



HAL
open science

Validation temporelle dans les réseaux embarqués temps réel

Frédéric Ridouard

► **To cite this version:**

Frédéric Ridouard. Validation temporelle dans les réseaux embarqués temps réel. Réseaux et télécommunications [cs.NI]. Université de Poitiers, 2023. tel-04603900

HAL Id: tel-04603900

<https://hal.science/tel-04603900>

Submitted on 6 Jun 2024

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.



HDR

Habilitation à Diriger la Recherche

Délivré par : **Université de Poitiers**

Présentée et soutenue le *Date de défense (27/11/2023)* par :

Frédéric Ridouard

Validation temporelle dans les réseaux embarqués temps réel

JURY

JEAN-LUC SCHARBARG	Professeur d'université	Président du Jury
LILIANA CUCU-GROSJEAN	Directrice de recherche	Rapporteuse
AHLEM MIFDAOUI	Chercheuse (HDR)	Rapporteuse
YE-QIONG SONG	Professeur d'université	Rapporteur
NICOLAS NAVET	Full Professor	Membre du jury
PASCAL RICHARD	Professeur d'université	Membre du jury

École doctorale :

MIMME : Mathématiques, Informatique, Matériaux, Mécanique, Energétique

Unité de Recherche :

LIAS - Laboratoire d'Informatique et d'Automatique pour les Systèmes (UR 20299)

à Nelly, Charlotte et Louise

Remerciements

Je veux remercier les membres du jury pour l'intérêt qu'ils ont porté à mes travaux de recherche. Nos nombreux échanges, au cours de ma carrière ont toujours contribué à m'améliorer et à explorer de nouvelles pistes.

Je tiens à remercier Pascal Richard pour ma formation à la recherche en ordonnancement temps réel. Travailler avec toi a toujours été un plaisir... Encore plus en démontrant que tes idées étaient régulièrement fausses. Merci aussi à Jean-Luc Scharbarg pour m'avoir initié aux réseaux embarqués et à l'avionique et pour avoir, ainsi, complété ma formation. Pascal et Jean-Luc, vous avez guidé et façonné le chercheur et l'encadrant que je suis aujourd'hui.

Je souhaite remercier Georges Arnaud Kemayo et Nassima Benammar qui ont travaillé avec moi durant leurs thèses. Une partie importante des travaux de ce présent manuscrit est issue ou à l'origine de leurs thèses. Travailler avec chacun a été un réel plaisir et très stimulant au quotidien. Ce plaisir se poursuit aujourd'hui avec les thèses de Richard Garreau, d'Adrien Roberty et de Luc Descharmes.

Je tiens aussi à remercier les collègues du laboratoire. Votre convivialité et nos discussions plus ou moins sérieuses contribuent fortement à mon bien-être au travail. En particulier, merci à Henri avec qui je partage mon bureau depuis de nombreuses années. Nous avons pu partager de nombreuses et productives discussions.

Merci à Nelly mon épouse, soutien infailible au quotidien et à mes deux rayons de soleil, Charlotte et Louise. Je veux également avoir une pensée émue pour mes parents qui nous ont quitté. Ils ont toujours été derrière moi et m'ont soutenu dans mes choix.

Table des matières

Introduction générale	3
I Résumé de carrière	5
1 Curriculum Vitae	7
1.1 État civil	9
1.2 Synthèse du parcours professionnel et contexte d'exercice	9
1.3 Investissement pédagogique	11
1.4 Activité scientifique	14
1.5 Annexes	25
II Contexte et motivations	33
2 Réseau temps réel	35
2.1 Système temps réel	35
2.2 Système temps réel distribué	36
2.3 Réseaux embarqués basés sur Ethernet	39
2.4 Propriétés des réseaux temps réel	44
2.5 Conclusion	45
3 Analyse du déterminisme d'un réseau temps-réel	47
3.1 Problématique	47
3.2 Méthodes d'analyse d'une borne supérieure pire cas sur les délais	50
4 Motivations	57
4.1 Bilan	57

4.2	Contributions	58
III	Contributions à la validation temporelle des réseaux embarqués temps-réel	59
5	Forward end-to-end delay Analysis en FIFO	61
5.1	Introduction	63
5.2	Modèle réseau	63
5.3	Principe général de la méthode FA	65
5.4	Méthode FA appliquée à FIFO	66
5.5	Expérimentations	74
5.6	Conclusion	78
6	La méthode FA en FP/FIFO	79
6.1	Introduction	81
6.2	Modèle réseau en FP/FIFO	81
6.3	Méthode FA appliquée à FP/FIFO	83
6.4	Expérimentations en FP/FIFO	89
6.5	Conclusion	92
7	La méthode FA en CBS	93
7.1	Introduction	95
7.2	Modèle réseau en CBS	95
7.3	Le calcul du délai pire cas d'une trame en CBS	97
7.4	Expérimentations	105
7.5	Conclusion	108
8	La méthode BDF pour le dimensionnement de files d'attente	111
8.1	Introduction	113

8.2	Modèle réseau de BDF	114
8.3	BDF : Dimensionnement d'une file d'attente en nombre de trames	115
8.4	Expérimentations	121
8.5	Conclusion	122
IV	Bilan et perspectives	125
9	Bilan de mes travaux de recherche	127
9.1	Analyse du délai de bout-en-bout de traversée du réseau	127
9.2	Dimensionnement des files d'attente	128
9.3	Bilan	128
10	Perspectives de recherche	131
10.1	<i>Time-Sensitive Networking</i> (TSN)	131
10.2	Ordonnancement hiérarchisé et chaînes fonctionnelles	131
10.3	Ordonnancement temps réel	132
	Bibliographie	143
	Liste des acronymes	145
	Liste des notations	147

Table des figures

2.1	Évolution des systèmes avioniques d'Airbus.	36
2.2	Partitionnement d'un CPM.	37
2.3	Architecture Modulaire intégrée de l'A380 (a.) et celle distribuée de l'A350 (b.)[MOG18].	38
2.4	Exemple d'architecture distribuée issue de l'automobile[Aut].	39
2.5	End System AFDX.	40
2.6	Commutateur AFDX.	40
2.7	Exemple de configuration AFDX.	41
2.8	Configuration industrielle AFDX de l'A380 [Bau11].	42
2.9	Exemple de configuration AVB[Ald+12]; [Ste+12].	44
2.10	Les sept couches du modèle OSI.	45
3.1	Analyse du délai de bout-en-bout d'une trame.	49
3.2	Courbe d'arrivée d'un flux en NC.	51
3.3	Schéma général du NC.	52
5.1	Modélisation d'un nœud h	64
5.2	Calcul itératif de R_i^h et $Smax_i^h$	65
5.3	Le scénario pire cas d'arrivée des trames générées par $v_i \in \Gamma^h$ dans le nœud h , maximisant l'arriéré de travail durant tout intervalle de temps $[0, t]$	68
5.4	La charge de travail maximale arrivant sur un nœud h via le lien d'entrée IP_x^h durant un intervalle de temps de longueur t	71
5.5	Les instants t à tester.	74
5.6	Modélisation par FA de la configuration illustrée Fig. 2.7.	76
5.7	Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA).	77

6.1	Modèle nœud en FP/FIFO.	82
6.2	Les scénarios d'arrivée pire cas des trames (tel que $v_j \in \Gamma^h$) où la rafale de $(1 + k_j^h)$ trames arrive à l'instant 0 (a) ou à l'instant t (b).	86
6.3	Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA) en FP/FIFO.	91
7.1	Modèle de nœud CBS.	96
7.2	Un exemple d'ordonnancement suivant le régulateur CBS.	97
7.3	Scénario pire cas pour la politique de service CBS.	98
7.4	Scénario pire cas défini et prouvé par Bordoloi <i>et al.</i> [Bor+14] maximisant le temps de transmission total généré par une classe Y sur un nœud CBS durant un intervalle de temps de durée t	99
7.5	Modélisation par FA de la configuration illustrée Fig. 2.9.	106
7.6	Port de sortie d'un nœud CBS.	107
7.7	Topologie linéaire.	108
7.8	Topologie en <i>cluster</i>	108
7.9	Comparaison des résultats (en μs) obtenus par FA et CPA pour la configuration de la Fig. 7.7.	109
7.10	Comparaison des résultats (en μs) obtenus par FA et CPA pour la configuration représentée par la Fig. 7.8.	109
8.1	Non-concordance des pires dimensionnement en termes de bits et de trames.	114
8.2	Modèle du nœud h utilisé par BDF.	114
8.3	Scénarios d'arrivée dans une file d'attente FIFO.	115
8.4	Arrivée au plus tôt des trames d'un flux $v_i \in \Gamma^h$ dans une file d'attente conservative h	117
8.5	Amélioration du dimensionnement des files d'attente par BDF en comparaison avec la méthode naïve sur une configuration industrielle.	122
8.6	Comparaison de la mémoire allouée pour chaque commutateur (SW) entre BDF et la méthode naïve.	123

Liste des tableaux

1.1	Bilan récapitulatif des publications	18
4.1	Bilan des principales approches d'analyse de performance dans les réseaux AFDX et AVB.	57
9.1	Contributions à l'analyse de performances dans les réseaux AFDX et AVB.	128

Introduction générale

Ce document présente la synthèse de mes travaux de recherches menés depuis l'obtention de mon doctorat en 2006 à l'Université de Poitiers. Mes travaux sur l'analyse de performance dans les réseaux embarqués ont débuté lors de mes deux années (post-Doctorat puis ATER) dans l'équipe *Ingénierie Réseaux et Télécommunications* (IRT) de l'Institut de Recherche en Informatique de Toulouse (IRIT). EN 2009, j'ai été nommé au poste de maître de conférences en informatique (section 27) à l'Université de Poitiers dans le département Génie Électrique et Informatique Industrielle (GEII) de l'IUT de Poitiers. J'ai, ainsi, initié puis développé la thématique des réseaux embarqués au sein du LISI (ex LIAS) et de son équipe *Systèmes Embarqués Temps Réel*. Équipe dans laquelle je suis toujours affecté depuis ma mutation à l'École Nationale Supérieure de Mécanique et d'Aérotechnique (ENSMA) en 2019.

J'ai obtenu mon doctorat dans la même équipe, en 2006, sous la direction de Pascal RICHARD. Durant ma thèse, J'ai travaillé sur des problèmes d'ordonnancement :

- **ordonnancement classique de machines à traitement par lot** : dans le cadre d'une activité contractuelle avec des laboratoires pharmaceutiques *Workload For Labs* (W4L) et en collaboration avec Patrick Martineau (Université de Tours), j'ai notamment défini l'algorithme αH^∞ et démontré qu'il a le meilleur ratio de compétitivité parmi les algorithmes en-ligne pour l'ordonnancement par machines à traitement par lots ;
- **ordonnancement temps réel de tâches à suspension**. Principalement, j'ai contribué à démontrer la difficulté de cet ordonnancement (complexité, anomalies, analyse de compétitivité, ordonnançabilité,...) pour les algorithmes en-ligne.

Après mon doctorat, je suis parti deux ans (en post-doctorat puis ATER) dans l'équipe IRT à travailler principalement avec Jean-Luc Scharbarg et Christian Fraboul. Nous avons travaillé sur l'analyse de performances dans les réseaux embarqués temps réel et en particulier, l'AFDX.

A mon arrivée, l'équipe *Systèmes Embarqués Temps Réel* était composée de sept titulaires et développait deux axes de recherche : l'ordonnancement temps réel et l'ingénierie dirigée par les modèles. Mon arrivée a permis de développer un nouvel axe, l'analyse de performances dans les réseaux embarqués. Ce thème a été renforcé par le recrutement d'Henri Bauer. Depuis mon arrivée, j'ai encadré trois stages de Master (Abdelaziz Ahmed Nacer, Patrick Bakabadio et Richard Garreau), deux stages ingénieur (Pierrick Richard et Tanguy Relo), un stage de fin d'études ingénieur (Luc Descharmes), deux thèses (Georges Arnaud Kemayo et Nassima Benammar) et j'encadre, actuellement, trois doctorants (Richard Garreau depuis le 01/10/2020, Adrien Roberty depuis le 01/02/2021 et Luc Descharmes depuis le 01/10/2022). Pour finir, je dirige l'équipe *Systèmes Embarqués Temps Réel* depuis 2017.

Ce document est rédigé pour mon Habilitation à diriger des recherches (HDR) mais il a aussi vocation à servir de synthèse pour l'ensemble des travaux qui ont été menés sur ce sujet pour tout étudiant ou chercheur intéressé par le sujet.

Mon Curriculum Vitae est détaillé partie I. Le contexte général de nos travaux ainsi que leurs motivations sont présentés dans la partie II. Les contributions principales développées dans ce document portent sur la définition et le développement de l'approche *Forward end-to-end Analysis* (FA). Comme nous le verrons dans la partie III, tous les résultats sont formellement démontrés. Les premiers travaux sur l'approche FA, menés dans un contexte AFDX, se sont inscrits dans le cadre du projet *Plateforme AvioNique moDulAire étendu* (PANDA) porté par le Conseil pour la Recherche

Aéronautique Civile (CORAC). Dans le cadre de ce projet, de type *Programme d'Investissements d'Avenir*, le LIAS a travaillé à définir une méthode d'analyse temporelle des applications avioniques AME (Avionique Modulaire Étendue), qui permette d'inter-opérer avec d'autres méthodes d'analyse autour d'un modèle commun de représentation d'une architecture AME.

Pour conclure ce document, la partie IV présente un bilan et mes perspectives de recherche en systèmes embarqués.

Première partie

Résumé de carrière

Curriculum Vitae

Sommaire

1.1	État civil	9
1.2	Synthèse du parcours professionnel et contexte d'exercice	9
1.2.1	Fonctions et titres universitaires	9
1.2.2	Principales responsabilités	11
1.2.3	Autres formations	11
1.3	Investissement pédagogique	11
1.3.1	Présentation synthétique de l'activité d'enseignement	11
1.3.2	Présentation des enseignements	12
1.3.3	Responsabilités pédagogiques	13
1.4	Activité scientifique	14
1.4.1	Présentation synthétique des thématiques de recherche	14
1.4.2	Publications et productions scientifiques	18
1.4.3	Encadrement doctoral et scientifique	18
1.4.4	Diffusion et rayonnement	20
1.4.5	Responsabilités scientifiques	23
1.4.6	Responsabilités et mandats locaux ou régionaux	24
1.5	Annexes	25
1.5.1	Tableau des enseignements	25
1.5.2	Liste classée des publications	27

1.1 État civil

Frédéric Ridouard

Age : 44 ans

Marié, 2 enfants

Établissement : ISAE-ENSMA

Unité de recherche : Laboratoire d'Informatique et d'Automatique pour les Systèmes (UR 20299)

Corps : Maître de Conférences (CN)

Discipline/section : Informatique (CNU 27)

1.2 Synthèse du parcours professionnel et contexte d'exercice

Je suis Maître de Conférences, section CNU 27, depuis le 1er septembre 2019 à l'École Nationale Supérieure de Mécanique et d'Aérotechnique (ENSMA) de Poitiers. Pour les enseignements, je suis rattaché au département Informatique et Automatique et en recherche, à l'équipe *Systèmes Embarqués Temps-Réel* (SETR) du Laboratoire d'Informatique et d'Automatique pour les Systèmes (LIAS).

1.2.1 Fonctions et titres universitaires

En 1996 après l'obtention de mon baccalauréat, je me suis inscrit à l'Université de Poitiers en DEUG *Mathématiques, informatique et applications aux sciences*. Une fois l'obtention de la **Maîtrise de mathématiques** (équivalence Master 1), je me suis réorienté en Informatique au niveau Licence. En 2003, j'ai effectué ma première expérience en recherche au cours du stage de **DEA (équivalence Master 2) au sein du laboratoire LISI (ex. LIAS)**.

J'ai poursuivi en thèse, toujours au LISI, sous la direction de Pascal Richard. La thématique était axée principalement sur la résolution de problèmes d'**ordonnancement temps réel**. En 2006, j'ai obtenu mon **Doctorat d'informatique à l'Université de Poitiers**. J'ai obtenu la **qualification en sections CNU 27 et 61**. C'est également durant cette thèse que j'ai commencé à enseigner en tant

Maître de conférences (section 27) à l'ISAE-ENSMA	2019-Présent
Maître de conférences (section 27) à l'Université de Poitiers	2009-2019
Ingénieur d'étude et chef de projet au sein de la société <i>SII</i> à Toulouse.	2008-2009
ATER à temps complet à l' <i>ENSEEIH</i> T à Toulouse.	2007-2008
Post-Doctorat à l'Institut de Recherche en Informatique de Toulouse.	2006-2007
Doctorat d'informatique , allocation du ministère, Mention très honorable.	2003-2006
- Titre : Contributions à des problèmes d'ordonnancement en-ligne : l'ordonnancement temps réel de tâches à suspension et l'ordonnancement par une machine à traitement par lot.	
- Directeur de Thèse : Pascal RICHARD	
- Laboratoire d'accueil : LISI, Université de Poitiers/ENSMA	
DEA de Traitement de l'Information : Informatique, Images, Automatique , mention Très Bien, Université de Poitiers.	2002-2003
Maîtrise d'informatique , mention Très Bien, Université de Poitiers.	2001-2002
Licence d'informatique , mention Bien, Université de Poitiers.	2000-2001
Maîtrise de Mathématiques , Université de Poitiers.	1999-2000
Licence de Mathématiques , Université de Poitiers.	1998-1999
DEUG MIAS , Université de Poitiers.	1996-1998
Baccalauréat Scientifique , Lycée Notre Dame (Niort, 79).	1996

que vacataire à l'Université de Poitiers et également à l'École Nationale Supérieure de Mécanique et d'Aérotechniques (ENSMA).

À la suite de l'obtention de mon doctorat, j'ai intégré l'équipe *Ingénierie Réseaux et Télécommunications* (IRT) de l'*Institut en Recherche Informatique de Toulouse* (**IRIT**) en **Post-Doctorat** puis en tant qu'**ATER à temps complet** à l'*École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, d'Hydraulique et des Télécommunications* (ENSEEIH)T de Toulouse. Durant cette expérience de deux ans, j'ai développé ma thématique de recherche sur l'analyse de performance dans les **réseaux temps réel embarqués**. Cette expérience m'a permis aussi de travailler sur les problématiques d'**AIRBUS** pour la validation de ses réseaux embarqués. À Toulouse en 2008, après mes deux ans passés à l'IRIT, j'ai été recruté par la société SII au poste d'**ingénieur d'étude et de chef de projet** pour le développement du système de maintenance centralisé dans le cadre du projet A350 XWB d'Airbus.

En 2009, j'ai été recruté au poste de **Maître de Conférences par l'Université de Poitiers en section 27**. J'ai ainsi intégré l'Institut Universitaire de Technologie de Poitiers et son département Génie Électrique et Informatique Industrielle. Pour la recherche, j'ai intégré en parallèle le *Laboratoire d'Informatique et d'Automatique pour les Systèmes* (LIAS) et son équipe *Systèmes Embarqués Temps Réel* (SETR). Je suis, actuellement, **Maître de Conférences à l'ISAE-ENSMA** depuis septembre 2019. J'enseigne au département *Informatique et Automatique* et j'effectue ma recherche au sein de l'équipe *Systèmes Embarqués Temps Réel* (SETR) du **LIAS**. Mon recrutement a permis d'élargir les thématiques du LIAS à une problématique connexe de l'ordonnancement temps réel, les réseaux embarqués temps réel.

1.2.2 Principales responsabilités

Responsable de l'équipe <i>Systèmes Embarqués Temps Réel</i>	<i>2017-Présent</i>
Membre du Conseil Scientifique de l'ISAE-ENSMA	<i>2021-Présent</i>
Membre du Conseil de l'Unité de Recherche (CUR) du LIAS	<i>2022-Présent</i>
Membre de la Commission d'Expertise Scientifique CNU 27 de l'ENSMA	<i>2020-Présent</i>
Membre du Conseil Scientifique du LIAS	<i>2011-2022</i>
Membre de la Commission d'Expertise Scientifique CNU 27 de l'Université de Poitiers	<i>2013-2018</i>
Membre du Conseil du Département GEII	<i>2012-2014</i>
Représentant de l'IUT de Poitiers pour le Certificat Informatique et Internet (C2I) de l'Université de Poitiers	<i>2009-2016</i>
Chef de projet pour le développement d'un système embarqué	<i>2008-2009</i>
Représentant des doctorants au sein du Conseil de l'École Doctorale (ED-SPI&A)	<i>2004-2006</i>
Représentant des doctorants au sein du Conseil d'Administration de l'ENSMA	<i>2004-2006</i>
Représentant des doctorants au sein du Conseil Scientifique de l'ENSMA	<i>2004-2006</i>

1.2.3 Autres formations

École d'été temps-réel 2021 à l'ENSMA (Poitiers) Thématiques : Développement d'applications temps réel, méthodes formelles, ordonnancement et supports d'exécution, réseaux temps réel et qualité de service.	<i>2021</i>
École d'été temps-réel 2011 (Brest) Thématiques : Développement d'applications temps réel, méthodes formelles, ordonnancement et supports d'exécution, réseaux temps réel et qualité de service.	<i>2011</i>
École d'été temps-réel 2003 (Toulouse) Thématiques : Développement d'applications temps réel, méthodes formelles, ordonnancement et supports d'exécution, réseaux temps réel et qualité de service.	<i>2003</i>
École d'automne en Recherche Opérationnelle 2003 (Tours) Thématiques : Ordonnancement d'atelier, algorithmique, complexité	<i>2003</i>

1.3 Investissement pédagogique

1.3.1 Présentation synthétique de l'activité d'enseignement

Je suis, depuis 2019, enseignant-chercheur en tant que Maître de Conférences à l'ISAE-ENSMA, une école d'ingénieurs de taille moyenne. L'ISAE-ENSMA diplôme environ 300 étudiants par an (niveau master, ingénieur ou docteur, formation initiale ou par apprentissage), en s'appuyant sur une équipe d'une cinquantaine d'enseignants ou enseignants-chercheurs.

