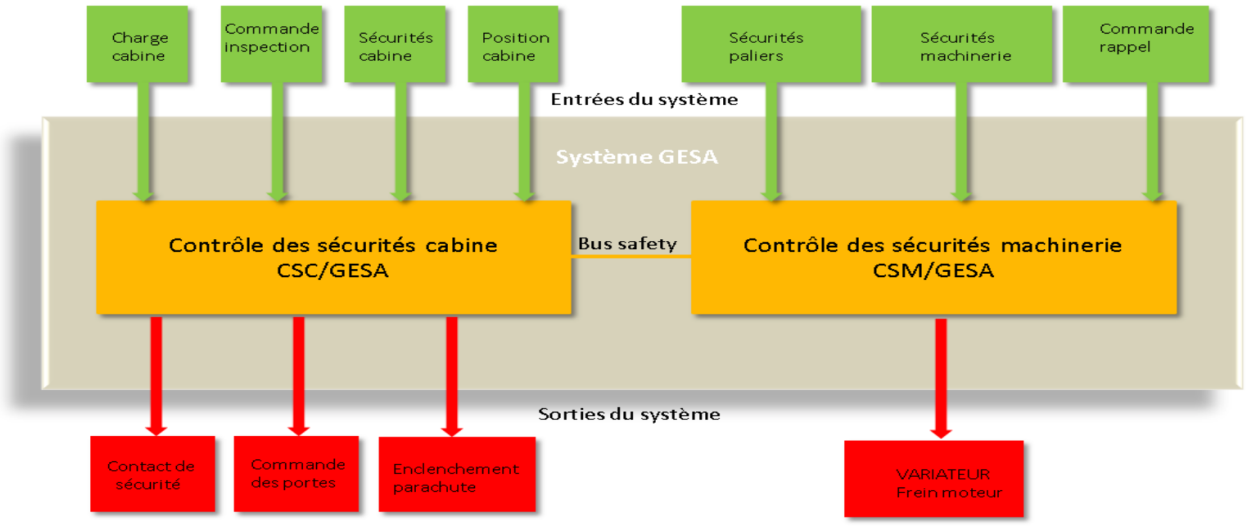


FIGURE B.1 – Vue globale du système de gestion électronique des sécurités ascenseur : GESA.

Fonctions de protection critiques

Pour conserver un état de sortie du système non critique contre plusieurs scénarios, la PESSRAL exige plusieurs fonctions de protection. Nous allons en décrire quelques une :

- La PESSRAL exige de couper (ouvrir) la chaîne de sécurité et le déclenchement du système de freinage dans un temps bien précis afin :
  - d'assurer la sécurité du mouvement incontrôlé de la cabine,
  - de protéger la cabine en montée contre la vitesse excessive et
  - de protéger la cabine contre la chute libre.
- Assurer la sécurité de portes est nécessaire pour maintenir la cabine à un niveau sûr.
- Afin d'autoriser le déplacement de la cabine en inspection en gardant une réserve réduite, la PESSRAL exige l'enclenchement du système de freinage avec une coupure de la chaîne de sécurité en mode normal.
- Pour ralentir le déplacement de la cabine et assurer la fin de course aux extrêmes, il faut couper la chaîne de fin de course et forcer la vitesse réduite.

Le plan de validation globale de la sécurité du système E/E/PE relatif à l'ascenseur se base sur le réseau de communication. En effet, les tests de validation des fonctions de protection sont réalisés par le système PES via le réseau sécurisé. Il s'agit des tests fonctionnels et des tests de mode de défaillance ainsi que des tests pour les arrêts du système. Les résultats du test décrivent les états des composants suivants :

- Chaîne de sécurité,
- Système de freinage,
- Moteur.

```
si État portes est ouvertes alors
    si "isonivelage" est "en cours" alors
        si "position cabine" est hors "zone de porte" alors
            "ouverture chaîne de sécurité" et "enclenchement du système de freinage"
            sinon si "position cabine" est hors "zone non enclenchement" alors
                "ouverture de la chaîne de sécurité" et "enclenchement système de freinage"
            fin
        fin
    fin
fin
```

Algorithme 1 : Fonction de sécurité : Sécurité mouvement incontrôlé de la cabine.