Pour les enseignements, j'ai intégré le département Informatique et Automatique (IA), dans une équipe de neuf enseignants-chercheurs (6 MCF et 3 PU). Cette intégration m'a permis une évolution majeure de mes enseignements. En effet à l'IUT de Poitiers, j'effectuais majoritairement des enseignements en Mathématiques. Maintenant à l'ENSMA, mes enseignements s'articulent autour de l'architecture matérielle et des systèmes embarqués. J'ai pris récemment la responsabilité du cours de *Systèmes Embarqués* de première année. Je mets à jour ce module avec la redéfinition du cours, des TDs et des TPs. Je suis également co-responsable d'un cours de troisième année, *Systèmes Avioniques* depuis 2011 (en vacation et maintenant en poste).

Depuis la rentrée 2021, l'ISAE-ENSMA a ouvert une formation sous statut apprenti, pour laquelle, j'interviens. Ainsi, je suis responsable du cours d'*Architecture des Systèmes Numériques* pour lequel j'ai dû tout construire depuis zéro. L'ISAE-ENSMA a la particularité d'avoir une formation orientée bureau d'étude, qui laisse une place importante aux séances de travaux pratiques, avec des taux d'encadrement de un pour douze. La charge d'enseignement générée est importante au niveau de l'ensemble du département, amplifiée par le départ de deux collègues (retraite et disponibilité) sans qu'ils aient été remplacés.

Durant mes dix années à l'IUT de Poitiers, j'ai travaillé au sein du département GEII. Ce département diplômait environ 70 élèves par an et était animé par une vingtaine d'enseignants. J'y ai enseigné principalement les Mathématiques, dans les différents modules des deux années DUT1 et DUT2 en formation initiale et en formation par apprentissage. J'ai ainsi eu la responsabilité de plusieurs modules de mathématiques. J'ai également pris la responsabilité du cours de *Gestion de projet* et des *Projets Tutorés* de DUT1. Je suis également intervenu en Base de Données et interfaçage en deuxième année et en Systèmes de gestion des données dans le département Gestion des Entreprises et des Administrations (DUT2). Au titre des vacances, j'ai également repris le cours de *Systèmes Avioniques* en A3 à l'ENSMA.

En 2017, constatant les difficultés à intéresser les étudiants aux enseignements de Mathématiques, j'ai décidé de complètement changer mon approche. Ainsi, je suis le premier du département à avoir mis en place la **Pédagogie inversée** composée :

- d'un polycopié complet de cours avec des exercices d'entraînement corrigés ;
- des séances en amphithéâtre pour répondre aux questions et présenter, par la pratique, les principaux concepts ;
- Mise à disposition de QCM en ligne pour l'autoévaluation ;
- des séances de TD avec les étudiants travaillant en autonomie et en groupes de quatre.

Pendant les dix années passées à l'IUT, j'ai été tuteur de cinq étudiants en formation par apprentissage. De plus, j'ai également été rapporteur et/ou examinateur de bon nombre de stages et projets d'étudiants en GEII. Pour finir, pour la promotion du département GEII et de l'IUT de Poitiers, je participe aux Journées Portes Ouvertes de l'Université de Poitiers ainsi qu'à la visite de Lycées.

1.3.2 Présentation des enseignements

J'enseigne sur l'ensemble des trois années de la formation ingénieur ENSMA (A1-A2-A3, de niveaux respectifs L3-M1-M2) sous statut étudiant (FISE). Les promotions, sont d'environ : 170 étu-

dians en A1, 200 en A2 et 30 en A3 (cours d'option). Les horaires mentionnés ci-dessous sont les volumes pour un étudiant.

- **Systèmes Embarqués** (A1-FISE) : responsable du cours, interventions en cours (12,5h), TD (11,25h), TP (9h), responsable de l'examen ;
- **Systèmes Avioniques** (A3-FISE et M2 Informatique de l'Université de Poitiers) : co-responsable du cours, interventions en cours (10h) et TD (10h), co-responsable de l'examen ;
- Automatique (A2-FISE) : un des deux principaux intervenants en TD (12,5h) et TP (12h) ;
- Projet Avionique (A2-FISE) : intervenant (18h) ;
- Défi Signal-Système-Simulation (A1-FISE) : co-responsable du projet (12h).

Tous les ans, je participe au suivi et à l'évaluation des stages des étudiants en formation initiale :

- Enseignant référent côté ENSMA pour les Projets de Fin d'Études (A3-FISE) ;
- Rapporteur et soutenance de stages pour les Projets de Fin d'Études (A3-FISE) ;
- Rapporteur et soutenance de stages pour les Stages Ingénieur (A2-FISE).

J'enseigne également en première année de la formation par apprentissage du groupe ISAE : Formation Ingénieur Industrialisation et Méthodes pour l'Aéronautique et l'Espace. L'effectif est composé d'une vingtaine d'apprentis :

- **Architecture des Systèmes Numériques** (A1-FISA) : responsable du cours , interventions en cours/TD (22h), TP (8h), responsable de l'examen.

1.3.3 Responsabilités pédagogiques

Lorsque je fus ATER à l'ENSEEIH, j'ai participé en 2007-2008 à la réforme des enseignements de la troisième année et en particulier, à l'élaboration de l'option Systèmes embarqués de la troisième année du département *Télécommunications et Réseaux* de l'ENSEEIH.

À l'IUT en 2013, un nouveau *Programme Pédagogique National* (PPN) a été mis en place dans tous les départements GEII de France. J'ai intégré la commission nationale chargée de réorganiser l'intégralité des enseignements de Mathématiques en GEII. Au niveau local, j'ai été responsable de la mise en place de ce nouveau PPN avec la :

- construction des modules *Outils Logiciels 1 et 2* : ce sont deux modules composés exclusivement de TPs qui consistent en l'apprentissage d'un logiciel de calculs formels (Mathematica) ;
- construction des modules *Conduite de Projets* : ce sont deux modules qui traitent de la gestion de projets ;
- mise en place et encadrement des *Projets Tutorés* (DUT1) : sujet, suivi et évaluation ;
- réorganisation de l'ensemble des modules de Mathématiques ;
- construction du module de *Probabilités et Statistiques*.

Après une année de découverte des enseignements du département à l'ISAE-ENSMA, j'ai pris la responsabilité de plusieurs enseignements. La responsabilité d'une matière pour une promotion entière s'accompagne d'une charge généralement importante (rédaction d'un polycopié de cours, écriture et correction des évaluations (examens, TP) et gestion des différents intervenants. En FISA, j'ai

également construit totalement le contenu (polycopié, TDs, TPs) pour les modules Mathématiques Appliquées et Architecture des Systèmes Numériques.

Exemples de nouveau TP durant ces quatre dernières années :

- Écriture d'un nouveau TP sur les FPGA, circuits logiques programmables, en première année FISA ;
- Écriture d'un TP de suivi de piste par un véhicule Picar-S, en python via une carte matérielle raspberry Pi ;
- Écriture d'un TP de suivi de piste avec détection d'obstacle sur des véhicules Yahboom tank dits à capteurs intelligents en programmation Ada multitâches ;

La liste complète de mes enseignements effectués au cours des quatre dernières années est donnée en **annexe 1**.

1.4 Activité scientifique

1.4.1 Présentation synthétique des thématiques de recherche

La modélisation, l'évaluation et la validation des systèmes temps-réel embarqués constituent la thématique principale de recherche :

- Ma thématique de recherche initiale développée durant mon doctorat réalisé au sein de l'équipe *Systèmes Embarqués Temps Réel* du laboratoire LISI (ex-LIAS) *Laboratoire d'Informatique et d'Automatique pour les Systèmes* portait sur l'analyse de problèmes d'ordonnancement temps réel.
- Depuis mes deux années passées à l'*Institut de Recherche en Informatique de Toulouse (IRIT)* au sein de l'équipe *Ingénierie Réseaux et Télécommunications (IRT)* en post-doctorat puis en contrat d'ATER à l'ENSEEIHT, je travaille sur la modélisation, l'évaluation et la validation des réseaux embarqués temps réel tels que l'*Avionic Full Duplex Switched Ethernet (AFDX)*, l'*Audio/Vidéo Bridging Switched Ethernet (AVB)* ou le *Time Sensitive Networking (TSN)*.

1.4.1.1 Problèmes d'ordonnancement en-ligne

L'ordonnancement consiste à planifier des activités (tâches informatiques, messages sur un réseau, produits dans une chaîne de production, . . .) sur des ressources (machines, processeurs, réseaux, . . .) en respectant des contraintes opérationnelles (capacités des ressources, contraintes temporelles, . . .). L'ordonnancement est utilisé dans plusieurs domaines qui sont aussi nombreux que variés (*p.ex.* l'usage de pièces en production à la chaîne, la gestion des ressources humaines, l'exécution de programmes informatiques. . .) et notamment dans les systèmes embarqués au niveau des calculateurs.

Mon travail porte sur la conception d'algorithmes d'ordonnancement en-ligne (les prochaines activités ou tâches ne sont pas connues à l'avance par l'algorithme) et sur l'évaluation de performances de

ces algorithmes. J'ai apporté ma contribution à deux problèmes d'ordonnancement, au cours de mon doctorat, l'ordonnancement temps réel de tâches à suspension et l'ordonnancement par une machine à traitement par lot.

Ordonnancement temps réel Les tâches à suspension sont des tâches périodiques qui au cours de leur exécution demandent à se suspendre pour effectuer une opération externe (calculs, RPC, opérations d'entrée/sortie...) et lorsque leur suspension est terminée, elles finissent leur exécution. Ces tâches se retrouvent dans bon nombre de systèmes embarqués. Mais les concepteurs contournent le problème en intégrant la durée de suspension dans la durée d'exécution de la tâche. Cette intégration rend le problème pessimiste.

Nous avons abordé ce problème en cherchant, dans un premier temps, à démontrer des résultats sur la difficulté à résoudre un tel problème d'ordonnancement. Nous avons ainsi analysé la complexité à résoudre ce problème d'ordonnancement et avons démontré qu'il est \mathcal{NP} -Difficile au sens fort. De plus, nous avons prouvé la présence d'anomalies d'ordonnancement sous *Earliest Deadline First* (EDF) et également pour les algorithmes à priorité fixe. Il existe plusieurs définitions pour les anomalies d'ordonnancement, mais dans le cadre de nos travaux, une anomalie d'ordonnancement survient dès lors que diminuer la durée d'exécution ou de suspension d'une tâche peut augmenter le temps de réponse d'une autre tâche. Finalement, nous avons démontré la non-optimalité des algorithmes en-ligne déterministes pour l'ordonnancement de tâches à suspension sporadiques.

Nous nous sommes ensuite intéressés à l'évaluation de performances des algorithmes en-ligne classiques et leur capacité à ordonner des configurations de tâches à suspension. Nous avons ainsi établi la non-compétitivité (au sens de l'analyse de compétitivité) des algorithmes en-ligne classiques (*Rate Monotonic*, *Deadline Monotonic*, *Earliest Deadline First* et *Least Laxity First*) pour deux critères de performances (minimisation du nombre de tâches en retard et minimisation du temps de réponse maximum).

Enfin, nous avons étudié avec l'analyse de compétitivité différents tests d'ordonnançabilité. Ces tests, établis pour les algorithmes à priorité fixe, sont basés sur le calcul du temps de réponse maximum. Ces tests fournissent des conditions suffisantes d'ordonnançabilité. Nous avons, pour chacun de ces tests, essayé de quantifier le pessimisme qu'ils introduisent.

Ordonnancement classique L'ordonnancement par une machine à traitement par lot s'inscrit dans la thématique de l'ordonnancement classique. Ce problème consiste à étudier l'ordonnancement sur un processeur qui a la capacité d'exécuter plusieurs activités en parallèle. Ce travail est intégré dans une activité contractuelle avec des laboratoires pharmaceutiques, (*W4L*, Workload For Labs).

Pour cette étude, nous avons supposé que la capacité du processeur était infinie, c'est-à-dire qu'une infinité d'activités peuvent s'exécuter en parallèle sur le processeur. L'objectif recherché était celui d'établir un algorithme qui minimise la longueur d'ordonnancement (durée d'exécution d'un ensemble d'activités qui arrivent au fur et à mesure dans le temps).

Dans un premier temps, j'ai démontré que pour un algorithme en-ligne quelconque, il était toujours possible de définir une configuration d'activités telle que la longueur d'ordonnancement obtenue par

l'algorithme en-ligne est au moins $\frac{1+\sqrt{5}}{2}$ fois plus grande que celle obtenue par un algorithme optimal.

J'ai également proposé plusieurs algorithmes en-ligne compétitifs (au sens de l'analyse de compétitivité) pour des sous-problèmes du problème général (tâches avec temps processeurs égaux ou agréables, configurations avec seulement deux dates d'activation). Finalement, nous avons établi un algorithme en-ligne αH^∞ qui fait partie des meilleurs algorithmes en-ligne pour le problème général étudié puisque son ratio de compétitivité est égal à la borne inférieure du problème, $\frac{1+\sqrt{5}}{2}$.

1.4.1.2 Réseaux temps réel embarqués

Les architectures des réseaux embarqués dans les domaines des transports tels que l'avionique ou l'automobile ont connu, depuis la fin des années 90, une évolution majeure. En effet, l'informatisation des systèmes, la démultiplication des fonctions embarquées et l'augmentation des échanges de données impliquent une modification majeure du support de communication. Pour répondre à ces besoins croissants, les systèmes temps réel se développent sur une architecture distribuée avec un protocole réseau à hautes performances (*p.ex.* Ethernet).

L'objectif de cette architecture est le partage de ressources et l'échange de données entre les fonctions via le réseau. Je travaille principalement à l'évaluation de performances et la validation des réseaux embarqués utilisés dans les systèmes temps réel distribués tels que l'*Avionic Full Duplex Switched Ethernet* (AFDX) dans le contexte avionique et l'*Audio Vidéo Bridging Switched Ethernet* (AVB) pour l'automobile et aujourd'hui *Time Sensitive Networking* dans l'industrie 4.0. Ces protocoles réseaux sont basés sur le standard Ethernet afin de bénéficier de ses avantages (haut débit, disponibilité, facilité de maintenance et faible coût). Cependant, des contraintes supplémentaires permettent de garantir leurs comportement temps réel. Dans le cadre de mes travaux, en se plaçant dans le cas nominal (absence de panne), l'objectif est de démontrer le déterminisme de ces réseaux. En particulier, je travaille à démontrer que le délai de bout-en-bout de chaque trame traversant le réseau respecte son échéance (contrainte de temps maximum) et qu'aucune trame ne peut être perdue dans le réseau (par débordement des files d'attente). Les méthodes exactes pour le calcul du délai de bout-en-bout dans des réseaux de paquets telles que le *Model Checking* entraînent des problèmes d'explosion combinatoire. Ainsi, je travaille sur des méthodes pessimistes qui déterminent des bornes supérieures sur les délais de bout-en-bout.

Contributions À l'IRIT, j'ai développé une version probabiliste du Network Calculus. Le Network Calculus est la méthode d'analyse des délais de bout-en-bout dans les réseaux la plus reconnue. Nous avons démontré formellement son applicabilité dans les réseaux de paquets avec une politique *First-In-First-Out* (FIFO) tels que l'AFDX. Nous avons utilisé le Network Calculus probabiliste sur une configuration AFDX industrielle fournie par Airbus pour déterminer les délais de bout en bout des différents flux. Nous avons comparé ces résultats avec ceux obtenus à l'aide du Network Calculus (classique). Le Network Calculus probabiliste induit du pessimisme puisque ce n'est pas la probabilité qui est réellement calculée mais une borne supérieure sur la probabilité de dépasser un délai qui est calculée. Cependant, il a été démontré que le Network Calculus probabiliste est bien moins pessimiste que le Network Calculus pour une majorité de flux.

Nous avons étendu le Network Calculus probabiliste à la politique de service *Fixed Priority/FIFO* (FP/FIFO) pour introduire une différenciation de service en fonction du niveau de priorité (alloué statiquement) des flux. Toujours dans le contexte de l'AFDX, je me suis ensuite intéressé à une autre méthode d'évaluation des délais de bout-en-bout émergente, la méthode des trajectoires. Après avoir démontré formellement comment améliorer les bornes pire cas déterminées par la méthode des trajectoires, nous avons finalement prouvé à l'aide d'un contre-exemple que cette approche (même sans notre amélioration), peut déterminer dans des cas rares, des délais de bout-en-bout optimistes. C'est-à-dire inférieurs aux pire cas réels.

Nous avons, par la suite, cherché à développer une nouvelle approche pour le calcul de borne supérieure pire cas sur les délais de bout-en-bout de traversée du réseau AFDX par une trame. Ainsi, en se basant sur la similarité entre la transmission d'une trame par un composant réseau et l'exécution d'une tâche par un système non-préemptif mono-processeur, nous avons proposé l'approche *Forward end-to-end delay Analysis* (FA). Cette approche inspirée de l'ordonnancement temps-réel a été définie et formellement démontrée dans un contexte AFDX avec la politique de service FIFO. Nous avons amélioré les résultats obtenus avec FA en implémentant le paradigme de l'*Effet de sérialisation* bien connu en réseaux de paquets. FA a été appliquée sur une configuration industrielle réelle (composée de plus de 6000 flux) et nous avons comparé les délais déterminés avec FA à ceux obtenus avec la méthode des trajectoires corrigée et le Network Calculus. Ainsi, aucune des trois approches ne domine les autres dans le sens où chaque méthode obtient des délais plus longs pour certains flux par rapport aux deux autres et plus courts pour d'autres flux. Cependant, la méthode FA obtient en général et pour une très grande majorité des délais plus serrés que ceux obtenus par les deux autres approches.

Nous avons également étendu FA à la politique de service FP/FIFO pour répondre aux besoins des industriels d'intégrer des flux avec des contraintes temporelles différenciées dans un même réseau. Par exemple, dans un avion, des flux critiques avec des contraintes temporelles strictes (*p.ex.* les consignes des commandes de vol) partagent le même réseau que des flux sans contrainte temporelle (*p.ex.* les caméras). FA a également été étendue à d'autres domaines que l'avionique. Ainsi, FA a été appliquée à l'automobile et son protocole réseau *Audio/Vidéo Bridging Switched Ethernet* (AVB). AVB est un réseau de paquets standardisé, basé sur Ethernet. Il inclut la différenciation de service avec la politique d'ordonnancement FP/FIFO. Cependant, afin de limiter l'impact des flux de plus haute priorité, l'algorithme *Credit Based Shaping* (CBS) est ajouté. Cette extension au réseau AVB a été également formellement démontrée. Actuellement, nous finalisons l'extension de FA au protocole TSN pour des applications liées à l'industrie 4.0.

En parallèle, j'ai contribué au dimensionnement des files d'attente en nombre de trames. Ce sujet est très connexe à celui des délais de bout-en-bout car le délai subi dans un commutateur par une trame correspond principalement à son temps d'attente dans la file du port de sortie et est donc dépendant de la quantité de données (trames) présente dans la file à son arrivée. C'était un sujet ouvert pour lequel il y a très peu de travaux alors que c'est indispensable pour démontrer le déterminisme d'un réseau. Généralement, les approches existantes sont basées sur une approche d'évaluation des délais pire cas des flux. Elles déterminent l'arriéré pire cas présent dans la file d'attente en nombre de bits, à partir du plus long temps d'attente qu'elle construit. Cependant, nous avons démontré que le pire cas en nombre de trames ne correspond pas nécessairement à celui en nombre de bits. Nous avons développé une approche *Buffer Dimensionning per Frame* (BDF) qui nécessite en entrée : une topologie, un

contrat de trafic et une méthode d'évaluation pire cas des délais (p.ex. FA, Network Calculus...). Cette approche détermine une borne supérieure en nombre de trames de la taille des files d'attente. Cette approche après avoir été formellement démontrée a été appliquée sur une configuration industrielle.

1.4.2 Publications et productions scientifiques

Type de publication	Nombre
Chapitres d'ouvrage	5
Revue d'audience internationale avec comité de lecture	7(+1)
Conférences et workshops d'audience internationale avec comité de lecture	23
Conférences et workshops d'audience nationale avec comité de lecture	4
Séminaire et autres publications	1

TABLE 1.1 – Bilan récapitulatif des publications

1.4.3 Encadrement doctoral et scientifique

J'ai été l'encadrant principal de deux doctorants qui ont soutenu. Ces thèses se sont déroulées sur 36 mois exactement avec respectivement quatre et cinq publications. Les deux docteurs ont été embauchés directement après l'obtention de leur diplôme et sont devenus tous les deux ingénieurs dans le privé.