Fonctions non critiques

Les fonctions non critiques se voient allouer un niveau d'intégrité de sécurité SIL 0. Le système doit assurer que ces fonctions ne dégradent pas le niveau d'intégrité de sécurité des fonctions critiques. Pour ce faire, une modularisation des fonctions parait indispensable en s'appuyant sur les capacités de cloisonnement du système d'exploitation (OS). Le réseau EPL doit assurer l'accès/transmission de ces fonctions.

Fonction de log interne au PES : Le système GESA conserve l'historique de log avec une profondeur de un an. L'accès aux logs doit se faire via le réseau sécurisé. Cette fonction étant non critique, elle est exécutée uniquement lorsque le système est disponible. Chaque log doit posséder un niveau de criticité (fatal, error, warning, notification, debug). Il y a une file de log par niveau de criticité (ceci afin de minimiser l'écrasement d'un log de criticité élevé). La file de log doit être cyclique. Si la liste est pleine le log le plus récent écrase le log le plus ancien. Ce service de log doit être disponible dès le démarrage du système avant tout autre service. Chaque log doit être horodaté à la seconde.

Fonction de gestion du mode debug : Le système GESA bascule en mode debug via une commande transmise par le réseau sécurisé. Ce mode permet la remontée de traces via le réseau. Les informations remontées sont ; les traces de debug, l'état du système. L'activation/désactivation du mode debug est assurée par le biais d'une commande transmise via le réseau.

Fonction du mode autodiagnostique : Le mode autodiagnostique a pour but de tester le système. À savoir des tests mémoire et cpu sont réalisés. Le résultat du test est sauvegardé dans la table de notification (log de criticité notification).

Analyse des modes de défaillance

Les organes essentiels du périmètre sont représentés par des fonctions de sécurité tirées de la norme PESSRAL ainsi qu'au cahier des charges de l'ascenseur. Pour analyser ces modes de défaillances, nous faisons référence au tableau des composants, table B.1. Dans ce tableau, nous avons pu identifier un ensemble restreint de composants qui ont une relation directe avec le modèle de communication que nous avons proposé.

Composant	Description
Bus de communication	Composant permettant la communication entre le calculateur cabine et le calculateur machinerie.
Calculateur cabine (CSC)	Assure l'exécution des fonctions de protections relatives à la sécurité cabine.
Calculateur machinerie (CSM)	Assure l'exécution des fonctions de protections relatives à la sécurité machinerie.
Codeur	Mesure la position de la cabine dans la gaine.
Frein	Système de freinage associé au moteur permettant d'assurer l'arrêt de la cabine en cas d'urgence.
Moteur	Permet de faire monter ou descendre la cabine dans la gaine à l'aide d'un câble.
Variateur de vitesse	Dispositif permettant la régulation de la puissance du moteur.

Table B.1 – Liste des composants matériels de l'ascenseur.

Modes de défaillance (AMDE)

L'AMDE a pour but de mettre en évidence l'ensemble des défaillances possibles, de leur effet et de leur criticité. Pour s'assurer que l'élément sous-contrôle (l'Element Under Control (EUC) en anglais) qui est la cabine, se trouve toujours dans un état sûr, il s'est avéré qu'une identification

de l'ensemble de défaillances aléatoires possibles (cause/conséquence) est obligatoire. En effet, nous pouvons identifier 3 principaux modes de défaillance :

— démarrage ou arrêt impossible,
— fonctionnement intempestif du système,
— fonctionnement dégradé.

Ces défaillances touchent des composants matériels et des parties logicielles. Il nous apparaît logique que la prise en charge de chaque défaillance puisse être réalisée par un choix matériel en ajoutant/modifiant un composant/architecture du système ou par un choix logiciel en faisant une comparaison des états ou en ajoutant des variables pour surveiller le fonctionnement d'un composant (i.e. signe de vie). Cette identification de défaillances n'est pas exhaustive. Elle se base sur 3 critères d'évaluation de criticité (probabilité d'apparition, gravité, probabilité de non détection).

Démarrage impossible du système

Le système de contrôle d'ascenseur peut rencontrer un démarrage impossible à cause d'une dégradation qui touche généralement une de ses fonctions de sécurité telle que :

— la fonction qui protège le mouvement incontrôlé de la cabine,
— la fonction qui protège la cabine en montée contre la vitesse excessive,
— la fonction qui protège la cabine contre la chute libre,
— la fonction qui force de réduire la vitesse pour ralentir le mouvement de la cabine aux extrêmes.

Ce démarrage impossible est dû à la défaillance de plusieurs composants tels que :

— le calculateur cabine,
— le codeur,
— le moteur.

Le démarrage impossible du calculateur cabine (CSC et CSM) provoque un traitement impossible des données du codeur. Ce qui rend l'action sur le frein et le variateur de vitesse impossible. Pour faire face à cette défaillance, il faut, ouvrir la chaîne de sécurité, arrêter le moteur et faire déclencher les freins. Comme mesure de sécurité, nous devons intégrer une surveillance du CSC et du CSM via le bus safety de communication. En effet, la perte de la fonction du **codeur** cause un démarrage impossible du composant. Par conséquent, il n'est plus possible de connaitre la position de la cabine. Cette défaillance engendre une dégradation qui touche particulièrement la fonction qui assure la sécurité du mouvement incontrôlé de la cabine. La prise en charge de cette défaillance est assurée par la fonction de protection contre la chute libre en ouvrant la chaîne de sécurité avec un enclenchement du système de freinage et un arrêt immédiat du moteur.

Ainsi, la perte de la fonction du **moteur** rend impossible le mouvement de la cabine. Ceci engendre une difficulté voire impossibilité d'arrêter la cabine par le biais du moteur. Cette défaillance touche directement la commande de montée/descente de la cabine. L'exécution de cette dernière devient impossible. L'ouverture de la chaîne de sécurité avec l'enclenchement des freins parait indispensable voire obligatoire dans ce cas. Ceci nécessite d'intervenir sur le variateur de vitesse. Il doit permettre de remonter l'état du moteur.

La coupure du câble, supporteur du bus de communication, rend impossible de récupérer la position de la cabine à cause de la perte de la fonction du bus. Cette défaillance touche toutes les fonctions de sécurité déjà citées, ce qui nécessite une ouverture de la chaîne de sécurité, un arrêt immédiat du moteur et un déclenchement des freins.

Fonctionnement intempestif du composant

Le fonctionnement intempestif d'un des composants suivants a un impact direct sur le contrôle du système.

— codeur,

— moteur,

— variateur de vitesse,

— calculateur cabine.

En effet, côté mécanique, le fonctionnement erratique du **moteur**, responsable du déplacement de la cabine, provoque un mouvement erratique de cette dernière. Ce dysfonctionnement nécessite le déclenchement des freins ainsi qu'à l'ouverture de la chaîne de sécurité. Comme mesure de sécurité, le variateur de vitesse doit permettre de remonter l'état du moteur. Par contre ce variateur de vitesse, composant indispensable pour la régulation de la puissance du moteur, peut avoir lui aussi un fonctionnement intempestif et qui aura une conséquence directe sur le mouvement de la cabine qui deviendrait erratique et sur le comportement du moteur. Comme mesure de sécurité, nous devons agir sur la chaîne de sécurité et déclencher les freins afin de maintenir la cabine au niveau.