Je suis actuellement co-encadrant de trois doctorants.

Richard a démarré sa thèse au début de la pandémie de la COVID19. Il dispose d'une bourse institutionnelle. Il est actuellement en troisième année et travaille sur l'extension de la méthode FA au réseau TSN. Ses résultats obtenus sont prometteurs et il est actuellement en phase de rédaction d'articles.

Adrien dispose d'une allocation du CEA. Il symbolise nos collaborations sur la configuration et la validation des réseaux TSN. En particulier, il travaille sur la définition de solution autonome de gestion des réseaux déterministes en utilisant les techniques de l'Intelligence Artificielle. Ses résultats ont fait l'objet d'une publication en 2023.

Finalement, depuis octobre 2022, je suis co-encadrant de Luc Descharmes, qui travaille sur les délais de bout-en-bout de chaînes fonctionnelles dans l'AFDX, comprenant une partie ordonnancement temps réel pour les applications partitionnées sur les calculateurs et une partie délai sur la traversée du réseau.

Encadrement de thèses de doctorat en cours :

- **Luc DESCHARMES** (01/10/2022-)
Financement : Bourse institutionnelle de l'Université de Poitiers.
Co-encadrement : **30%**, sous la direction d'Annie GENIET.
Sujet : Délai de bout-en-bout des applications partitionnées communiquant sur réseaux commutés.
- **Adrien ROBERTY** (01/02/2021-)
Financement : allocation du CEA.
Co-encadrement : **30%** avec Siwar BEN HADJ SAID, sous la direction d'Annie GENIET.
Sujet : Modélisation, évaluation et validation des systèmes temps réel distribués.
- **Richard GARREAU** (01/10/2020-)
Financement : Bourse institutionnelle de l'Université de Poitiers.
Co-encadrement : **70%**, sous la direction de Pascal RICHARD.
Sujet : Réseaux embarqués hétérogènes.

Encadrement de thèses de doctorat soutenues :

- **Nassima BENAMMAR** (01/10/2015-17/09/2018)
Financement : Bourse institutionnelle de l'Université de Poitiers.
Co-encadrement : **50%** avec Henri BAUER, sous la direction de Pascal RICHARD.
Sujet : Modélisation, évaluation et validation des systèmes temps réel distribués.
Publications : 1 revue, 3 conférences internationales et 2 *workshops* internationaux.
Situation actuelle : Emploi permanent dans le secteur privé en qualité d'ingénieur.
- **Georges Arnaud KEMAYO** (01/10/2011-23/09/2014)
Financement : Bourse institutionnelle de l'Université de Poitiers.
Co-encadrement : **80%** avec Henri BAUER, sous la direction de Pascal RICHARD.
Sujet : Évaluation et validation des systèmes distribués avioniques.
Publications : 3 conférences internationales et 1 *workshop* international.
Situation actuelle : Emploi permanent dans le secteur privé en qualité d'ingénieur.

Participations ponctuelles à des thèses de doctorat extérieures :

- **Albekaye TRAORE**
sous la direction de Rodolphe VAUZELLE (Poitiers, XLIM-SIC) et Airbus
- **Xiaoting LI**
sous la direction de Christian FRABOUL et de Jean-Luc SCHARBARG (Toulouse, IRIT)
Publications liées aux travaux : 1 revue et 1 *workshop* d'audience internationale

Encadrement de stagiaires

- **Tanguy RELO** (05/06/2023-29/09/2023)
Encadrement : **100%**.
Sujet : *Implémentation d'un ordonnancement hiérarchisé en python*,
Stage Ingénieur (A2) de l'ISAE-ENSMA.
- **Thomas VILEO** (06/06/2022-30/09/2022)
Encadrement : **100%**.
Sujet : *Prise en main et développement d'une station sol de contrôle du drone TELLO de DJI*,
Stage Ingénieur (A2) de l'ISAE-ENSMA.

- **Luc DESCHARMES** (06/04/2021-30/09/2021)
Encadrement : **100%**.
Sujet : *Intégration de l'outil PyCPA à l'approche FA*,
Stage de Projet de fin d'études (A3) de l'ISAE-ENSMA.
- **Richard GARREAU** (01/04/2020-30/09/2020)
Co-encadrement : **50%** avec Henri BAUER.
Sujet : *Validation temporelle distribuée dans le contexte robotique*,
Stage de Master 2 de l'Université de Poitiers et poursuite en thèse.
- **Pierrick RICHARD** (18/06/2020–10/09/2020)
Co-encadrement : **50%** avec Henri BAUER.
Sujet : *Intégration de l'outil Python PyCPA à l'outil PyFA*,
Stage Ingénieur de l'ENSMA.
- **Abdelaziz AHMED NACER** (04/01/2011-08/07/2011)
Co-encadrement : **50%** avec Pascal RICHARD.
Sujet : *Ordonnancement dans les systèmes multiprocesseurs temps réels*
Stage de Master 2 de l'Université de Poitiers.
- **Patrick BAKABADIO** (01/02/2010-30/06/2010)
Encadrement : **100%**.
Sujet : *Etude du protocole FlexRay et de son application aux réseaux avioniques*
Stage de Master 2 de l'Université de Poitiers.

1.4.4 Diffusion et rayonnement

1.4.4.1 Projets

Projets déposés

- **ANR JCJC Avionics SURE REconfiguration (ASURE), 2013** : Ce projet "Jeunes chercheurs" a été déposé en 2013. Son objectif était de définir une méthode de reconfiguration à la volée du réseau AFDX à la suite d'une panne (p.ex. la perte d'un commutateur). Cette méthode devait également pouvoir servir d'aide à la configuration initiale du réseau pour le concepteur. Pour finir, un outil devait être développé afin d'implémenter cette approche ;
- **ANR PRC Probabilistic analysis of a real-time embedded network (PIANe), 2014** : L'objectif de ce projet déposé en collaboration avec l'IRIT et Jean-Luc Scharbarg était le développement d'une méthode d'analyse probabiliste du réseau AFDX. Ainsi, en se basant notamment sur mes travaux sur le *Stochastic Network Calculus*, l'objectif était de pouvoir déterminer une borne sur le délai de bout-en-bout de traversée du réseau par une trame en termes de probabilité de dépassement ;
- **ANR PRCE Smart & Safe TSN traffic scheduling for OPC UA (TSAR), 2021** : Pour ce projet, nous nous sommes associés au *CEA-LIST* et nous avons constitué un consortium avec deux partenaires privés *Systerel* et *Safe Connect Systems (SCS)*. L'objectif de ce projet était d'utiliser des techniques de *Machine Learning* développées par le CEA afin de générer des

configurations valides de réseaux TSN basées sur les commutateurs développés par SCS. La validation des configurations repose sur les méthodes formelles développées par Systerel et sur nos approches de calculs de délai de bout-en-bout pire cas ;

- **ANR PRCE *Smart & Safe TSN traffic scheduling for OPC UA (TSAR2)*, 2022** : Ce projet est une mise à jour du projet présenté l'année précédente avec le retrait de SCS pour interruption d'activités. Nous avons apporté des modifications à la vue des retours. L'infrastructure réseau cible est essentiellement filaire avec quelques liens terminales Wifi et potentiellement une connexion 5G. Ce projet a reçu le soutien d'Intel France, Schneider Electric et Thalès ;

Participation à des projets

- **CORAC PANDA** (*Plateforme AvioNique moDulAire étendue*) : Ce projet est de type *Programme d'Investissements d'Avenir (PIA)* avec un budget de 149000€ ayant pour clients : la Direction générale de l'armement (DGA), la Direction générale de l'aviation civile (DGAC) et l'ONERA ; le coordinateur est THALES et il associe Airbus, DASSAULT Aviation, Airbus Helicopters et SAFRAN. Plusieurs partenaires sont associés parmi lesquels l'ISAE-ENSMA et l'IRIT. Dans ce projet, nous avons travaillé à définir une méthode d'analyse temporelle des applications avioniques AME (Avionique Modulaire Étendue), qui permette d'inter-opérer avec d'autres méthodes d'analyse autour d'un modèle commun de représentation d'une architecture AME. Nos travaux sur la définition et la démonstration de notre approche *Forward end-to-end delay Analysis (FA)* dans un contexte AFDX sont au cœur de nos contributions sur ce projet. Trois rapports ont été délivrés dont le dernier d'environ 200 pages.
- **ADCN+ et ADCN+2** (*Advanced Data Communication Network +*) : L'objectif de ces projets, qui se sont déroulés sur la période 2006-2010, est d'analyser le réseau "nouvelle génération" du système avionique embarqué dans les avions d'Airbus et notamment, l'A380, afin de proposer des possibilités d'évolution. Les principaux axes abordés sont : le modèle, l'architecture, l'analyse de performances, et la supervision. Ce projet de recherche implique Airbus et trois laboratoires : IRIT, ONERA-CERT et le LAAS. Mes travaux sur le Stochastique Network Calculus s'inscrivent dans ce projet et notamment dans la partie *analyse de performances*. Nous avons rédigé deux rapports (intermédiaire puis final) intitulé *Outils d'évaluation de borne maximale de délais des différents éléments de l'architecture de référence*.
- **Workload For Labs (W4L)** : L'objectif de ce projet impliquant notamment le LISI (ex LIAS) et l'Université de Tours est de proposer des solutions aux laboratoires pharmaceutiques pour le traitement d'échantillons par des machines à traitement par lots (*p.ex.* centrifugeuses).

1.4.4.2 Organisation de Conférences

Président du comité d'organisation de l'École d'été Temps Réel, (ETR 2021)	2021
Co-General Chair de la 26 ^{ième} Conférence internationale <i>Real-Time Networks and Systems</i> (RTNS 2018)	2018
Membre du comité d'organisation de la 7th IFAC International Conference on Fieldbuses Networks in Industrial Embedded Systems (FET 2007), Toulouse	2007-2008
Membre du comité d'organisation et responsable du site internet de la conférence <i>14th international Conference on Real-Time and Network Systems (RTNS 2006)</i>	2005-2006
Membre du comité d'organisation de la conférence FuturView 2003	2002-2003

1.4.4.3 Participation à la communauté temps réel

Membre du comité de programme de la 29th International Conference on Real-Time and Network Systems (RTNS 2021)	2021
Membre du comité de programme de l' <i>International Workshop on Real-Time Networks</i> (RTN 2017)	2017
Membre du comité de programme de la conférence <i>IEEE International Workshop on Factory Communication Systems</i> (WFCS 2014)	2014
Membre du comité de programme de la session Junior Workshop de la 16th International Conference on Real-Time and Network Systems (RTNS 2008)	2007-2008
Membre du comité de programme de la session Junior Workshop de la 15th International Conference on Real-Time and Network Systems (RTNS 2007)	2006-2007
Membre du comité de programme des Premières Rencontres des Jeunes Chercheurs en Informatique Temps Réel 2005 (RJCITR 2005)	2004-2005
Activités de relecture pour des revues ou conférences (RTSS, RTNS, ETFA, RTN, WFCS, Elsevier, RTS Journal...)	

1.4.4.4 Jurys de thèse extérieurs

Participation à des jurys de thèses, en tant qu'examineur :

- **Aakash SONI** (05/06/2020)
Direction : Jean-Luc SCHARBARG et Jérôme ERMONT.
Sujet : *Real-time performance analysis of a QoS based industrial embedded network*, Institut National Polytechnique de Toulouse.
- **Xiaoting LI** (19/09/2013)
Direction : Jean-Luc SCHARBARG et Christian FRABOUL.
Sujet : *Worst-case delay analysis of real-time switched Ethernet networks with flow local synchronization*, Institut National Polytechnique de Toulouse.

1.4.4.5 Autres activités de vulgarisation

Journées Portes Ouvertes de l'Université de Poitiers, présentation de l'IUT et du département GEII	2010-2019
Visite de lycées pour la promotion de l'IUT et du département GEII	2014-2019
Responsable du site internet du LISI	2003-2007
Journées Portes Ouvertes de l'Université de Poitiers , présentation des différentes filières informatiques	2004-2005
Journées Portes Ouvertes de l'Université de Poitiers , présentation d'un projet réalisé en maîtrise d'informatique	2001-2002

1.4.5 Responsabilités scientifiques

Depuis 2017, je suis responsable de l'équipe **Systèmes Embarqués Temps Réel** du LIAS. C'est une équipe composée de trois Professeurs d'Université, de sept Maîtres de Conférences (dont 1 HDR), et en moyenne de huit doctorants (et post-doctorants).

En 2021, en tant que responsable de l'équipe, j'ai constitué le dossier de l'équipe Systèmes Embarqués Temps Réel dans le cadre de l'évaluation du LIAS par le **Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur** (Hcéres). J'ai présenté le bilan de l'équipe et ses perspectives durant l'audition. Au final, le rapport d'évaluation a été très positif pour l'équipe en soulignant notamment la très bonne qualité scientifique et en ayant salué son animation.

J'ai également intégré le Conseil Scientifique du LIAS en 2011 puis le Conseil de l'Unité de Recherche en 2022 et le Conseil Scientifique de l'ENSMA.

1.4.6 Responsabilités et mandats locaux ou régionaux

Direction de l'équipe <i>Systèmes Embarqués Temps Réel</i>	<i>2017-Présent</i>
Membre du Conseil Scientifique de l'ISAE-ENSMA	<i>2021-Présent</i>
Membre du Conseil de l'Unité de Recherche (CUR) du LIAS	<i>2022-Présent</i>
Membre du Membre de la commission d'admission L3/M1 à l'ISAE-ENSMA	<i>2022-Présent</i>
Membre du Conseil Scientifique du LIAS	<i>2011-2022</i>
Membre de la Commission d'Expertise Scientifique CNU 27 de l'Université de Poitiers	<i>2013-2018</i>
Membre du Conseil du Département GEII	<i>2012-2014</i>
Représentant de l'IUT de Poitiers pour le Certificat Informatique et Internet (C2I) de l'Université de Poitiers	<i>2009-2016</i>
Chef de projet pour le développement d'un système embarqué	<i>2008-2009</i>
Représentant des doctorants au sein du Conseil de l'École Doctorale (ED-SPI&A)	<i>2004-2005</i>
Représentant des doctorants au sein du Conseil d'Administration de l'ENSMA	<i>2004-2005</i>
Représentant des doctorants au sein du Conseil Scientifique de l'ENSMA	<i>2004-2005</i>

1.5 Annexes

1.5.1 Tableau des enseignements

Année	Niveau	Diplôme	Intitulé	Type de formation	Nature	Effectifs	Volume horaire annuel
2023	A1-L3	ENSMA	Systèmes Embarqués	FISE	Cours	170	18,75 hTD
2023	A1-L3	ENSMA	Systèmes Embarqués	FISE	TD	30	42,5 hTD
2023	A1-L3	ENSMA	Systèmes Embarqués	FISE	TP	12	45 hTD
2023	A3-M2	ENSMA	Systèmes Avioniques	FISE	Cours	30	3,75 hTD
2023	A3-M2	ENSMA	Systèmes Avioniques	FISE	TD	30	5 hTD
2023	A2-M1	ENSMA	Projet Avionique	FISE	Projet	12	24 hTD
2023	A1-L3	ENSMA	Défi 3S ¹	FISE	Projet	12	34 hTD
2022	A2-M1	ENSMA	Automatique	FISE	TD	30	25 hTD
2022	A2-M1	ENSMA	Automatique	FISE	TP	12	81 hTD
2022	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	Cours	21	12 hTD
2022	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	TD	21	14 hTD
2022	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	TP	11	12 hTD
2022	A1-L3	ENSMA	Systèmes Embarqués 1	FISE	Cours	170	18,75 hTD
2022	A1-L3	ENSMA	Systèmes Embarqués 1	FISE	TD	30	47,5 hTD
2022	A1-L3	ENSMA	Systèmes Embarqués 1	FISE	TP	12	78 hTD
2022	A3-M2	ENSMA	Systèmes Avioniques	FISE	Cours	30	3,75 hTD
2022	A3-M2	ENSMA	Systèmes Avioniques	FISE	TD	30	5 hTD
2022	A2-M1	ENSMA	Projet Avionique	FISE	Projet	12	21 hTD
2022	A1-L3	ENSMA	Défi 3S ²	FISE	Projet	12	24 hTD
2022	A1-L3	ENSMA	Math. Appliquées	FISA	Cours	18	21 hTD
2022	A2-M1	ENSMA	Stage Ingénieur	FISE	Soutenance	1	12 hTD
2022	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Soutenance	1	22 hTD
2022	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Suivi	1	9 hTD
2021	A2-M1	ENSMA	Automatique	FISE	TD	30	15 hTD
2021	A2-M1	ENSMA	Automatique	FISE	TP	12	60 hTD
2021	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	Cours	18	12 hTD
2021	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	TD	18	14 hTD
2021	A1-L3	ENSMA	Archi. des Syst. Num.	FISA	TP	10	12 hTD
2021	A3-M2	Master2	Systèmes Avioniques	FISE	Cours	35	3,75 hTD
2021	A3-M2	Master2	Systèmes Avioniques	FISE	TD	35	5 hTD
2021	A3-M2	ENSMA	Systèmes Avioniques	FISE	Cours	30	3,75 hTD
2021	A3-M2	ENSMA	Systèmes Avioniques	FISE	TD	30	5 hTD

Année	Niveau	Diplôme	Intitulé	Type de formation	Nature	Effectifs	Volume horaire annuel
2021	A2-M1	ENSMA	Projet Avionique	FISE	Projet	12	33 hTD
2021	A2-M1	ENSMA	Systèmes Embarqués 2	FISE	TD	35	7,5 hTD
2021	A2-M1	ENSMA	Systèmes Embarqués 2	FISE	TP	12	42 hTD
2021	A2-M1	ENSMA	Stage Ingénieur	FISE	Soutenance	1	12 hTD
2021	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Soutenance	1	22 hTD
2021	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Suivi	1	9 hTD
2020	A2-M1	ENSMA	Automatique	FISE	TP	12	36 hTD
2020	A1-L3	ENSMA	ISE ³	FISE	Cours	30	9,375 hTD
2020	A1-L3	ENSMA	ISE	FISE	TD	30	30 hTD
2020	A1-L3	ENSMA	ISE	FISE	TP	12	66 hTD
2020	A2-M1	ENSMA	Systèmes Embarqués 2	FISE	TD	35	7,5 hTD
2020	A1-L3	ENSMA	UED ⁴	FISE	TD	30	12,5 hTD
2020	A2-M1	ENSMA	Stage Ingénieur	FISE	Soutenance	5	10 hTD
2020	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Soutenance	5	10 hTD
2020	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Suivi	1	9 hTD
2019	A2-M1	ENSMA	Automatique	FISE	TP	12	33 hTD
2019	A1-L3	ENSMA	ISE	FISE	TD	30	35 hTD
2019	A1-L3	ENSMA	ISE	FISE	TP	12	57 hTD
2019	A2-M1	ENSMA	Stage Ingénieur	FISE	Soutenance	1	16 hTD
2019	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Soutenance	1	12 hTD
2019	A3-M2	ENSMA	Projet de Fin d'Études	FISE	Suivi	1	5 hTD

Ce tableau liste la totalité des enseignements que j'ai effectués depuis mon recrutement à l'ISAE-ENSMA. Cependant, pour l'année 2019-2020, j'ai subi en début d'année 2020, une opération avec un arrêt de trois mois.

Dans le tableau ci-dessous, j'ai résumé ma dernière année 2018-2019, représentative de mes dix années passées à l'IUT de Poitiers.

-
1. Défi Signal-Système-Simulation.
 2. Introduction aux Systèmes Embarqués.
 3. Utilisation et Exploitation des Données

Année	Niveau	Diplôme	Intitulé	Type de formation	Nature	Effectifs	Volume horaire annuel
2018	DUT1-L1	DUT-GEII	Soutien	FISA	TD	18	10 hTD
2018	DUT1-L1	DUT-GEII	Mathématiques1	FISA	Cours	18	18 hTD
2018	DUT1-L1	DUT-GEII	Mathématiques1	FISA	TD	18	40 hTD
2019	DUT2-L3	DUT-GEII	Encadrant Apprenti	FISA	Suivi	1	10 hTD
2018	DUT1-L1	DUT-GEII	Mathématiques1	FISE	Cours	100	10,5 hTD
2018	DUT1-L1	DUT-GEII	Mathématiques1	FISE	TD	35	24 hTD
2019	DUT1-L1	DUT-GEII	Mathématiques2	FISE	TD	35	24 hTD
2019	DUT2-L2	DUT-GEII	MCM1	FISE	Cours	25	12 hTD
2019	DUT2-L2	DUT-GEII	MCM1	FISE	TD	25	12 hTD
2019	DUT1-L1	DUT-GEII	Outils Logiciel 1	FISE	TP	18	12hTD
2019	DUT1-L1	DUT-GEII	Outils Logiciel 2	FISE	TP	18	12hTD
2019	A3-M2	ENSMA	Systèmes Avioniques	FISE	Cours	30	3,75 hTD
2019	A3-M2	ENSMA	Systèmes Avioniques	FISE	TD	30	5 hTD
2019	DUT2-L2	DUT-GEII	Base de Données	FISE	TD	30	10,5 hTD
2019	DUT2-L2	DUT-GEII	Base de Données	FISE	TP	30	16 hTD
2019	DUT2-L2	DUT-GEA	Base de Données	FISE	TD	35	80 hTD
2019	DUT2-L2	DUT-GEII	Stage de fin d'études	FISE	Soutenance	1	2 hTD
2019	DUT2-L2	DUT-GEII	Stage de fin d'études	FISE	Suivi	1	2 hTD
2019	DUT1-L2	DUT-GEII	Projet Tutoré	FISE	Projet	35	4,25 hTD
2019	DUT1-L2	DUT-GEII	Conduite de Projet	FISE	Cours	100	9 hTD
2019	DUT1-L2	DUT-GEII	Conduite de Projet	FISE	TP	35	20 hTD

1.5.2 Liste classée des publications

La liste exhaustive de toutes mes publications, triée par type :

Chapitres d'ouvrage

- [CO-1] Yassine OUHAMMOU, **Frédéric RIDOUARD**, Emmanuel GROLLEAU, Mathieu JAN, Moris BEHNAM, Proceedings of the 26th International Conference on Real-Time Networks and Systems (RTNS 2018), ACM ICPS, 2018, pp. 275
- [CO-2] Emmanuel GROLLEAU, Michael RICHARD, Pascal RICHARD, **Frédéric RIDOUARD**, Ordonnancement temps réel - Ordonnancement monoprocesseur, Techniques de l'ingénieur (T.I.), edited by Editions T.I., 2013, pp. 1-30
- [CO-3] Emmanuel GROLLEAU, Michael RICHARD, Pascal RICHARD, **Frédéric RIDOUARD**, Ordonnancement temps réel - Ordonnancement réparti, Techniques de l'ingénieur (T.I.), edited by Editions T.I., 2013, pp. 1-27
- [CO-4] Emmanuel GROLLEAU, Michael RICHARD, Pascal RICHARD, **Frédéric RIDOUARD**, Ordonnancement temps réel - Ordonnancement multiprocesseur, Techniques de l'ingénieur (T.I.), edited by Editions

T.I., 2013, pp. 1-20

- [CO-5] Pascal RICHARD, **Frédéric RIDOUARD**, Systèmes Temps Réel 2 : Ordonnancement, Réseaux, Qualité de Service, Chapitre : "Ordonnancement monoprocesseur" *Hermes Science Publications*, ISBN :2746213-044 (2006)

Revues d'audience internationale avec comité de lecture

- [ACL-1] Richard GARREAU, Nassima BENAMMAR, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Buffer Dimensioning per Frames in packet networks, *IEEE Transactions on Industrial Informatics*, [**SJR Q1** ; **IF=8.59**], (*en cours de soumission*)
- [ACL-2] Nassima BENAMMAR, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Forward End-to-End Delay for AFDX Networks, *IEEE Transactions on Industrial Informatics (IEEE/TII 2018)*, vol. 14, 2018, pp. 858–865 [**SJR Q1** ; **IF=8.59**]
- [ACL-3] Jian-Jia CHEN, Geoffrey NELISSEN, Wen-Hung HUANG, Maolin YANG, Bjorn BRANDENBURG, Konstantinos BLETSAS, Cong LIU, Pascal RICHARD, **Frédéric RIDOUARD**, Neil AUDSLEY, Raj RAJKUMAR, Dionisio DE NIZ, Georg VON DER BRUGGEN, Many suspensions, many problems : a review of self-suspending tasks in real-time systems, *Real-Time Systems (RTS 2018)*, Sep, 2018 [**SJR Q2** ; **IF=2.37**]
- [ACL-4] Yassine OUHAMMOU, Anh Toan BUI LONG, Emmanuel GROLLEAU, Henri BAUER, **Frédéric RIDOUARD**, Pascal RICHARD, Michael RICHARD, A Model-based Process for the Modeling and the Analysis of Avionic Architectures, *International Journal of Intelligent Information and Database Systems (IJIDS)*, 2017 [**SJR Q4**]
- [ACL-5] Xiaoting LI, Jean-Luc SCHARBARG, Christian FRABOUL, **Frédéric RIDOUARD**, Existing offset assignments are near optimal for an industrial AFDX network, *ACM SIGBED - Embedded Systems (ACM SIGBED 2011)* [**SJR Q2**]
- [ACL-6] Jean-Luc SCHARBARG, **Frédéric RIDOUARD**, Christian FRABOUL, A probabilistic analysis of end-to-end delays on an AFDX avionic network, *IEEE Transactions on Industrial Informatics*, vol. 5, n. 1, Février, 2009, pp. 38-49 [**SJR Q1** ; **IF=3.05**]
- [ACL-7] **Frédéric RIDOUARD**, Pascal RICHARD, Patrick MARTINEAU, On-line scheduling on a batch processing machine with unbounded batch size to minimize the makespan, *European Journal of Operational Research (EJOR 2008)*, vol. 189, n. 3, 2008, pp. 1227-1242 [**SJR Q1** ; **IF=2.25**]
- [ACL-8] **Frédéric RIDOUARD**, Pascal RICHARD, Francis COTTET, Karim TRAORE, Some results on scheduling tasks with self-suspensions, *Journal of Embedded Computing*, 2006, vol. 2, pp. 301-312

Conférences et workshops d'audience internationale avec comité de lecture

- [ACTI-1] Richard GARREAU, Matheus LADEIRA, Emmanuel GROLLEAU, Henri BAUER, **Frédéric RIDOUARD**, Pascal RICHARD, Link between real-time scheduling and time-triggered networks, *Proceedings of the 44th International Conference on Real-Time Systems Symposium (RTSS 2023)*, *accepté, à paraître*, **Core A***
- [ACTI-2] Adrien ROBERTY, Siwar BEN HADJ SAID, **Frédéric RIDOUARD**, Henri BAUER, Annie GENIET, Reinforcement Learning for Time-Aware Shaping (IEEE 802.1Qbv) in TimeSensitive Networks, *Proceedings of the Work-In-Progress session of IEEE International Conference on Emerging Technologies and Factory Automation (ETF A 2023)*

- [ACTI-3] Nassima BENAMMAR, Henri BAUER, **Frédéric RIDOUARD**, Pascal RICHARD, Timing analysis of AVB Ethernet network using the Forward end-to-end Delay Analysis, Proceedings of the 26th International Conference on Real-Time Networks and Systems (RTNS 2018), ACM, 2018, pp. 223–233 [**Equiv. Core B**]
- [ACTI-4] Nassima BENAMMAR, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Forward end-to-end delay Analysis extension for FP/FIFO policy in AFDX networks, 22th IEEE International Conference on EMerging Technologies and Factory Automation (ETFA 2017), edited by IEEE, 2017
- [ACTI-5] Nassima BENAMMAR, Henri BAUER, **Frédéric RIDOUARD**, Pascal RICHARD, Tighter buffer dimensioning in AFDX networks, The 14th International Workshop on Real-Time Networks in conjunction with the 28th Euromicro International Conference on Real-Time Systems (RTN 2016), edited by SIGBED review, 2016
- [ACTI-6] Nassima BENAMMAR, Henri BAUER, **Frédéric RIDOUARD**, Pascal RICHARD, Buffer Dimensioning based on FA Approach in the AFDX Context, Real-Time Networks Seminar (RTN Seminar -FNRS), FNRS group contact, 2016
- [ACTI-7] Georges KEMAYO, Nassima BENAMMAR, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Improving AFDX End-to-End delays analysis, 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2015), edited by IEEE, 2015
- [ACTI-8] Georges KEMAYO, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, A Forward end-to-end delays Analysis for packet switched networks, 22nd International Conference on Real-Time Networks and Systems (RTNS 2014), edited by ACM, 2014rtns [**Equiv. Core B**]
- [ACTI-9] Georges KEMAYO, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Optimistic problems in the trajectory approach in FIFO context, 18th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2013), edited by IEEE, 2013
- [ACTI-10] Georges KEMAYO, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Optimism due to serialization in the trajectory approach for switched Ethernet networks, 7th Junior Researcher Workshop on Real-Time Computing (JRWRTC 2013), Sophia Antipolis, France, 2013
- [ACTI-11] Joël GOOSSENS, Pascal RICHARD, Markus LINDSTROM, Irina LUPU, **Frédéric RIDOUARD**, Job partitioning strategies for multiprocessor scheduling of real-time periodic tasks with restricted migration, Int. Conf. on Real-Time and Networked Systems (Nancy), 2012, edited by ACM, 2012 [**Equiv. Core B**]
- [ACTI-12] Georges KEMAYO, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Scheduling analysis to enhance the trajectory approach, 6th Junior Researcher Workshop on Real-Time Computing (JRWRTC 2012), Novembre, 2012
- [ACTI-13] Pascal RICHARD, Georges KEMAYO, **Frédéric RIDOUARD**, Emmanuel GROLLEAU, Thi Huyen Chau NGUYEN, Response Time bounds for Static-Priority Tasks and Arbitrary Relative Deadlines with Resource Augmentation, IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2012), edited by IEEE, 2012
- [ACTI-14] Xiaoting LI, Jean-Luc SCHARBARG, Christian FRABOUL, **Frédéric RIDOUARD**, Existing offset assignments are near optimal for an industrial AFDX network, RTN - The 10th International Workshop on Real-Time Networks (RTN 2011), July, 2011
- [ACTI-15] **Frédéric RIDOUARD**, Pascal RICHARD, Scheduling of self-suspending tasks : state of art and new insights, 1st International Real-Time Scheduling Open Problems Seminar (RTSOPS 2010), July, 2010
- [ACTI-16] **Frédéric RIDOUARD**, Jean-Luc SCHARBARG, Christian FRABOUL, Probabilistic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network, 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), vol. 1, Hambourg (Allemagne), Septembre, 2008

- [ACTI-17] **Frédéric RIDOUARD**, Jean-Luc SCHARBARG, Christian FRABOUL, Stochastic network calculus for buffer overflow evaluation in an avionics switched Ethernet, 1st Junior Researcher Workshop on Real-Time Computing (JRWRTC 2007), Nancy, March 2007
- [ACTI-18] **Frédéric RIDOUARD**, Jean-Luc SCHARBARG, Christian FRABOUL, Stochastic Network Calculus for end-to-end delay evaluation of avionics multi-hop virtual links, 7th IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems, vol. 1, Toulouse (France), Novembre, 2007
- [ACTI-19] **Frédéric RIDOUARD**, Jean-Luc SCHARBARG, Christian FRABOUL, Stochastic Network Calculus for end-to-end delays distribution evaluation on an avionics switched Ethernet, 5th IEEE International Conference on Industrial Informatics (INDIN 2005), vol. 1, Vienne (Autriche), Juillet, 2007, pp. 559-564
- [ACTI-20] **Frédéric RIDOUARD**, Pascal RICHARD, Worst-case analysis of feasibility tests for self-suspending tasks, proc. 14th Real-Time and Network Systems (RTNS 2006), Poitiers, 2006 [**Equiv. Core B**]
- [ACTI-21] **Frédéric RIDOUARD**, Pascal RICHARD, Patrick MARTINEAU, On-line minimization of makespan for single batching machine scheduling problems, Euro Workshop on Project Management and Scheduling (PMS 2004), 2004
- [ACTI-22] **Frédéric RIDOUARD**, Pascal RICHARD, Francis COTTET, Negative results for scheduling independent hard real-time tasks with self-suspensions, The 25th IEEE International Real-Time Systems Symposium (RTSS 2004), edited by IEEE, December, 2004 [**Core A***]
- [ACTI-23] Pascal RICHARD, **Frédéric RIDOUARD**, Patrick MARTINEAU, On-line scheduling on a single batching machine to minimize the makespan, 6th Int. Conference on Industrial Engineering and Production Management (IEPM 2003)

Conférences et workshops d'audience nationale avec comité de lecture

- [ACT-1] Adrien ROBERTY, **Frédéric RIDOUARD**, Henri BAUER, Siwar BEN HADJ SAID, Annie CHOQUET, Deep Q-learning pour la gestion de réseaux déterministes, École d'été Temps- Réel ETR2021, Poitiers, 2021
- [ACT-2] Richard GARREAU, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Dimensionnement des buffers en nombre de trames dans l'Ethernet commuté, École d'été Temps-Réel ETR2021, Poitiers, 2021
- [ACT-3] **Frédéric RIDOUARD**, Jean-Luc SCHARBARG, Christian FRABOUL, Stochastic upper bounds for heterogeneous flows using a Static Priority Queueing on an AFDX network, Journées FAC, 2008
- [ACT-4] **Frédéric RIDOUARD**, Pascal RICHARD, Francis COTTET, Ordonnancement de tâches indépendantes avec suspension, 13th INTERNATIONAL CONFERENCE ON Real-Time Systems (RTS 2005), Avril, 2005
- [ACT-5] Stéphane JEANNENOT, Pascal RICHARD, **Frédéric RIDOUARD**, Ordonnancement temps réel avec profils variables de consommation d'énergie, proc. Real-Time Embedded Systems (RTS 2004), 2004
- [ACT-6] **Frédéric RIDOUARD**, Pascal RICHARD et Francis COTTET, Algorithmes d'ordonnancement en-ligne à une machine, *École d'Automne de Recherche Opérationnelle octobre 2003, Tours*

Rapports de recherche

- [RR-1] Richard GARREAU, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Analyse pire cas des délais de traversée et des tailles de files d'attente dans les réseaux embarqués (Ethernet commuté), 2020

[RR-2] Nassima BENAMMAR, **Frédéric RIDOUARD**, Henri BAUER, Pascal RICHARD, Forward End-To-End delay Analysis for AFDX networks, 2017

Séminaire et autre publications

[SAP-1] Présentation de l'Ordonnancement de tâches temps réel avec suspension lors de la Journée couplée ORDO/GOThA du 28/01/2005, Paris

Thèse

[THE-1] **Frédéric RIDOUARD**, Contributions à des problèmes d'ordonnancement en-ligne : l'ordonnancement temps réel de tâches à suspension et l'ordonnancement par une machine à traitement par lot, Thesis, Science pour l'Ingénieur et Aéronautique, november, 2006

Deuxième partie

Contexte et motivations

Réseau temps réel

2.1 Système temps réel

Un système temps réel est un système informatique réactif. Il est composé d'un ensemble de programmes soumis constamment à des sollicitations [HP85]. Un système temps réel contrôle un procédé qui évolue dans un environnement. Le système récupère des informations sur l'état du procédé via un ensemble de capteurs. Il effectue des calculs et agit sur ce procédé via des actionneurs. Le système doit conserver le procédé contrôlé dans un état de fonctionnement normal.

Un système temps réel est un système soumis à des contraintes temporelles et il existe, dans la littérature, plusieurs définitions d'un système temps réel (*p.ex.* [LO11]; [Gro14]). Nous retenons celle de J.A. STANKOVIC :

Définition 2.1

Un système temps réel [Sta88] est un système pour lequel sa correction ne dépend pas seulement des résultats mais également du temps auquel ils sont fournis.

Les systèmes temps réel sont utilisés dans nombre de domaines tels que le transport (*p.ex.* l'avionique ou l'automobile), la production industrielle mais également dans les applications du quotidien comme le multimédia, la téléphonie mobile, la domotique, etc. Des exemples sont détaillés dans [SLT05].

Selon les conséquences de son non-respect, les contraintes temporelles d'un système temps réel sont classifiées en différentes catégories [Emm+14] :

- **relative** : des fautes temporelles sont tolérables (*p.ex.* les jeux vidéo ou la téléphonie mobile). Un système composé uniquement de contraintes temporelles relatives est appelé *Système temps réel mou* ;
- **stricte** : aucune faute temporelle n'est tolérable pour le fonctionnement du procédé contrôlé (*p.ex.* les commandes de vol d'un système avionique). Un système composé uniquement de contraintes temporelles strictes est appelé *Système temps réel dur ou critique* ;
- **ferme** : des fautes temporelles sont acceptables dans une certaine mesure (*p.ex.* vidéo à la demande). Au-delà d'un certain seuil, le comportement du système est trop dégradé.

Remarque 2.1

Les systèmes temps réel sont, en général, découpés en un ensemble de sous-systèmes avec des ni-

veaux de criticité différents. Ils sont donc composés simultanément de contraintes strictes, fermes et relatives. Par exemple, pour le développement d'un système avionique, la norme *Software considerations in airborne systems and equipment certification (DO-178C)* [RTC11] spécifie cinq niveaux de criticité différents. Ainsi, à chaque sous-système est attribué un niveau de Design Assurance Level (DAL) entre A (une faute peut contribuer à une panne catastrophique) et E (une faute n'entraîne pas de conséquence sur la capacité opérationnelle de l'avion).

Une erreur commune est de considérer qu'un système temps réel est un système *rapide*. En effet, le système évolue à la vitesse du procédé qu'il contrôle. Ainsi, l'unité de temps est de l'ordre de la microseconde pour des systèmes critiques tels que ceux de l'avionique mais aussi de l'ordre de l'heure voire du jour pour le contrôle de réactions chimiques industrielles. Ainsi, un système temps réel n'est pas un système rapide mais un système qui effectue la bonne action au bon moment.

2.2 Système temps réel distribué

A partir des années 80, la généralisation de l'informatique nécessite une évolution rapide des technologies. Cette évolution impacte également les systèmes temps réels. En effet, ils doivent intégrer de plus en plus de fonctions et qu'elles soient de plus en plus complexes. Elles nécessitent plus de données échangées et plus de puissance de calculs. Prenons l'exemple de l'évolution des besoins en termes de volume logiciel, puissance de calculs et en nombre de bus numériques pour les avions développés par Airbus, comme illustrée par la Fig. 2.1. Cette évolution est exponentielle, passant par exemple d'un volume de quelques Méga-Octets au début des années 1970 à plusieurs Giga-octets pour l'A380 au tout début de années 2000.

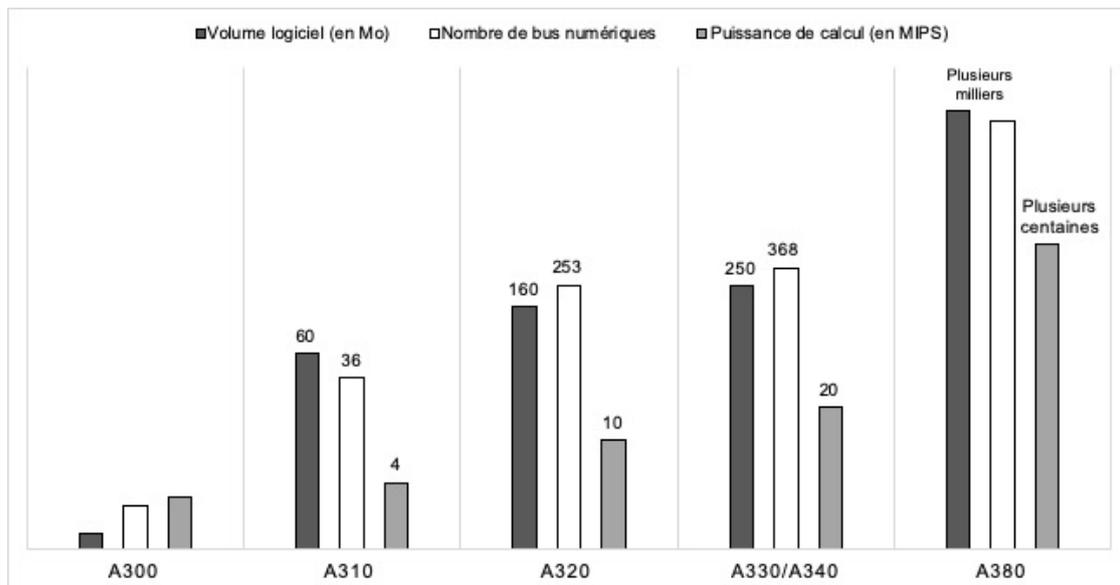


FIGURE 2.1 – Évolution des systèmes avioniques d'Airbus.

Les architectures jusque là utilisées (centralisées, fédérées ou réparties) sont inadéquates pour ré-

pondre à ces nouveaux besoins. La solution réside dans le partage de ressources. Ainsi, les systèmes embarqués temps réel sont distribués. Ils sont définis par un ensemble de calculateurs reliés entre eux par un médium de communication à hautes performances. Nous détaillons ci-après l'exemple, dans le domaine de l'avionique, de l'architecture Modulaire Intégrée ou IMA, puis celui du domaine de l'automobile.

2.2.1 Architecture IMA

Cette Architecture est issue des projets européens de recherche *Process for Advanced Management of End of life Aircraft* (PAMELA) [HP01] et *Validation platform for Integration of standard Components, Technologies and tools in an Open, modulaR and Improved Aircraft electronic system* (VICTORIA) [vic04]. L'architecture IMA a été standardisée par la norme ARINC 651 [ARI97]. L'IMA relie un ensemble de calculateurs via un réseau. L'objectif de l'IMA est le partage des ressources (réseau, temps processeur, mémoire, etc.) entre les fonctions avioniques.