Du point de vue traitement des données, le fonctionnement intempestif du **codeur** provoque l'incertitude de la position de la cabine. Ce type de défaillance oblige le déclenchement de frein avec l'ouverture de la chaîne de sécurité afin de minimiser la dégradation de la fonction qui assure la sécurité de la cabine. Dans ce cas il faut exécuter la fonction, qui protège contre la chute libre afin d'éviter les conséquences lourdes. Cette défaillance peut toucher le traitement des données assuré par le calculateur cabine. En effet, le fonctionnement intempestif de calculateur dégrade, pratiquement, toutes les fonctions de sécurité du système ce qui entraîne l'ouverture de la chaîne de sécurité. La surveillance mutuelle du CSC et CSM via le bus de communication permet de prendre en charge cette anomalie. Le fonctionnement intempestif du bus de communication rend la récupération de la position cabine non assurée. Le système doit réagir directement sur les freins. En effet, il doit déclencher le système de freinage, arrêter le moteur et bien évidemment ouvrir la chaîne de sécurité. Comme mesure, la PESSRAL exige que le bus doit être certifié SIL 3.

Fonctionnement dégradé

Le fonctionnement dégradé du codeur rend la position de la cabine incertaine, ce qui provoque un déclenchement intempestif/non déclenchement d'une fonction de sécurité. Pour protéger le système contre cette défaillance, il faut réagir directement sur les frein, le moteur et la chaîne de sécurité. Un bus de communication certifié SIL 3, permet d'assurer la circulation de l'information dans tout le système.

Pour le calculateur cabine (CSC et CSM), cette défaillance dégrade le traitement des données. Cette dégradation agit directement sur le comportement intempestif de fonctions de sécurité. Pour faire face à cette dégradation, le moteur doit s'arrêter et un déclenchement de frein doit être fait dans un laps de temps bien borné (temps de réaction court). La solution proposée pour corriger cette défaillance c'est de faire une comparaison mutuelle des résultats entre les deux micro-contrôleur du CSC et CSM via le bus de communication.

Bibliographie

[61784-3, 2016] IEC 61784-3. Industrial communication networks - profiles - part 3 : Functional safety fieldbuses - general rules and profile definitions. Standard, IEC 61784-3, 2016.

[Addad, 2011] Boussad Addad. *Evaluation analytique du temps de réponse des systèmes de commande en réseau en utilisant l'algèbre (max,+).* PhD thesis, École normale supérieure de Cachan-ENS, 2011.

[Alves *et al.*, 2000] Mário Alves, Eduardo Tovar, and Francisco Vasques. Ethernet goes real-time : a survey on research and technological developments. Technical report, Polytechnic Institute of Porto School of Engineering (ISEP-IPP), 2000.

[Arlat *et al.*, 2006] Jean Arlat, Yves Crouzet, Yves Deswarte, Jean-Charles Fabre, Jean-Claude Laprie, and David Powell. Tolérance aux fautes. *Les Editions Vuibert, J. Akoka, I. Comyn-Wattiau (Eds)*, pages 241–270, 2006.

[Armoush, 2010] Ashraf Armoush. *Design patterns for safety critical embedded systems.* PhD thesis, RWTH Aachen University, 2010.

[Avižienis *et al.*, 2001] Algirdas Avižienis, Jean-Claude Laprie, Brian Randell, et al. *Fundamental concepts of dependability.* University of Newcastle upon Tyne, Computing Science Newcastle upon Tyne, UK, 2001.

[Avižienis *et al.*, 2004] Algirdas Avižienis, Jean-Claude Laprie, and Brian Randell. Dependability and its threats : a taxonomy. In *Building the Information Society*, pages 91–120. Springer, 2004.

[Boyer *et al.*, 2016] Marc Boyer, Hugo Daigmorte, Nicolas Navet, and Jörn Migge. Performance impact of the interactions between time-triggered and rate-constrained transmissions in tte-thernet. 2016.

[Brenier, 2001] Henri Brenier. *Les spécifications fonctionnelles : automatismes industriels et temps réel.* Dunod, 2001.

[Bush *et al.*, 2017] Stephen Francis Bush, Joel Frederick Markham, Andrew William Berner, et al. Communication system and method for controlling data distribution quality of service in time sensitive networks, September 21 2017. US Patent App. 15/292,709.

[Cena *et al.*, 2009] Gianluca Cena, Lucia Seno, Adriano Valenzano, and Stefano Vitturi. Performance analysis of Ethernet Powerlink networks for distributed control and automation systems. *Computer Standards & Interfaces*, 31(3) :566–572, 2009.

[Chabrol, 2006] Damien Chabrol. *Étude, conception et mise en œuvre d'un protocole de communication synchrone tolérant aux fautes et prédictible sur des composants réseaux standards.* PhD thesis, Paris 11, 2006.

[Ciame *et al.*, 1999] Groupe Ciame, Isabelle Augé-Blum, Mireille Bayart, Martine Wahl, Guy Benoit, Eric Benoit, Laurent Cauffriez, Philippe Charpentier, André Chovin, Joseph Ciccotelli, et al. *Réseaux de terrain : description et critères de choix.* Hermes-Lavoisier, 1999.

[Ciame *et al.*, 2009] Groupe Ciame, Isabelle Augé-Blum, Mireille Bayart, Martine Wahl, Guy Benoit, Eric Benoit, Laurent Cauffriez, Philippe Charpentier, André Chovin, Joseph Ciccotelli, et al. *Réseaux de terrain-critères de sûreté de fonctionnement*. Hermes-Lavoisier, 2009.

[De Carvalho, 1996] Francisco Vasques De Carvalho. *Sur l'intégration de mécanismes d'ordonnancement et de communication dans la sous-couche MAC de réseaux locaux temps réel*. PhD thesis, Université Paul Sabatier-Toulouse III, 1996.

[Decotignie, 2005] Jean-Dominique Decotignie. Ethernet-based real-time and industrial communications. *Proceedings of the IEEE*, 93(6) :1102–1117, 2005.

[Dorow, 2003] Kevin Dorow. Flexible fault tolerance in configurable middleware for embedded systems. In *Computer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International*, pages 563–569. IEEE, 2003.

[Doyle, 2004] Paula Doyle. Introduction to real-time Ethernet II. *The Extension—A Technical Supplement to Control Network*, 5(4), 2004.

[Dubrova, 2013] Elena Dubrova. *Fault-tolerant design*. Springer, 2013.

[Eth, 2017] Ethercat technology group. `http://www.ethercat.org`, 2017. Accessed : 2017-03-30.

[Felser and Sauter, 2004] Max Felser and Thilo Sauter. Standardization of industrial ethernet-the next battlefield ? In *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pages 413–420. IEEE, 2004.

[Felser, 2005] Max Felser. Real-time ethernet–industry prospective. *Proceedings of the IEEE*, 93(6) :1118–1129, 2005.

[Gavrilut *et al.*, 2017] Voica Gavrilut, Bahram Zarrin, Paul Pop, and Soheil Samii. Fault-tolerant topology and routing synthesis for ieee time-sensitive networking. In *25th International Conference on Real-Time Networks and Systems*, 2017.

[Georges *et al.*, 2006] Jean-Philippe Georges, Nicolas Krommenacker, Thierry Divoux, and Eric Rondeau. A design process of switched ethernet architectures according to real-time application constraints. *Engineering Applications of Artificial Intelligence*, 19(3) :335–344, 2006.

[Ghostine, 2008] Rony Ghostine. *Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau*. PhD thesis, Institut National Polytechnique de Lorraine-INPL, 2008.

[Jasperneite *et al.*, 2002] Jürgen Jasperneite, Peter Neumann, Michael Theis, and Kym Watson. Deterministic real-time communication with switched ethernet. In *Proceedings of the 4th IEEE International Workshop on Factory Communication Systems*, pages 11–18, 2002.

[Jasperneite *et al.*, 2007] 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.

[Kaczmarczyk *et al.*, 2011] Václav Kaczmarczyk, M. Sir, and Z. Bradac. Ethernet powerlink asynchronous phase examination. In *Proceedings of the 5th WSEAS international conference on Communications and information technology*, pages 152–157. World Scientific and Engineering Academy and Society (WSEAS), 2011.

[Kalappa *et al.*, 2006] Naveen Kalappa, Kristen Acton, Marco Antolovic, Siddharth Mantri, Jonathan Parrott, Jon Luntz, James Moyne, and Dawn Tilbury. Experimental determination of real time peer to peer communication characteristics of ethernet/ip. In *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1061–1064. IEEE, 2006.