Dans l'IMA utilisée par Airbus dans ses avions, les fonctions avioniques sont hébergées à l'intérieur de partitions Fig. 2.2 sur des calculateurs appelés aussi *Core Processing Module* (CPM ou CPIOM).

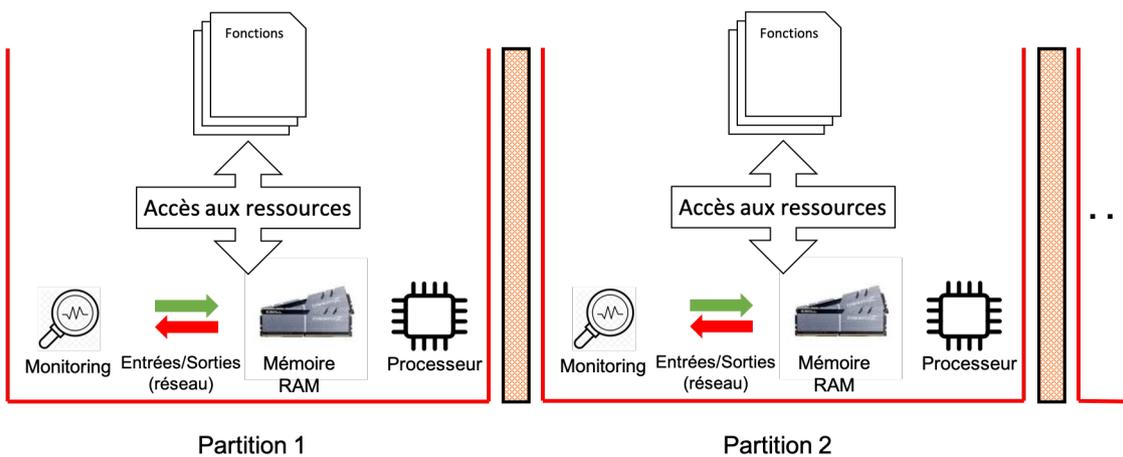


FIGURE 2.2 – Partitionnement d'un CPM.

Chaque partition possède des ressources garanties qui lui sont propres : temps processeur, mémoire RAM, accès au réseau et monitoring pour surveiller son fonctionnement ainsi que celui des fonctions qu'elle héberge. L'accès aux ressources entre les fonctions hébergées par la partition est concurrentiel. De plus, les partitions offrent un isolement de ses fonctions par rapport à celles des autres partitions garantissant ainsi l'accès à ses ressources.

D'un point de vue réseau, l'IMA se base sur un réseau cœur à hautes performances dénommé aussi *Aircraft Data Communication Network* (ADCN). Par exemple, chez Airbus l'ADCN est généralement de l'*Avionics Full Duplex switched ethernet* (AFDX) [66402] alors que chez Boeing, c'est de l'ARINC 629 [Isi10]; [Moo89]. A ce réseau sont reliés d'autres protocoles réseaux (CAN, ARINC 429, etc.).

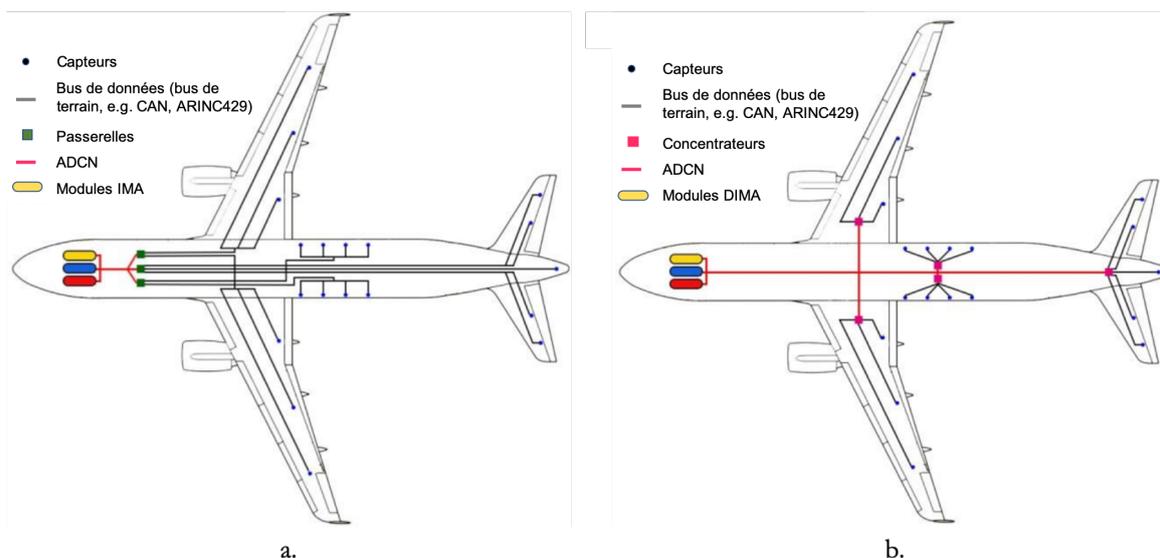


FIGURE 2.3 – Architecture Modulaire intégrée de l’A380 (a.) et celle distribuée de l’A350 (b.)[MOG18].

La Fig. 2.3.a. présente l’architecture déployée par Airbus dans son A380 et aussi A400M. Nous y retrouvons le réseau cœur ADCN (qui est de l’AFDX) auquel sont connectés les modules IMA (dont les calculateurs) et les autres protocoles réseaux. Cette architecture a évolué avec l’A350 en limitant la présence des autres réseaux au profit d’une généralisation de l’ADCN (cf. Fig. 2.3.b.). Cette architecture est dénommée *Distributed Integrated Modular Avionics* (DIMA). Les modules *Concentrateurs* servent à convertir les données des capteurs au format numérique. Le principe de l’architecture DIMA est semblable à celle de l’IMA. Cependant, elle induit une proximité plus grande entre les calculateurs d’une part, et les capteurs et actionneurs d’autre part. L’autre conséquence est une diminution de la longueur et donc du poids du câblage.

Remarque 2.2

En plus des CPMs, d’autres modules réseaux sont connectés au réseau. Ce sont des modules qui servent de passerelle entre le réseau cœur AFDX et les autres protocoles réseaux. Ainsi, Input/Output Module (IOM) permet la liaison entre l’AFDX et des protocoles tels que l’ARINC 429 ou le CAN. De plus, le module GateWay Module (GWM) sert de passerelle entre l’AFDX et Ethernet.

2.2.2 Exemple de l’automobile

Cette topologie n’est pas propre à l’avionique et se retrouve dans bon nombre de systèmes temps réel distribués. Prenons, un autre exemple, avec le domaine de l’automobile comme illustré par la Fig. 2.4. Les calculateurs sont modélisés par des *Electronic Control Unit* (ECU) qui sont placés à proximité des capteurs et des actionneurs qu’ils contrôlent. Le réseau est, quant à lui, composé d’un ensemble de protocoles interconnectés entre eux (LIN, MOST, AVB, etc.).

Chaque ECU est dédiée à une ou plusieurs fonction(s) de l’automobile, avec par exemple, le

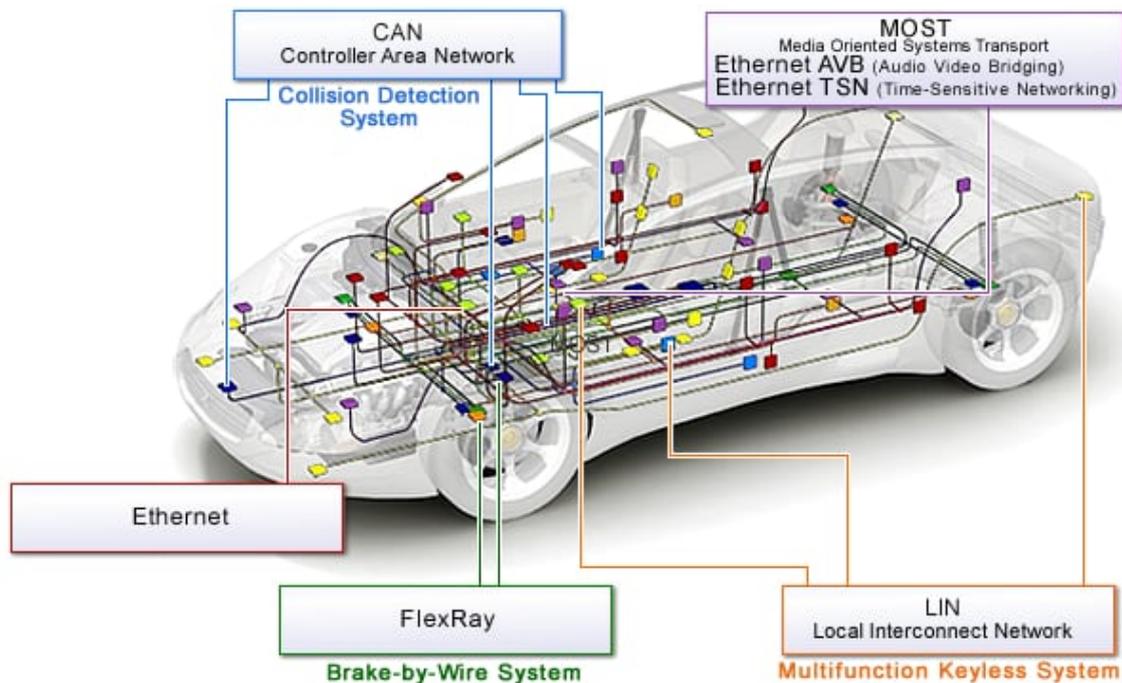


FIGURE 2.4 – Exemple d’architecture distribuée issue de l’automobile[Aut].

contrôle du châssis, l’aide à la conduite, l’ouverture des vitres ou la gestion de la radio.

2.3 Réseaux embarqués basés sur Ethernet

Les solutions choisies pour définir un réseau à hautes performances se basent généralement sur Ethernet commuté. En effet, Ethernet est un standard [Com18] décrit dès le début des années 80. Il est connu et reconnu avec une longue histoire comme une technologie fiable. Les principaux outils de développement ont déjà été développés et les composants matériels sont généralement libres d’accès. Son débit est de 10 ou 100Mbps et peut même atteindre 1 Gbps. Cependant, le principal inconvénient du protocole Ethernet pour des applications critiques est son non-déterminisme.

En effet, la méthode d’accès au support physique le *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)* [ZS14] est non déterministe par définition. En effet, un élément réseau, avant de transmettre une trame sur Ethernet écoute le support pour s’assurer que celui-ci est libre (pas d’autre transmission depuis un autre élément matériel). Une fois que le réseau est libre, il transmet sa trame. Cependant s’il détecte, durant cette transmission, qu’un autre élément réseau émet également une trame, il stoppe sa transmission, attend une durée aléatoire avant d’essayer de reprendre sa transmission. Ce temps aléatoire est par définition non déterministe.

Pour obtenir un réseau déterministe à hautes performances, plusieurs industriels ont développé des solutions basées sur de l’Ethernet commuté mais en garantissant son déterminisme. Par exemple, l’AFDX se base sur des liens *Full Duplex* qui éliminent toute collision possible sur les liens physiques

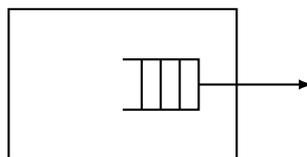


FIGURE 2.5 – End System AFDX.

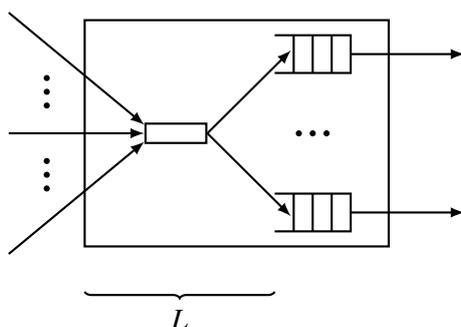


FIGURE 2.6 – Commutateur AFDX.

et ainsi toute composante aléatoire. Quant à l'AVB, il supporte un ensemble de protocoles pour assurer la synchronisation d'horloge entre les nœuds du réseau et avoir des garanties en termes de qualité de service.

2.3.1 Avionics Full Duplex switched ethernet

L'AFDX est composé d'un ensemble de *End-Systems* (ES), reliés entre eux par un ensemble de commutateurs et de liens physiques. Les ESs représentent les points d'entrée/sortie de l'AFDX. Ils sont composés d'un unique port de sortie modélisé par une file d'attente FIFO. La Fig. 2.5 dépeint la modélisation des ES. Les trames sont générées par des applications hébergées par le ES. La partie logicielle du ES, se charge de l'encapsulation du message, de son contrôle et de son placement dans la file d'attente. L'analyse d'une trame commence à son arrivée dans la file d'attente.

Les commutateurs ou *Switch* (S) sont composés d'un ensemble de ports entrants et un ensemble de ports sortants. Le modèle des commutateurs est illustré Fig. 2.6

Les ports entrants sont de type *Store and Forward*, ils n'induisent pas de délai d'attente. Le commutateur récupère les trames sur les ports entrants, il effectue des fonctions de contrôle sur la trame puis l'expédie sur le(s) port(s) de sortie en fonction de son routage stocké dans les tables de configuration statiques du commutateur. Le délai subi par une trame pour le routage dans un commutateur est une constante L , appelée aussi latence technologique, égale à $16 \mu s$ [Gri04]. Les files d'attente des ports de sortie sont de type FIFO et servent les trames à la vitesse du lien physique sortant.

Les ESs s'échangent des trames à travers des liens virtuels ou *Virtual Links* (VL). Un VL est un lien de communication virtuel avec réservation de la bande passante. Il définit un lien logique, unidirectionnel à destination d'un ou de plusieurs ESs. Chaque VL v est contraint à son entrée sur le

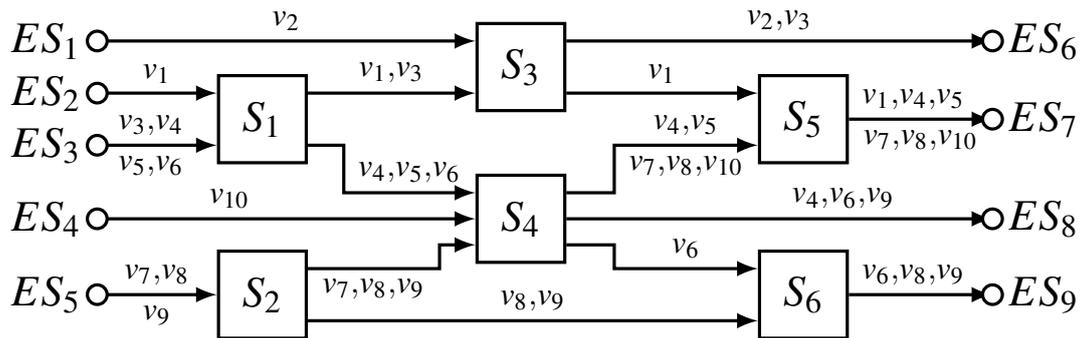


FIGURE 2.7 – Exemple de configuration AFDX.

réseau par un contrat de trafic composé de :

- la taille maximale des trames (F_{max}) qu'il peut générer, F_{max} ;
- le temps minimum entre la génération de deux trames consécutives appelé aussi *Bandwidth Allocation Gap* (BAG).

Le chemin (suite des éléments réseaux traversés par le flux entre la source et chaque destinataire) est le troisième paramètre caractérisant le flux. Il est statiquement défini. La Fig. 2.7 illustre un exemple de configuration AFDX composée de 9 ESs qui s'échangent 10 VLs au travers de 6 commutateurs.

Les communications entre les éléments réseaux (*End Systems* et commutateurs) de l'AFDX sont asynchrones. En effet, il n'y a pas d'horloge globale partagée dans l'AFDX. De plus, les liens *full-duplex* garantissent qu'aucune collision n'est possible sur les liens physiques.

Les files d'attente des éléments réseaux sont de type FIFO. Cependant, Airbus prévoit de différencier le service en fonction des flux dans l'AFDX, avec la politique de service *Fixed Priority/FIFO* (FP/FIFO).

En FP/FIFO, il y a n ($n \in \mathbb{N}^*$) niveaux de priorité (plus l'indice est petit, plus la priorité est grande). A chaque VL est attribué un niveau de priorité (le même pour l'ensemble des trames qu'il génère). Les éléments réseaux disposent, eux, d'une file d'attente par niveau de priorité. A chaque décision, l'ordonnanceur transmet les trames de la file d'attente non vide avec le plus haut niveau de priorité.

Un exemple de configuration industrielle est celle de l'A380 dépeinte Fig. 2.8 composé d'une petite centaine de ESs (représentés par des rectangle blancs) qui s'échangent des trames générées par environ 1000 VLs au travers de 8 commutateurs (rectangles gris) de 24 ports chacun. D'un point de vue de la fiabilité, le point faible de l'AFDX est le commutateur. En effet, si un commutateur tombe en panne, toutes les communications passant par ce commutateur seront perdues car il n'y a pas de reconfiguration à la volée. Afin de remplir les contraintes de la certification en termes de fiabilité et de sécurité, le réseau est dupliqué. Ainsi, il y a deux réseaux identiques qui fonctionnent en parallèle. Les commutateurs du réseau parallèle de la Fig. 2.8 sont en noirs, en superposition des commutateurs gris.

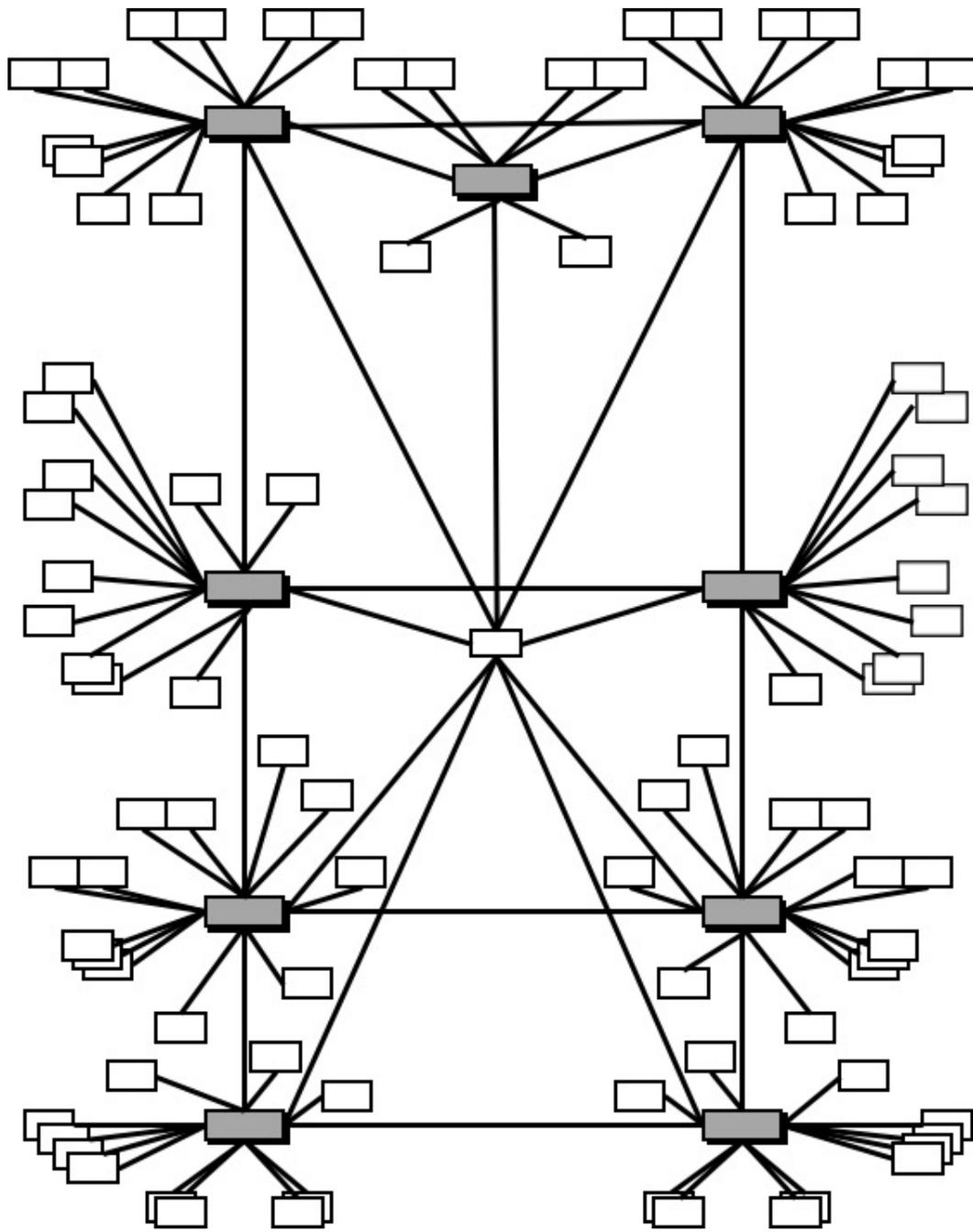


FIGURE 2.8 – Configuration industrielle AFDX de l’A380 [Bau11].

2.3.2 Audio Video Bridging Switched Ethernet

A l'origine, l'AVB faisait partie intégrante de IEEE802.3 *Residential Ethernet Study Group*. Puis un groupe de travail spécifique, IEE802.1 *Audio Video Bridging Task Group* a pris le relais. Il a été renommé en novembre 2012, IEE802.1 *Time-Sensitive Networking (TSN) Task Group* [80299], en intégrant les travaux sur l'AVB plus globalement dans le protocole *Time-Sensitive Networking* (TSN).

L'AVB est un protocole réseau, basé sur Ethernet, afin de fournir une qualité de service garantie. Ainsi, il a été développé, à ses débuts, pour transmettre des flux audio, vidéo et de contrôle ; tout en assurant une qualité suffisante aux flux audio et vidéo pour leur diffusion. Cependant les performances de l'AVB ont intéressé les industriels pour différencier le service en fonction de la criticité des flux.

Contrairement à l'afdx, les horloges des différents éléments réseau sont synchronisés, via le protocole *Timing and Synchronization for Time-Sensitive Applications* (TAS) IEEE 802.1AS [80206]. Ainsi, les communications sont synchrones.

Les flux de données sont modélisés, comme dans l'AFDX, par des liens logiques unidirectionnels avec réservation de bande passante. Chaque flux est émis sur le réseau par un module appelé *talker* et est à destination de un ou plusieurs *Listener*. Il est défini, à son entrée sur le réseau suit un chemin statiquement défini, composé de liens physiques et de commutateurs et est défini par un contrat de trafic composé par le *Class Measurement Interval* (CMI), le *Maximum Interval Frames* (MIF) et le *Maximum Frame Size* (MFS). Ainsi durant un intervalle de temps de longueur CMI, un flux génère au plus MIF trames de taille maximale MFS. La technologie des liens physiques de l'AVB (similairement à l'AFDX) assure qu'aucune collision n'est possible en AVB.

Afin d'assurer la différenciation de service, les flux sont répartis dans huit classes *Stream Reservation* (SR) de priorités. Cependant, en général, seules les deux premières (classes A et B) sont effectivement utilisées. Une classe supplémentaire, appelée *Best-Effort* (BE), est destinée aux autres flux non-prioritaires. La Fig. 2.9 illustre un exemple de configuration AVB. Cette configuration détaillée dans [Ald+12]; [Ste+12] est composée de 8 ECUs qui s'échangent 6 flux au travers de deux commutateurs.

Les différentes classes (SR et BE) sont gérées dans les éléments réseaux via la politique de service FP/FIFO. Cependant, afin de garantir une qualité de service à l'ensemble des flux, la transmission des classes de plus hautes priorités est limitée. En effet, les classes SR, en plus de FP/FIFO sont contraintes par un crédit et l'algorithme *Credit Based Shaper* (CBS) [80209].

Une extension de l'AVB, le protocole réseau *Time-Sensitive Networking* (TSN) [80299] s'affirme actuellement. Il reprend les principes fondamentaux de l'AVB (synchronisation des horloges, différenciation de service, crédit...). En plus des flux BE et des flux dits AVB modélisés par les classes SR, TSN introduit une nouvelle catégorie de flux. Ces flux, appelés *Time Triggered* (TT), sont les plus prioritaires. Afin de s'assurer que les flux TT aient une qualité de service forte avec des délais de bout-en-bout courts, ils sont transmis dans des fenêtres temporelles prédéfinies selon l'algorithme *Time Division Multiple Access* (TDMA).

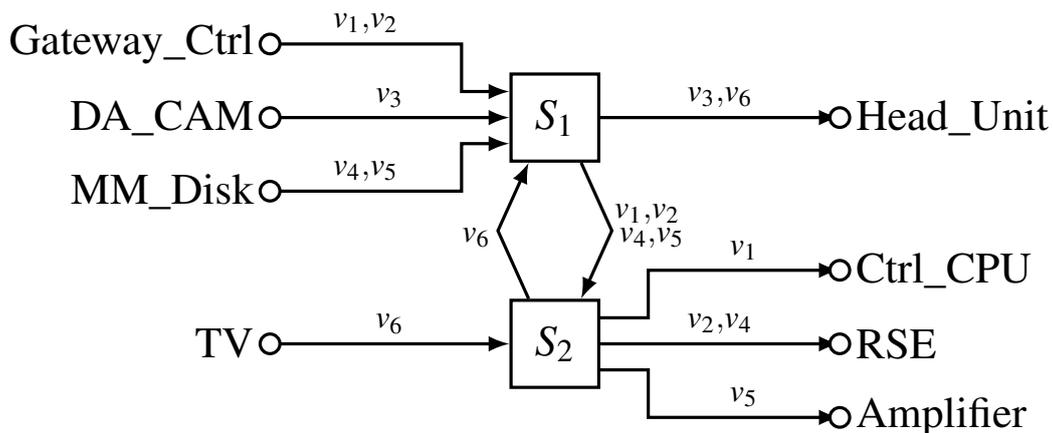


FIGURE 2.9 – Exemple de configuration AVB[Ald+12]; [Ste+12].

2.4 Propriétés des réseaux temps réel

Les communications à l'intérieur d'un réseau informatique ont été standardisées par le modèle *Open Systems Interconnection* (OSI) [74994], dépeint Fig. 2.10. Dans chaque hôte (ou système informatique), les traitements sont découpés en couches superposées. Chaque couche traite l'un des aspects de la communication et transmet les données à la couche inférieure (pour un envoi) ou supérieure (en réception). La même couche de deux systèmes communiquent en utilisant les protocoles de communication communs, propres à la couche.

Nos travaux correspondent à la couche 2 dénommée *Liaison de données* du modèle OSI pour laquelle les données sont définies sous la forme de trames.

Un réseau est dit temps réel [JPMV95]; [MZ95], s'il dispose de l'ensemble des propriétés suivantes :

- **Statique** : Toute la configuration (architecture, tables de routage, caractéristiques des flux,*etc.*) du réseau est statiquement définie au démarrage du réseau et ne peut pas évoluer en cours d'utilisation.
- **Sûreté** : le réseau doit détecter les erreurs et garantir qu'elles ne sont pas propagées dans le réseau.
- **Sécurité** : le réseau doit être résistant aux tentatives malveillantes de corruption ou récupération de données. Il doit pouvoir détecter les données corrompues et les supprimer du réseau.
- **Fiabilité** : les éléments du réseau doivent avoir une fiabilité suffisante pour que le réseau reste opérationnel. La fiabilité peut s'exprimer, par exemple, par le nombre moyen d'heures entre deux pannes consécutives).
- **Déterminisme** : Les protocoles réseaux utilisés ne doivent comporter aucune composante aléatoire.

Pour la certification des réseaux critiques, le déterminisme du réseau doit être prouvé. La démonstration du déterminisme du réseau a plusieurs implications. En particulier, sous l'hypothèse d'être dans le cas nominal (absence de panne et d'acte malveillant), nous nous focalisons à démontrer :

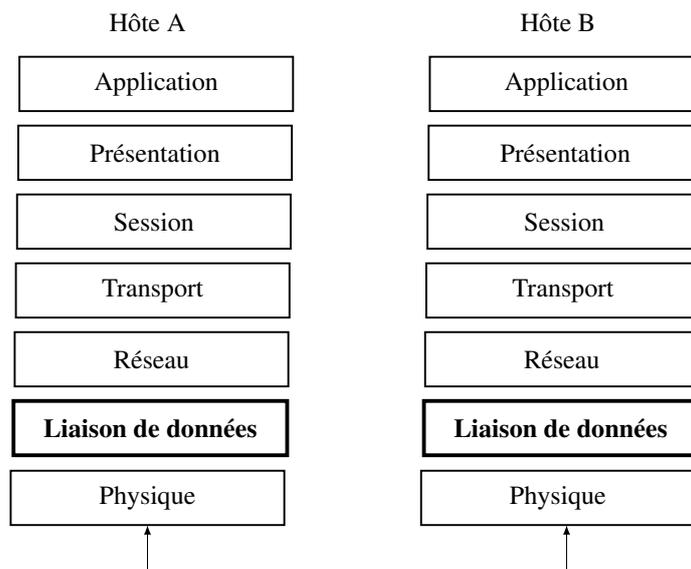


FIGURE 2.10 – Les sept couches du modèle OSI.

- qu’aucune trame ne soit perdue par le réseau ;
- que le délai de bout-en-bout de traversée du réseau pour n’importe quel flux respecte son échéance.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les systèmes temps réel embarqués. En particulier, nous nous sommes focalisés sur les réseaux embarqués basés sur le protocole Ethernet. Nous avons ainsi détaillé l’AFDX utilisé dans l’aéronautique et l’AVB dans le domaine de l’automobile. Les principales propriétés des réseaux temps-réel ont été définies.

Dans les applications embarquées critiques, les propriétés temps réel des réseaux sont importantes. Nous nous focalisons particulièrement sur la démonstration du déterminisme d’un réseau temps réel et en particulier, la non-perte de trame et l’analyse des délais de bout-en-bout des trames traversant le réseau. Dans le chapitre 3, nous présentons un état de l’art du domaine.

Analyse du déterminisme d'un réseau temps-réel

Dans le cadre de la certification, la démonstration du déterminisme de tout protocole réseau est requise pour son utilisation dans les domaines critiques tels que l'avionique. Le déterminisme est une notion complexe qui regroupe bon nombre de domaines (mécanique, informatique, réseau, électrotechnique. . .) et de contraintes à garantir. D'un point de vue réseau, au cours de ces travaux, nous nous sommes focalisés à démontrer uniquement que dans le cas nominal :

- la non-perte de trame sur le réseau ;
- le respect des échéances pour l'ensemble des flux du réseau.

La nature même des liens physiques (*p.ex. full-duplex* en AFDX) assure qu'ils ne peuvent être à l'origine d'aucune collision et donc de perte de trame, que ce soit dans les réseaux AFDX et AVB. Ainsi, seul le dépassement de la capacité d'une file d'attente dans un commutateur (ou ES) peut entraîner la perte d'une trame. Par conséquent, en plus de l'analyse du délai de bout-en-bout des flux, nous avons aussi contribué sur le calcul d'une borne supérieure garantie sur les tailles de files d'attente.

Dans le paragraphe 3.1, nous présentons la problématique liée à ces deux contraintes à garantir et notamment leur proximité en termes de calcul. Paragraphe 3.2, nous présentons un état de l'art complet sur l'analyse du délai de traversée du réseau par un flux ainsi que celle sur le calcul des tailles de files d'attente.

3.1 Problématique

Nous nous plaçons dans le cas nominal. C'est-à-dire qu'il n'y a aucune erreur (*p.ex.* aucun lien, commutateur ou calculateur défaillant), ni acte de malveillance (*p.ex.* intrusion, ajout de trame), présent dans le réseau.

Nous détaillons, dans la suite, la problématique de l'analyse du déterminisme dans les réseaux temps réel :

- Dimensionnement des files d'attente ;
- Délai de bout-en-bout de traversée du réseau par les trames.

3.1.1 Dimensionnement des files d'attente

Déterminer la taille maximum d'une file d'attente, suppose de connaître l'arriéré pire cas présent dans la file à n'importe quel instant. Cependant quand nous analysons le délai pire cas de bout-en-bout d'une trame dans un réseau, nous construisons dans chaque nœud traversé, le pire scénario. C'est-à-dire, celui maximisant son temps d'attente dans le nœud ou la quantité de travail, en nombre de bits, présente dans la file d'attente à son arrivée dans celle-ci. Ainsi, après avoir calculé le délai de traversée d'un nœud pour tous les flux passant par lui, la taille maximum de la file d'attente, en nombre de bits, peut être déduite.

Cependant, en fonction du contexte, les tailles de files d'attente ne doivent pas être dimensionnées en nombre de bits mais en nombre de trames. En effet, la taille d'une file d'attente peut être définie en nombre de bits ou en nombre de trames. Une taille d'attente définie en nombre de bits, correspond à de l'allocation dynamique de la mémoire. C'est-à-dire qu'à chaque demande d'allocation, la taille souhaitée peut varier et, à chaque fois, le nombre exact de bits demandé est attribué. Lorsque la taille d'une file d'attente est exprimée en nombre de trames, l'allocation est dite statique. C'est-à-dire que quelle que soit la taille de la demande d'allocation, un emplacement de la taille maximale (sur l'ensemble des trames) est alloué.

Dans les systèmes embarqués critiques tels que l'avionique et afin de limiter les coûts de certification, l'allocation statique est privilégiée. Cependant, dans la littérature, il n'y a pas de travaux spécifiques sur le dimensionnement des tailles de files d'attente en nombre de trames utilisables dans un contexte industriel. Néanmoins, le nombre maximum de trames présentes dans une file d'attente peut être déduit du nombre de maximum de bits, en divisant ce dernier, par la taille de trame minimale. Cette approche implique, par construction, un surdimensionnement important des files d'attente dès que les tailles de trames sont variées.

Dans la littérature, les travaux portant sur les tailles de files d'attente sont peu nombreux. Dû à leur connexité, ils sont déduits des méthodes d'analyse des délais de bout-en-bout. Ainsi, dans la suite de ce paragraphe, nous nous focalisons sur l'analyse du délai de bout-en-bout.

3.1.2 Délai de bout-en-bout

Plusieurs travaux sont menés sur l'analyse du délai de bout-en-bout d'une trame dans réseau commuté. La Fig. 3.1 permet de les visualiser graphiquement.

La distribution dépeinte Fig. 3.1 est le résultat statistique de l'analyse d'un unique flux générant des trames depuis un nœud source vers un unique destinataire. D'un point de vue théorique, cette distribution peut être construite en mesurant, sur une application réelle, le délai nécessaire à chaque trame générée par le flux depuis son entrée jusqu'à sa sortie du réseau. Cette mesure doit être relevée pendant toute la durée de vie de l'application. Ce qui d'un point de vue pratique est quasiment impossible à obtenir. Surtout que l'évaluation d'un réseau, dans les systèmes critiques, doit être menée bien avant son utilisation en configuration réelle.

La majeure partie de cette distribution peut être construite par simulation [PJM12] en amont de

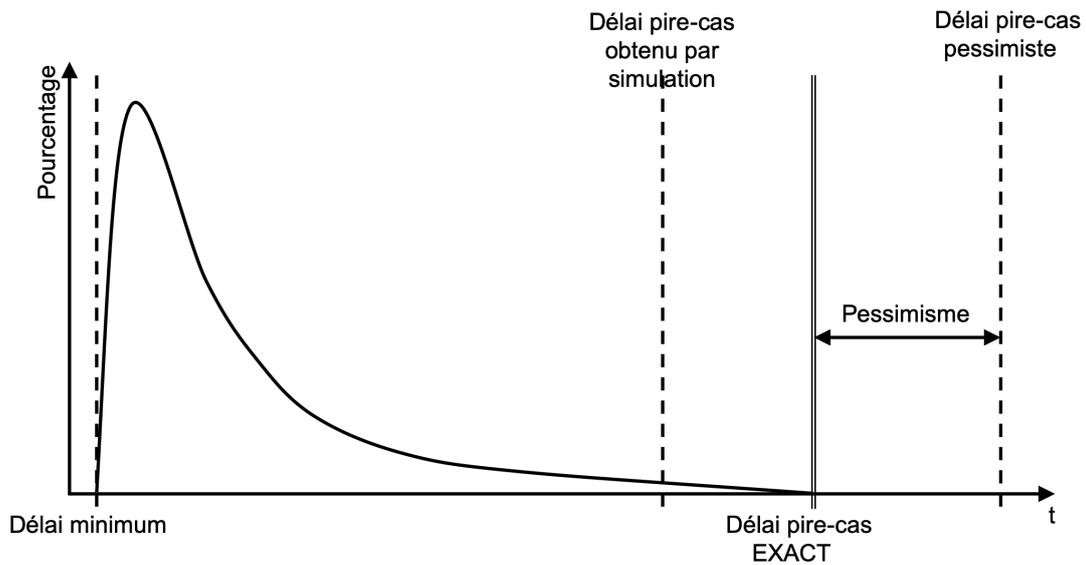


FIGURE 3.1 – Analyse du délai de bout-en-bout d'une trame.

son utilisation. C'est-à-dire que le réseau est simulé en partant des caractéristiques de la configuration complète. Cette simulation est répétée un bon nombre de fois en modifiant aléatoirement les paramètres afin d'explorer le plus de cas possibles.

De nombreuses plate-formes de simulation, commerciales ou non, existent pour modéliser les réseaux. Dans les principales, nous pouvons citer, par exemple, OMNeT++ [OMN20], NS3 [NS320] ou encore *Opnet* [Riv20].

La simulation est un bon outils pour analyser le comportement moyen d'un réseau si les paramètres d'entrée modélisent correctement le système. De nombreux travaux ont été menés dans l'AFDX [Cha+06]; [Cha07]; [Bas+10]; [Din+10]; [SS+15]; [RSS17]; [Hot+17] et l'AVB [IJH09]; [LVH11]; [Ste+11]; [Lim+12]; [Gey+13]; [Mey+13]. Cependant, comme représenté par la Fig. 3.1, le délai pire cas obtenu par simulation n'est pas garanti. Il peut se retrouver inférieur au pire cas exact, si le scénario correspondant n'a pas été simulé.

Le délai pire cas exact (Fig. 3.1 peut être déterminé par le *Model Checking* [Edm+18]. En effet, cette approche formelle se base sur modélisation à base d'automates temporisés [Alu91]; [AD94] et analyse tous les scénarios possibles. Ainsi, le scénario menant au délai pire cas est testé.

Le *Model Checking* a été appliqué à l'AFDX [Ana+06]; [Cha+06]; [Bas+10]; [Adn13] mais également dans l'AVB [IJH09]; [Kha+15]; [Ash+17]. Néanmoins, même si certains travaux tentent de limiter le nombre de scénarios à analyser [ASF11]; [Adn+12]; [Lau12], ce nombre est tel qu'il implique, généralement, une explosion combinatoire même pour des configurations très simples. Ainsi, le *Model Checking* est, en général, non utilisable dans un contexte industriel pour déterminer les délais pire cas exacts des différents flux. De plus, il a été démontré que le calcul du pire-cas exact sur le délai de transmission d'une trame de bout en bout dans un réseau de paquet est NP-difficile au sens fort [Adn+10].

Il n'existe donc pas de méthode permettant de calculer les délais pire cas exacts pour les flux d'une configuration AFDX ou AVB industrielle réelle. Des méthodes ont ainsi été définies pour calculer uniquement une borne supérieure sur le délai pire cas de chaque flux Fig. 3.1. Par construction, cette borne est garantie, dans le sens où elle ne peut jamais être dépassée. Elle peut donc être utilisée pour la validation du système. Cependant, cette borne n'étant pas le délai pire cas exact (*cf.* Fig. 3.1), elle est pessimiste et entraîne, par conséquence, un surdimensionnement du réseau. Ainsi, l'objectif majeur de ces approches est de limiter le pessimisme Fig. 3.1 qu'elles induisent, par rapport à une approche exacte.

Dans le paragraphe suivant, nous présentons un état de l'art complet sur les approches calculant une borne supérieure pire cas sur les délais de bout-en-bout et, éventuellement, sur les tailles de files d'attente.

3.2 Méthodes d'analyse d'une borne supérieure pire cas sur les délais

Le dimensionnement au plus juste dans les réseaux embarqués est un problème industriel majeur. Dans ce travail nous nous concentrons sur l'analyse des délais de bout-en-bout et le dimensionnement de files d'attente dans les réseaux Ethernet commuté et plus précisément, dans l'AFDX et l'AVB. Ces deux problèmes sont connexes et liés. Cependant, la majorité des travaux porte sur l'analyse du délai. La recherche du délai de bout-en-bout pire cas exact est un problème difficile. Ainsi, il n'existe pas d'approche permettant de déterminer le délai pire cas exact dans une configuration industrielle. Nous nous concentrons donc uniquement sur des méthodes calculant une borne supérieure de ce délai pire cas. Dans la suite, nous présentons ces approches selon un ordre qui se veut chronologique.

3.2.1 La méthode holistique

La méthode holistique a été proposée par Tindell [Tin94]; [TC94]. Elle a été conçue, initialement pour l'analyse de systèmes distribués mais avec un modèle réseau très simple. L'analyse holistique construit un pire-scénario et détermine ainsi une borne supérieure sur le délai, localement dans chaque nœud traversé par la trame. La borne supérieure du délai de bout-en-bout est obtenue en sommant les différents délais locaux. Le calcul du délai pire cas sur un nœud, se fait par un calcul de point fixe. Initialement, la méthode holistique a été conçue pour les systèmes temps réel avec ordonnancement préemptif. Elle a été étendue à l'ordonnancement non-préemptif dans [GRS96]. Dans [MM05], l'analyse holistique a été appliquée aux réseaux de paquets mais sans preuve formelle. Pour finir, Gutiérrez *et al.* [GPGH12], ont appliqué la méthode holistique pour analyser spécifiquement les *End Systems* du réseau AFDX.