[Knezic *et al.*, 2016] Mladen Knezic, Branko Dokic, and Zeljko Ivanovic. Improving the flexibility of the ethernet powerlink pollresponse chaining mechanism. In *Factory Communication Systems (WFCS), 2016 IEEE World Conference on*, pages 1–4. IEEE, 2016.

[Knight, 2012] John Knight. *Fundamentals of Dependable Computing for Software Engineers*. CRC Press, 2012.

[Kopetz, 1991] Hermann Kopetz. Event-triggered versus time-triggered real-time systems. In *Operating Systems of the 90s and Beyond*, pages 86–101. Springer, 1991.

[Kurose *et al.*, 1984] James F Kurose, Mischa Schwartz, and Yechiam Yemini. Multiple-access protocols and time-constrained communication. *ACM Computing Surveys (CSUR)*, 16(1) :43–70, 1984.

[Laprie *et al.*, 1996] Jean-Claude Laprie, Jean Arlat, Alain Costes, and J-P Blanquart. *Guide de la sûreté de fonctionnement*. Cépaduès-éditions, 1996.

[Laprie, 1985] Jean-Claude Laprie. Dependable computing and fault-tolerance. *Digest of Papers FTCS-15*, pages 2–11, 1985.

[Laprie, 1989] Jean-Claude Laprie. Sûreté de fonctionnement des systèmes informatiques et tolérance aux fautes. *Techniques de l'ingénieur. Informatique industrielle*, 2(R7595) :R7595–1, 1989.

[Lelionnais, 2014] Cédrick Lelionnais. *Contribution à la considération du comportement des plates-formes d'exécution logicielles temps réel*. PhD thesis, École Centrale de Nantes, 2014.

[Limal *et al.*, 2007] Steve Limal, Stéphane Potier, Bruno Denis, and Jean-Jacques Lesage. Formal verification of redundant media extension of ethernet powerlink. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pages 1045–1052. IEEE, 2007.

[Limal, 2009] Steve Limal. *Architectures de contrôle-commande redondantes à base d'Ethernet Industriel : modélisation et valdiation par modèle-checking temporisé*. PhD thesis, École normale supérieure de Cachan-ENS, 2009.

[Malcolm and Zhao, 1995] Nicholas Malcolm and Wei Zhao. Hard real-time communication in multiple-access networks. *Real-Time Systems*, 8(1) :35–77, 1995.

[Malcolm *et al.*, 1990] Nicholas Malcolm, Wei Zhao, and Chris Barter. Guarantee protocols for communication in distributed hard real-time systems. In *INFOCOM'90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. The Multiple Facets of Integration. Proceedings, IEEE*, pages 1078–1086. IEEE, 1990.

[Marques *et al.*, 2012] Luis Marques, Verónica Vasconcelos, Paulo Pedreiras, and Luis Almeida. Tolerating transient communication faults with online traffic scheduling. In *Industrial Technology (ICIT), 2012 IEEE International Conference on*, pages 396–402. IEEE, 2012.

[Marsal, 2006] Gaëlle Marsal. *Evaluation of time performances of Ethernet-based automation systems by simulation of high-level Petri nets*. PhD thesis, École normale supérieure de Cachan-ENS, 2006.

[Marszal and Scharpf, 2002] E.M. Marszal and E.W. Scharpf. *Safety integrity level selection : Systematic methods including layer of protection analysis*. ISA, 2002.

[Molle and Kleinrock, 1985] M Molle and Leonard Kleinrock. Virtual time csma : Why two clocks are better than one. *IEEE transactions on Communications*, 33(9) :919–933, 1985.

[Novak and Tamandl, 2007] T. Novak and T. Tamandl. Architecture of a safe node for a fieldbus system. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 1, pages 101–106. IEEE, 2007.

[Novak *et al.*, 2007] Thomas Novak, Albert Treytl, and Peter Palensky. Common Approach to Functional Safety and System Security in Building Automation and Control Systems. In *Emerging Technologies and Factory Automation, 2007. ETFA'07. IEEE Conference on*, pages 1141–1148. IEEE, 2007.