3.2.2 Network Calculus

Le *Network Calculus* (NC) [Cru91a]; [Cru91b]; [BBLC18] est l'une des premières approches développées pour l'analyse de performances dans les réseaux [Le 98]; [Cha00]; [BT01] et, en particu-

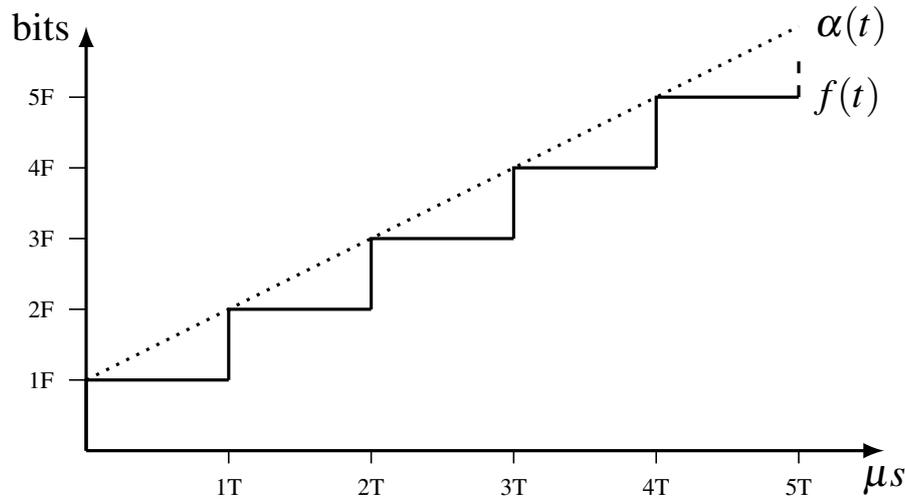


FIGURE 3.2 – Courbe d’arrivée d’un flux en NC.

lier, dans l’AFDX. Cette approche basée sur des fondamentaux mathématiques, des dioïdes [Bac+92], semblables à l’algèbre (min +) et (max, +). C’est également, à ce jour, l’outil utilisé par Airbus pour certifier ses réseaux AFDX (A380, A350...).

Outre celui développé pour Airbus, d’autres outils implémentant le NC pour analyser les réseaux ont été développés. Par exemple, nous pouvons citer *DiscoDNC* [SZ06] (en version déterministe) et *DiscoSNC* [BS13], *COINC* [Bou+09] et *RTaW-PEGASE* [Boy+10].

Le NC modélise tout élément (flux, nœud...) du réseau en courbes. Par exemple, considérons un flux avionique qui émet une trame de longueur F périodiquement tous les $T \mu s$. Ainsi, une trame de F bits arrive tous les $T \mu s$. La courbe ainsi décrite correspond au tracé de la fonction en escalier $f(t)$ de la Fig. 3.2. Cependant, le NC approxime sa courbe d’arrivée par une enveloppe maximum représentée par la droite affine $\alpha(t)$. Cette droite affine permet de simplifier les calculs. Cependant, elle induit du pessimisme car elle surapproxime la fonction $f(t)$. Le délai, ainsi calculé, définit une borne supérieure garantie par définition.

Pour analyser un nœud N traversé par un flux v , on trace sur un même graphe (cf. Fig. 3.3) :

- la courbe d’arrivée $\alpha(t)$ modélisant les trames émises par v et entrant dans N ;
- la courbe de service $\beta(t)$ qui définit le nombre maximum de bits que peut transmettre N durant un intervalle de temps t ($t \geq 0$).

L’écart horizontal maximal entre $\alpha(t)$ et $\beta(t)$, noté $h(\alpha, \beta)$, (cf. Fig. 3.3) définit le temps maximal passé par un bit dans N , par extrapolation, elle définit également le délai pire cas de traversée de N par une trame générée par v . Ce délai se calcule tant que la file d’attente est non vide ou $\beta(t) \leq \alpha(t)$. De plus, l’écart vertical entre $\beta(t)$ et $\alpha(t)$ mesuré tant que la file d’attente est non vide, $\beta(t) \leq \alpha(t)$ définit le nombre maximum de bits présents simultanément à tout instant dans la file d’attente de N . Cependant, ce nombre de bits ne permet pas de savoir le nombre de trames correspondant.

Dans l’AFDX, les premiers à appliquer le NC, avec la politique de service FIFO, dans l’AFDX

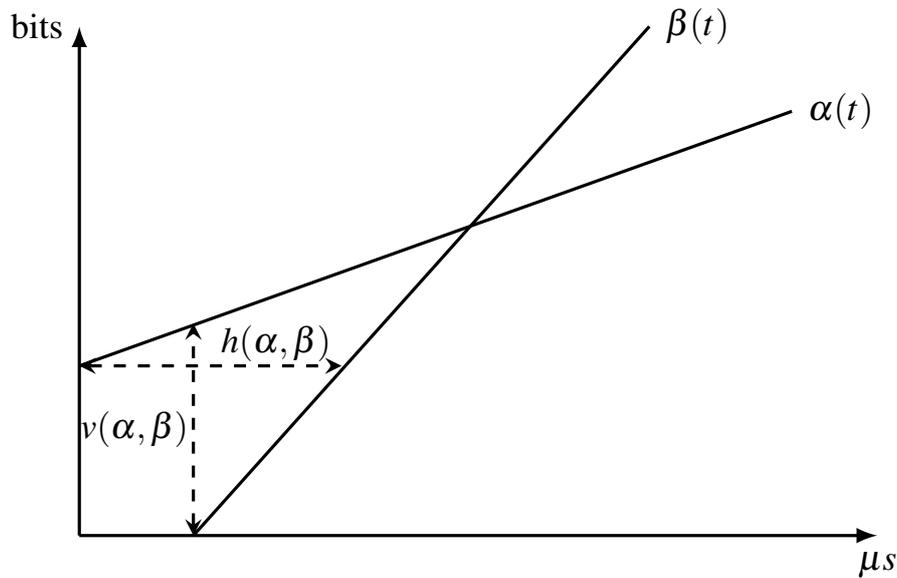


FIGURE 3.3 – Schéma général du NC.

sont Grieu *et al.* [Gri04]; [FFG06]. Leurs travaux, comme la majorité des travaux en NC ont pour objectif principal d’analyser le délai de bout-en-bout d’un flux. Ces travaux ont servi de base au développement par Airbus de son outil de certification de l’AFDX. Une comparaison entre ces travaux et la simulation ainsi que le *Model Checking* est détaillée dans [Cha+06]. Les travaux menés depuis, même encore aujourd’hui, essaient de minimiser le pessimisme induit par le NC dans l’AFDX ou tout autre réseau commuté. Ainsi, dans [BF08], les auteurs mesurent l’impact d’agréger tous les flux entrant dans un nœud. Ils généralisent cette agrégation, sur chacun des liens traversés par une trame pour obtenir le délai de bout-en-bout [Boy10]. Un travail pour une modélisation plus fine des éléments réseaux par des courbes, et ainsi pour réduire le pessimisme est détaillé dans [BMN11]. Dans [BD12], les auteurs présentent des travaux sur l’ordonnancement temps-réel afin de minimiser les courbes d’arrivée des flux. Un programme linéaire à base de NC est présenté dans [BS15a] afin de pouvoir déterminer les délais de bout-en-bout des trames.

Des travaux ont également été menés pour d’autres politiques de service comme les priorités statiques [SB11], priorités fixes [BJ11] ou même sans connaître, a priori, la politique de service [BS15b]. L’analyse de performances avec NC dans les réseaux de paquets a aussi été étudiée afin de déterminer les chemins des flux qui minimisent les délais de bout-en-bout mais aussi les tailles de file d’attente [Bou+08]. Une version probabiliste du NC a été proposée dans [RSF08a]; [RSF08b]; [SRF09] dans le cadre de l’AFDX avec la politique de service FIFO. De plus, l’outil DiscoSNC [BS13] est en version probabiliste.

Plus spécifiquement, des travaux sur le dimensionnement des tailles de files d’attente à l’aide du NC ont été menés mais ils sont peu nombreux. Ainsi, dans [Gri04], la taille des files d’attente est donnée en bits par analyse verticale des courbes (*cf.* Fig. 3.3). Cependant, dans [BFG11], les auteurs se basent sur un modèle de *Packetization* de [Cha00]; [BT01] pour évaluer la taille des files d’attente en nombre de trames. Ils modélisent ainsi deux courbes d’arrivée (une minimum et une maximum) de

paquets pour chaque flux et obtiennent des résultats bien meilleurs que ceux obtenus sur le modèle de flux classique. Ces résultats sont obtenus avec la politique de service *Round-Robin* mais ne peuvent s'étendre efficacement à FIFO ou aux priorités fixes.

Des travaux ont permis d'appliquer NC à l'AVB. Ainsi, une modélisation complète du protocole AVB et de sa politique de service CBS est définie dans [DAB14] et une première analyse des délais est présentée dans [Que12]. Les autres travaux se sont portés ensuite sur le TSN, l'extension d'AVB. Ainsi, la modélisation de TSN avec ces trois types de flux a été détaillée dans un rapport de recherche [DBZ18]. L'impact de la politique de service CBS est évalué pour les flux selon leurs priorités [HZL17]. Dans [Zha+18], les auteurs se focalisent plus spécifiquement sur les délais des flux AVB du réseau TSN. Les délais de bout-en-bout des flux du réseau TSN sont analysés [Mig+18]. Pour finir, les auteurs de [ZPC18] analysent les délais de bout-en-bout des différentes trames, en jouant sur les paramètres d'ordonnement des flux les plus prioritaires.

A notre connaissance, il n'y a pas de travaux spécifiques sur le dimensionnement des files d'attente en AVB ou TSN.

3.2.3 La méthode des Trajectoires

La méthode des trajectoires (TA) [MM06b] a été développée initialement pour l'analyse des délais dans les réseaux de paquets sous la politique d'ordonnement FIFO. Elle a ensuite été étendue à la politique de priorités fixes FP/FIFO [MM06a] puis à *Fixed Priority/Earliest Deadline First* (FP/EDF*) [MMG06]. La principale différence avec les approches précédentes, est qu'elle détermine le délai en construisant un pire scénario simultanément sur l'ensemble des nœuds traversés par la trame, depuis le dernier vers le premier.

TA a été appliquée dans le réseau AFDX [BSF09b]; [BSF09a] en FIFO, pour l'analyse des délais de bout-en-bout. Une comparaison entre les résultats obtenus avec TA et le NC sur une configuration AFDX est présentée dans [BSF10]. La méthode des Trajectoires a été appliquée pour l'analyse des délais du AFDX mais avec la politique d'ordonnement des priorités fixes [BSF12]. Une analyse du pessimisme de TA dans les réseaux Ethernet commuté est présentée dans [LSF11]. Concernant le dimensionnement des files d'attente, Coelho *et al.* déduisent à partir de l'analyse de délai pire cas de traversée d'une file d'attente, sa taille maximum en nombre de trames [CFS15]. Ce calcul n'est toutefois possible que si toutes les trames ont la même taille. La méthode des trajectoires a été également appliquée à l'analyse des délais de bout-en-bout dans l'AVB [LG17].

Cependant, malgré tous les travaux menés, la définition même de TA n'a jamais été formellement démontrée. Son utilisation pour la validation d'applications critiques est ainsi difficile. En effet, il n'est pas garanti que la borne supérieure calculée sur chaque délai de bout-en-bout soit garantie. Ainsi, dans [Kem+13], nous avons démontré que dans certains cas rares, la méthode des trajectoires pouvait déterminer une borne supérieure optimiste, c'est-à-dire inférieure au délai pire cas exact. En analysant un exemple, Li *et al.* [LCG14] ont proposé une correction de TA mais sans preuve formelle.

3.2.4 Real-Time Calculus et *Compositional Performance Analysis*

Real-Time Calculus (RTC) est basé principalement sur le NC [Cha00]; [BT01] mais utilise également des concepts d'ordonnancement temps réel [TCN00]. Il permet l'analyse de performances dans les systèmes embarqués (ordonnancement sur les processeurs et réseau). Cependant, plusieurs différences sont notables avec NC. Concernant le contexte de nos travaux, nous pouvons en retenir deux :

- Généralement, comme modélisé Fig. 3.2, en NC la courbe d'arrivée des flux est un maximum et, de même, celle de service (modélisant le nœud) est un minimum. Ce qui induit du pessimisme. En *Real-Time Calculus* les flux (*resp.* le nœud) sont modélisés par deux courbes d'arrivée (*resp.* de service), une minimum et une maximum.
- Le *Real-Time Calculus* utilise les flux d'évènements afin de générer toute arrivée ou génération d'éléments (*p.ex.* l'arrivée d'une trame).

Le RTC a été développé pour l'analyse d'ordonnancement des systèmes temps-réel critiques distribués [TCN00]; [Wan+06] mais avec une composante réseau très simplifiée. Les travaux appliquant le RTC dans les réseaux sont peu nombreux et concernent principalement le domaine de l'automobile. Ainsi dans [Rev+11], les auteurs démontrent l'application du *Real-Time Calculus* sur un exemple de réseau Ethernet/IP basé sur IEEE802.3 [Hea20] pour déterminer des délais de bout-en-bout garantis. Les auteurs de [Rei+13] utilisent les travaux issus de [Wan+06] pour définir une analyse fine des délais de bout-en-bout dans l'AVB avec deux études de cas.

L'approche *compositional performance analysis* (CPA) [DRE12]; [DTE12] basée sur le *Real-Time Calculus* a été développée afin d'analyser l'AVB puis elle a été généralisée aux réseaux Ethernet commutés. Les bornes supérieures sur les délais déterminés par CPA ont été améliorés en prenant en compte de l'effet de sérialisation bien connu dans les réseaux de paquets [Gri04] dans l'AVB [Axe+14] puis généralisés à l'Ethernet commuté [Thi+14]. Pour finir, CPA a été étendue aux priorités fixes [TAE15]. L'outil *pyCPA* [DAE12]; [Lux12], développé en Python implémente CPA.

Des travaux pour développer une version probabiliste du *Real-Time Calculus* ont été initiés dans [SCG11]. Le paradigme du *Pay-Burst-Only-Once* (PBOO) [Gri04] bien connu en NC a été intégré au *Real-Time Calculus* [Tan+19].

A notre connaissance, il n'existe pas dans la littérature des travaux sur l'application du *Real-Time Calculus* ou de la méthode CPA afin d'analyser le réseau AFDX ou afin de déterminer les tailles de files d'attente sur les réseaux Ethernet commutés.

3.2.5 Méthode de Bordoloi

Bordoloi *et al.* [Bor+14] ont développé la première approche d'analyse des délais de bout-en-bout spécifique à l'AVB et à sa politique d'ordonnancement à crédit CBS. Elle est définie pour l'analyse d'un seul nœud et s'étend à une analyse du délai sur plusieurs commutateurs en reprenant le principe de la méthode holistique. Ces travaux ont été étendus au réseau TSN dans [Ash+17].

3.2.6 La méthode des intervalles éligibles

En 2016, Cao *et al.* [Cao+16]; [Cao+16] ont développé une nouvelle approche spécifique pour l'analyse des délais de bout-en-bout dans l'AVB. Les auteurs voient en l'approche par périodes d'activités (utilisées généralement par les autres méthodes), une limitation forte due à la politique de service à crédit CBS. Les périodes d'activités ne prévoient pas, en général, les instants oisifs dus, par exemple, à un crédit négatif ne sont ainsi pas pris en compte. Ils proposent donc une approche basée sur les intervalles éligibles. Pour une classe *Stream Reservation* (SR) de priorité (*cf.* paragraphe 2.3.2), un intervalle éligible $[a, b[$ démarre par le premier instant a (avec $a \geq 0$) tel que son crédit est nul et au moins une de ses trames est en attente de transmission dans sa file d'attente. Cet intervalle se termine à l'instant b ($a \leq b$). b est le premier instant après a tel qu'il correspond à la fin de transmission d'une trame du flux et que soit le crédit de la classe devient négatif, soit la file d'attente est vide.

Basé sur ce concept, Cao *et al.* ont analysé une configuration composée d'un seul commutateur AVB. Ils analysent le délai de traversée du commutateur pour les trames du flux de classe M retardé par un flux de plus haute priorité et un flux de plus basse priorité [Cao+16]; [Cao+16]. Ils comparent leurs résultats avec ceux de CPA et de Bordoloi *et al.* et concluent à de meilleures bornes (plus proches du délai pire cas exact). Leurs travaux sont étendus au cas de plusieurs flux de plus haute et de plus basse priorité [Cao+18]. A noter que dans le cas avec plusieurs flux de plus haute priorité, une condition de validité est rajoutée. Pour finir ces travaux sont utilisés dans un commutateur TSN afin de définir la bande passante minimum nécessaire pour les flux AVB.

Motivations

Le dimensionnement au plus juste des réseaux embarqués est un besoin fort pour les industriels afin d'éviter les coûts liés au surdimensionnement. En particulier, l'analyse des délais de bout-en-bout de traversée d'un réseau est un problème difficile. L'obtention d'un délai pire cas exact pour l'ensemble des flux d'une configuration industrielle réelle est, à ce jour, impossible en un temps raisonnable. Ainsi, de nombreuses approches ont été définies pour déterminer une borne supérieure de ce délai pire cas, dans les réseaux Ethernet commutés et plus particulièrement, l'AFDX et l'AVB. Les caractéristiques des principales méthodes sont résumées dans le tableau 4.1.

	Preuve formelle	Modèle généraliste	AFDX FIFO	AFDX FP/FIFO	AVB	File d'attente bits	trames
Méthode Holistique	-	✓	-	-	-	-	-
Network Calculus (NC)	✓	✓	✓	✓	✓	✓	~
Méthode des Trajectoires (TA)	-	✓	✓	✓	✓	✓	~
<i>Computational Performance Analysis</i> (CPA)	✓	✓	-	-	✓	-	-
Travaux de Bordoloi <i>et al.</i>	✓	-	-	-	✓	-	-
La méthode des intervalles éligibles	✓	-	-	-	✓	-	-

TABLE 4.1 – Bilan des principales approches d'analyse de performance dans les réseaux AFDX et AVB.

4.1 Bilan

La méthode holistique présentée par Tindell *et al.* [Tin94] a été démontrée formellement mais son application à un réseau de paquets [MM05] ne l'a pas encore été. De plus, même si le modèle réseau est généraliste, il n'existe à ce jour, aucun exemple d'application à un protocole réel.

Le NC est la plus complète des approches et a été appliquée à plusieurs réseaux Ethernet commuté tels que l'AFDX et l'AVB. Des travaux ont également permis d'analyser les tailles de file d'attente en nombre de bits. Il existe peu de travaux dans la littérature [BFG11] pour le dimensionnement en nombre de trames. Cependant, l'outil de dimensionnement des files d'attente des commutateurs de l'AFDX, utilisé par Airbus et ses fournisseurs, est basé sur NC. A notre connaissance, il n'existe malheureusement aucune source décrivant cet outil.

La méthode des Trajectoires a fait l'objet de nombreux travaux permettant ainsi son application à

l'AFDX et l'AVB ainsi qu'à l'évaluation des tailles de file d'attente en nombre de bits mais également en nombre de trames sous la contrainte très restrictive que toutes les trames aient la même taille. Toutefois, cette approche qui est basée sur un modèle réseau de paquets généraliste ne dispose d'aucune démonstration formelle.

Les autres approches ont été définies pour l'AVB et se concentrent uniquement sur le délai. Quant à CPA, elle a été également étendue à l'Ethernet commuté. Leurs travaux respectifs ont fait l'objet de démonstrations formelles. Les travaux initiés par Bordoloi *et al.* ont été généralisés au réseau TSN. Il en est de même pour ceux portant sur les intervalles éligibles. Cependant ceux-ci se limitent à l'analyse d'un unique nœud.

4.2 Contributions

Nos travaux sur l'analyse de performances ont commencé dans l'AFDX. Les premiers travaux ont porté sur l'extension de NC en une approche probabiliste ainsi qu'une démonstration des problèmes d'optimisme de la méthode des Trajectoires. Comme le NC était l'unique méthode utilisable dans un contexte industriel, nous avons, ensuite, cherché à définir une nouvelle approche de calcul d'une borne supérieure d'un délai de bout-en-bout dans le réseau AFDX. En utilisant la culture réseau acquise dans l'équipe Ingénierie Réseaux et Télécommunications de l'IRIT et celle sur l'ordonnancement temps réel de ma formation initiale durant ma thèse dans l'équipe Temps Réel du LISI (ex-LIAS), la méthode *Forward end-to-end delay Analysis* (FA) a été définie. La suite de ce document présente les principales contributions obtenues :

- Définition de la méthode FA et de son modèle réseau pour l'évaluation des délais de bout-en-bout dans les réseaux Ethernet commutés sous les politiques d'ordonnancement FIFO et FP/FIFO ;
- Application de FA au protocole réseau AVB ;
- Dimensionnement des files d'attente en nombre de trames.
- Autre contributions

Troisième partie

Contributions à la validation temporelle des réseaux embarqués temps-réel

Forward end-to-end delay Analysis en FIFO

Sommaire

5.1	Introduction	63
5.2	Modèle réseau	63
5.3	Principe général de la méthode FA	65
5.4	Méthode FA appliquée à FIFO	66
5.4.1	Arriéré pire cas	67
5.4.2	Maximisation de l'arriéré de travail en fonction des caractéristiques des flux	67
5.4.3	Maximisation de l'arriéré de travail en fonction des caractéristiques des liens	70
5.4.4	Condition suffisante pour l'arrêt du calcul	71
5.4.5	Algorithme de FA dans un contexte FIFO	72
5.5	Expérimentations	74
5.5.1	Introduction	74
5.5.2	Transposition de l'AFDX au modèle réseau de FA	75
5.5.3	Configuration industrielle en FIFO	76
5.6	Conclusion	78

5.1 Introduction

L'approche *Forward end-to-end delay Analysis*, FA, a été développée pour déterminer des bornes supérieures garanties sur les délais de bout-en-bout dans un contexte AFDX. L'objectif était de proposer une alternative au NC qui était la seule approche applicable sur une configuration AFDX industrielle et déterminant des bornes supérieures sur les délais garantis sans induire d'optimisme. En effet, le *Model Checking* induit une explosion combinatoire même pour des configurations simples. Concernant la méthode des trajectoires, paragraphe 3.2.3, l'optimisme dans certains cas rares avait déjà été démontré. Ainsi, notre objectif avec FA était de proposer une approche alternative à NC pour le calcul des délais de bout-en-bout dans les réseaux en induisant moins de pessimisme.