[Omn, 2016] Omnet++, discrete event simulator. `http://omnetpp.org`, 2016. Accessed : 2016-09-30.

[Pailler, 2006] Stéphane Pailler. *Analyse hors ligne d'ordonnançabilitc d'applications temps réel comportant des tâches conditionnelles et sporadiques.* PhD thesis, École Nationale Supérieure de Mécanique et d'Aérotechnique, Université de Poitiers, 2006.

[Prytz, 2008] Gunnar Prytz. A performance analysis of ethercat and profinet irt. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 408–415. IEEE, 2008.

[Ricque and Vieille, 2005] Bertrand Ricque and Jean Vieille. Guide d'interprétation et d'application de la norme IEC 61508 et des normes dérivées. Technical report, The Instrumentation, Systems, and Automation Society, ISA, 2005.

[Robert *et al.*, 2012] Jérémy Robert, Jean-Philippe Georges, Eric Rondeau, and Thierry Divoux. Minimum cycle time analysis of Ethernet-based real-time protocols. *International Journal of Computers, Communications and Control*, 7(4) :743–757, 2012.

[Seno and Vitturi, 2007] Lucia Seno and Stefano Vitturi. A simulation study of ethernet powerlink networks. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, pages 740–743. IEEE, 2007.

[Seno *et al.*, 2009] Lucia Seno, Stefano Vitturi, and Claudio Zunino. Analysis of ethernet powerlink wireless extensions based on the ieee 802.11 wlan. *IEEE Transactions on Industrial Informatics*, 5(2) :86–98, 2009.

[Seno, 2011] Lucia Seno. *Real-time networks and protocols for industrial automation.* PhD thesis, Università degli Studi di Padova, 2011.

[Smith and Simpson, 2004] David Smith and Kenneth Simpson. *Functional safety.* Routledge, 2004.

[Song *et al.*, 2002] Yeqiong Song, Anis Koubaa, and Francois Simonot. Switched Ethernet for real-time industrial communication : Modelling and message Buffering delay evaluation. In *Factory Communication Systems, 2002. 4th IEEE International Workshop on*, pages 27–35. IEEE, 2002.

[Soury *et al.*, 2015a] Ayoub Soury, Melek Charfi, Denis Genon-Catalot, and Jean-Marc Thiriet. Performance analysis of Ethernet Powerlink protocol : Application to a new lift system generation. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, volume 2015-Sept, pages 1–6. IEEE, sep 2015.

[Soury *et al.*, 2015b] Ayoub Soury, Denis Genon-Catalot, and Jean-marc Thiriet. New lift safety architecture to meet PESSRAL requirements. In *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, volume 2015-March, pages 1–5. IEEE, mar 2015.

[Storey, 1996] Neil R Storey. *Safety critical computer systems.* Addison-Wesley Longman Publishing Co., Inc., 1996.

[Thiele *et al.*, 2016] Daniel Thiele, Johannes Schlatow, Philip Axer, and Rolf Ernst. Formal timing analysis of can-to-ethernet gateway strategies in automotive networks. *Real-time systems*, 52(1) :88–112, 2016.

[Thomesse, 2005] J-P Thomesse. Fieldbus technology in industrial automation. *Proceedings of the IEEE*, 93(6) :1073–1101, 2005.

[Varga, 2005] András Varga. Omnet++ discrete event simulation system user manual-version 3.2. *Technical University of Budapest/Hungary*, 2005.

[Villemeur, 1988] Alain Villemeur. Sûreté de fonctionnement des systèmes industriels. *Collection de la Direction des Études et Recherches d'Électricité de France*, 1988.

[Wollschlaeger *et al.*, 2017] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. The future of industrial communication : Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1) :17–27, 2017.

[Zammali, 2016] Amira Zammali. *Approche d'intégrité bout en bout pour les communications dans les systèmes embarqués critiques : application aux systèmes de commande de vol d'hélicoptères.* PhD thesis, Universite Toulouse III Paul Sabatier, 2016.