La méthode FA est basée sur des résultats d'ordonnancement temps réel. En effet, l'ordonnancement temps réel est mon domaine de recherche initial et nous sommes partis de l'analogie entre la transmission d'une trame par un port de sortie d'un élément réseau avec l'ordonnancement (non-préemptif) d'une tâche par un processeur.

Le modèle réseau sur lequel se base FA (en FIFO) est décrit paragraphe 5.2. Le principe général du calcul itératif de FIFO est détaillé paragraphe 5.3 et les formules de FA avec la politique de service FIFO) sont démontrées paragraphe 5.4. Les expérimentations sont regroupées paragraphe 5.5, avant la conclusion.

5.2 Modèle réseau

La méthode FA s'applique aux réseaux commutés de paquets avec la politique de service FIFO [Kem+14]; [Ben+17b]. Nous nous plaçons sous les hypothèses suivantes :

- fonctionnement nominal : aucune panne n'est présente sur le réseau ;
- pas de problème de sécurité : aucune attaque (modification de la taille de trame, du routage...);
- pas de protocole de synchronisation : les horloges peuvent dériver et les nœuds sont indépendants.

Remarque 5.1

La synchronisation n'est pas considérée dans nos calculs. Cependant, si un réseau analysé avec FA possède un protocole de synchronisation (p.ex. AVB ou TSN), les résultats avec FA seront potentiellement pessimistes mais jamais optimistes. En effet, la synchronisation permet de limiter les interférences possibles entre les trames et donc de limiter les délais de bout-en-bout.

FA considère un réseau composé d'un ensemble de nœuds, noté S , traversé par un ensemble de flux sporadiques, noté Γ^h . L'ensemble des flux traversant un nœud $h \in S$ est dénoté Γ^h .

Les nœuds modélisent les points de multiplexage du réseau. Chaque nœud h est ainsi modélisé par n ($n \in \mathbb{N}$) liens physiques entrants (notés IP_x^h avec $1 \leq x \leq n$), une file d'attente qui sert les trames avec un débit r^h vers un unique lien de sortie. Ce lien de sortie peut être connecté à un ou plusieurs

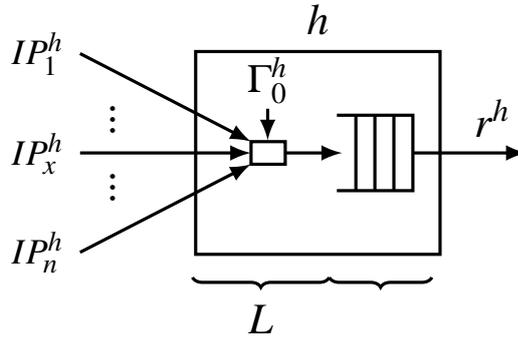


FIGURE 5.1 – Modélisation d'un nœud h .

autres nœuds. Dans le cadre de ce chapitre, la politique de service est FIFO. Le modèle de nœud est illustré Fig. 5.1.

L'ensemble des flux arrivant sur h via le lien IP_x^h est noté Γ_x^h (avec $1 \leq x \leq n$). De plus, l'ensemble des flux générés par h est dénoté Γ_0^h . Ainsi,

$$\Gamma^h = \bigcup_{x=0}^n \Gamma_x^h$$

Pour une trame générée par un flux v_i , le délai de traversée d'un nœud h se décompose en (cf. Fig. 5.1) :

- L : un délai constant incluant, la propagation électrique de la trame sur le lien, les éventuelles fonctions de contrôle (respect du contrat de trafic, routage, *etc.*);
- le temps d'attente de f_i dans la file d'attente avant la fin de sa transmission sur le lien de sortie. Ce délai peut subir de grandes variations en fonction de la quantité de données (trames) présentes dans la file à l'arrivée de f_i ;

Chaque flux $v_i \in \Gamma$ est sporadique, unidirectionnel et *unicast*. Il est généré par un unique nœud source, noté $first_i$ et suit un chemin statique, dénoté \mathcal{P}_i (constitué d'une liste ordonnée de nœuds) vers un nœud destination $last_i$.

En plus, de son chemin, un flux v_i est caractérisé par un contrat de trafic à son entrée sur le réseau :

- $Fmax_i$, la taille maximum (en bits) d'une trame générée par v_i . La durée maximale de transmission d'une trame générée par le flux v_i sur un nœud h , dénotée C_i^h , est égale à $C_i^h = \frac{Fmax_i}{r^h}$;
- T_i , le délai minimum (en μs) entre la génération de deux trames consécutives par le flux v_i sur son nœud source $first_i$.

A partir de ce modèle, nous détaillons maintenant, le principe général de FA.

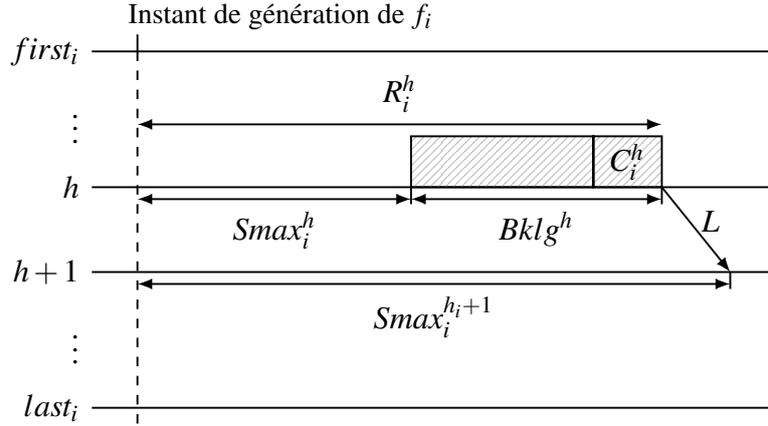


FIGURE 5.2 – Calcul itératif de R_i^h et $Smax_i^h$.

5.3 Principe général de la méthode FA

Pour chaque flux $v_i \in \Gamma$, une borne supérieure du délai de bout-en-bout de traversée du réseau est calculée. Le calcul effectué par FA est itératif et séquentiel dans l'ordre du chemin \mathcal{P}_i . Nous analysons une trame f_i (générée par v_i) et nous construisons sur chaque nœud h traversé par f_i ($h \in \mathcal{P}_i$), le pire scénario de traversée du nœud. Ce pire scénario est construit indépendamment des autres nœuds. Ainsi, le délai final obtenu est généralement pessimiste. En effet, le pire scénario construit sur un nœud n'est pas nécessairement compatible avec celui des autres nœuds.

Pour chaque nœud h , traversé par la trame f_i (du flux v_i), nous déterminons deux termes (cf. Fig. 5.2), $Smax_i^h$ et $Bklg^h$ que nous définissons maintenant.

Définition 5.1

Pour un nœud $h \in \mathcal{P}_i$, $Smax_i^h$ désigne une borne supérieure du délai de traversée du réseau depuis son instant de génération sur le nœud $first_i$ jusqu'à son arrivée sur h .

Définition 5.2

Dans un nœud h , l'arriéré de travail représente la quantité de données (en bits) présente simultanément dans la file d'attente (de h). Une définition temporelle de l'arriéré de travail peut être donnée comme le temps de transmission total des trames en attente dans la file. $Bklg^h$ définit une borne supérieure garantie de l'arriéré de travail temporel dans la file d'attente de sortie du nœud h à n'importe quel instant.

À partir de $Smax_i^h$ et $Bklg^h$ le délai pire cas dénoté R_i^h (cf. Fig. 5.2), entre la génération d'une trame f_i de v_i sur son nœud source et sa sortie de chaque nœud h est donné par :

$$R_i^h = Smax_i^h + Bklg^h \quad (5.1)$$

Le calcul de $Smax_i^h$ [Kem+14] est itératif. Soit h et $h+1$ deux nœuds consécutifs de \mathcal{P}_i . Par

construction, $Smax_i^{h+1}$ (cf. Fig. 5.2) est la somme du temps maximum nécessaire à la trame f_i de v_i pour atteindre h ($Smax_i^h$), plus l'arriéré de travail temporel pire cas subi dans la file d'attente de h , $Bklg^h$, et plus la latence technologique L :

$$\begin{cases} Smax_i^{first_i} & = 0 \\ Smax_i^{h+1} & = Smax_i^h + Bklg^h + L \end{cases} \quad (5.2)$$

Remarque 5.2

Le terme $Bklg^h$ est indépendant du flux v_i analysé. Il inclut la durée de transmission de la trame f_i du flux v_i analysé qui est toujours transmise en dernier.

Similairement à $Smax_i^h$, nous pouvons définir $Smin_i^h$:

Définition 5.3

Pour un nœud $h \in \mathcal{P}_i$, $Smin_i^h$ dénote le délai minimum de traversée du réseau depuis son instant de génération sur le nœud $first_i$ jusqu'à son arrivée sur h .

Pour tout flux v_i , de manière analogue à $Smax_i^h$, $Smin_i^h$ est calculé itérativement en considérant que la file d'attente de sortie de chaque nœud h est vide à l'arrivée de la tram f_i :

$$\begin{cases} Smin_i^{first_i} & = 0 \\ Smin_i^{h+1} & = Smin_i^h + C_i^h + L \end{cases} \quad (5.3)$$

Où $h+1$ est le nœud suivant h dans le chemin de \mathcal{P}_i .

Pour finir, le pire délai de traversée du réseau pour les trames générées par le flux $v_i \in \Gamma$, dénoté R_i^h est égal à, par définition, $R_i^{last_i}$.

Dans la suite, nous présentons le calcul du délai de bout-en-bout pire cas déterminé par la méthode FA.

5.4 Méthode FA appliquée à FIFO

Nous analysons le pire délai de la trame f_i (du flux $v_i \in S$). Pour déterminer le délai pire cas de traversée du réseau par f_i , FA détermine pour chaque nœud h traversé par S , l'arriéré temporel pire cas, $Bklg^h$ (cf. définition 5.2).

L'arriéré de travail présent dans une file d'attente, peut subir de grandes variations au fil du temps, en fonction du scénario d'arrivée des trames. Dans le paragraphe 5.4.1, nous présentons son expression. Ensuite dans les paragraphes 5.4.2 et 5.4.3, nous détaillons deux méthodes afin d'exprimer une borne supérieure de cet arriéré temporel pire cas dans un nœud h . Une condition suffisante d'arrêt des calculs est définie paragraphe 5.4.4 et pour finir paragraphe 6.3.3, l'algorithme de calculs des délais de bout-en-bout par FA est résumé.

5.4.1 Arriéré pire cas

Dans ce paragraphe, nous analysons un nœud h . Le démarrage du nœud h est fixé arbitrairement (et sans perte de généralité) à l'instant $t = 0 \mu\text{s}$ (aucune trame ne peut être générée par h ou arriver dans h avant 0).

Pour obtenir $Bklg^h$, nous analysons, à chaque instant, la différence entre les trames qui sont déjà arrivées et celles qui ont pu être transmises. Ainsi, nous définissons la fonction $W^h(t)$ (cf. définition 5.4).

Définition 5.4

Soit h un nœud. La fonction positive et cumulative $W^h(t)$ qui à chaque instant t ($t \geq 0$), associe la durée totale de transmission des trames qui sont arrivées sur h ou ont été générées par h durant l'intervalle $[0, t]$.

De plus, indépendamment du débit de lien de sortie (r^h), durant l'intervalle $[0, t]$ ($t \geq 0$), un nœud h a une durée de transmission de t . Par conséquent, à chaque instant t ($t \geq 0$), l'arriéré temporel présent dans la file d'attente du nœud h est égal à $W^h(t) - t$. Pour obtenir, $Bklg^h$ le maximum est conservé :

$$Bklg^h = \max_{t \geq 0} (W^h(t) - t) \quad (5.4)$$

Remarque 5.3

$Bklg^h$ n'est jamais négatif. En effet, à l'instant 0, par définition, $W^h(0) - 0 \geq 0$.

Pour déterminer $Bklg^h$ nous devons maximiser $W^h(t)$, dès l'instant 0. Une borne supérieure de $W^h(t)$ est déterminée selon deux approches : la première en considérant les contrats de trafic des flux (développée paragraphe 5.4.2) et l'autre en se basant sur les caractéristiques physiques du réseau et notamment le débit des liens (cf. paragraphe 5.4.3).

5.4.2 Maximisation de l'arriéré de travail en fonction des caractéristiques des flux

Pour tout flux v_i traversant h ($v_i \in \Gamma^h \setminus \Gamma_0^h$), nous présentons et démontrons, théorème 5.1, le scénario permettant de maximiser le nombre d'arrivées de trames du flux v_i , pour tout intervalle $[0, t]$ (avec $t \geq 0$).

Théorème 5.1

Dans une file d'attente FIFO, une borne supérieure de la période d'activité et donc de l'arriéré de travail pire cas généré par un flux v_i dans un nœud h appartenant à \mathcal{P}_i durant un intervalle de temps $[0, t]$ ($t \geq 0$) est obtenue lorsque :

- (i) les trames de v_i sont générées périodiquement chaque T_i unités de temps sur son nœud source ($first_i$);
- (ii) la première trame générée par le flux v_i , dénotée f_i , arrive dans h à l'instant 0;

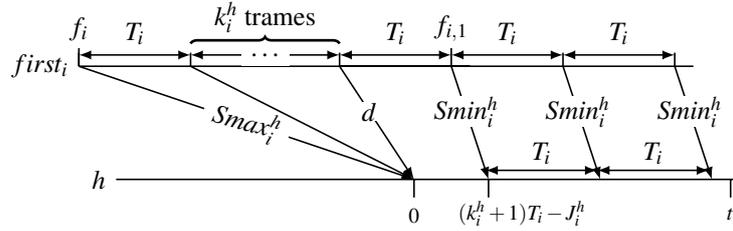


FIGURE 5.3 – Le scénario pire cas d’arrivée des trames générées par $v_i \in \Gamma^h$ dans le nœud h , maximisant l’arriéré de travail durant tout intervalle de temps $[0, t]$.

- (iii) la trame f_i atteint le nœud h en ayant subi le pire délai de traversée ($Smax_i^h$) alors que toutes les trames suivantes (générées par v_i) arrivent sur h en ayant mis un minimum de temps; mais sans jamais l’atteindre avant 0 (cf. Fig. 5.3).

Démonstration : La démonstration est détaillée dans [Ben+17b]; [Ben18] □

Le scénario décrit par le théorème 5.1 est dépeint Fig. 5.3. La première trame f_i , générée par le flux $v_i \in \Gamma^h$ arrive en h à l’instant 0 avec un délai égal à $Smax_i^h$. Pour les trames suivantes, ce délai peut varier entre $Smin_i^h$ et $Smax_i^h$. Ainsi, une ou plusieurs trames peuvent rattraper la trame f_i , générée avant elle(s) mais, selon la politique de service FIFO, sans jamais la dépasser.

Ainsi, k_i^h trames (avec $k_i^h \geq 0$) ont rattrapé f_i comme illustré dans la Fig. 5.3. Ces trames ont encouru des délais plus courts que celui de f_i pour pouvoir la rattraper. Cependant, ces délais ne peuvent pas être inférieurs à $Smin_i^h$, par définition. $f_{i,1}$ représente la première trame qui arrive dans h après les $k_i^h + 1$ trames arrivées à l’instant 0. Dans le but de maximiser l’arriéré de travail, l’écart à l’arrivée sur h avec les trames précédentes doit être minimisé. $f_{i,1}$ encourt ainsi un délai de $Smin_i^h$ pour atteindre h . De plus, puisque les trames suivantes ne peuvent arriver sur h avec un délai inférieur à $Smin_i^h$, elles arrivent toutes périodiquement dans h toutes les T_i unités de temps.

k_i^h est dépendant (cf. Fig. 5.3) de $Smin_i^h$, $Smax_i^h$ et T_i . Ainsi, nous avons démontré dans [Ben+17b] que

$$k_i^h = \left\lfloor \frac{J_i^h}{T_i} \right\rfloor \quad (5.5)$$

où $J_i^h = Smax_i^h - Smin_i^h$ est la gigue pire cas de v_i pour le nœud h .

Et plus généralement, en se basant sur l’analogie entre la transmission d’une trame par un nœud et l’exécution d’une tâche par un processeur en ordonnancement temps réel, nous pouvons définir la *Request Bound Function* (RBF) dans un réseau.

Définition 5.5

La Request Bound Function $rbf_i^h(t)$ définit le temps de transmission total, pire cas, des trames générées par le flux $v_i \in \Gamma^h$ et arrivant durant n’importe quel intervalle de temps de longueur t dans le nœud h .

Dans le théorème 5.2, nous définissons la fonction $rbf_i^h(t)$ représentant le scénario d'arrivée des trames du flux v_i selon la Fig. 5.3.

Théorème 5.2

Dans un nœud FIFO h , la Request Bound Function ($rbf_i^h(t)$) du flux $v_i \in \Gamma^h$ durant un intervalle de temps de durée t , modélisant le scénario du théorème 5.1, est :

$$rbf_i^h(t) = \left(1 + \left\lfloor \frac{t + J_i^h}{T_i} \right\rfloor\right) C_i^h \quad (5.6)$$

Où J_i^h est la gigue maximum du flux v_i dans h avec $J_i^h = Smax_i^h - Smin_i^h$.

Démonstration : La démonstration est détaillée dans [Ben+17b] ; [Ben18]. □

Remarque 5.4

Le théorème 5.2 ne différencie pas l'expression de $rbf_i^h(t)$ selon que le flux est généré directement par h ($v_i \in \Gamma_0^h$) ou que le flux arrive sur h via un lien d'entrée IP_x^h ($v_i \in \Gamma_x^h$ avec $1 \leq x \leq n$). En effet, si le flux est généré par h ($h = first_i$) alors $J_i^{first_i} = Smax_i^{first_i} - Smin_i^{first_i} = 0 - 0 = 0$, par définition, est :

$$rbf_i^{first_i}(t) = \left(1 + \left\lfloor \frac{t}{T_i} \right\rfloor\right) C_i^h$$

Ce qui correspond bien au pire cas pour le flux v_i : une trame générée périodiquement toutes T_i unités de temps, dès l'instant 0. Donc la formule (5.6) peut être étendue aux flux de Γ_0^h .

Nous en déduisons, une borne supérieure de la fonction cumulative $W^h(t)$ (cf. définition 5.4) :

$$W^h(t) = \sum_{v_i \in \Gamma^h} rbf_i^h(t) \quad (5.7)$$

La formule (5.7) peut être reformulée afin de différencier les flux arrivant sur un nœud par un lien entrant et ceux générés par le nœud lui-même. Ainsi, nous définissons la fonction $W_x^h(t)$.

Définition 5.6

Soit h un nœud. La fonction positive et cumulative $W_x^h(t)$ définit, à chaque instant t ($t \geq 0$), la durée totale de transmission des trames qui sont arrivées sur h via le lien IP_x^h avec $1 \leq x \leq n$, durant l'intervalle $[0, t]$.

Afin de différencier les flux générés par un nœud h , de ceux atteignant h via un des ses liens entrants, nous adaptons la formule de $W^h(t)$:

$$W^h(t) = \sum_{v_i \in \Gamma_0^h} rbf_i^h(t) + \sum_{x=1}^n W_x^h(t) \quad (5.8)$$

Où $W_x^h(t)$ peut être bornée par la somme des fonctions RBF des flux de Γ_x^h :

$$W_x^h(t) \leq rbf_x^h(t) = \sum_{v_i \in \Gamma_x^h} rbf_i^h(t) \quad (5.9)$$

L'approximation de $W^h(t)$, formule (5.8), ne tient pas compte des caractéristiques physiques des liens qui sont limités à un débit constant et ne permettent pas l'émission simultanée de plusieurs trames, au travers d'un même lien. En particulier, les fonctions RBF peuvent surestimer $W^h(t)$.

5.4.3 Maximisation de l'arriéré de travail en fonction des caractéristiques des liens

Dans les réseaux de paquets, les trames sont nécessairement sérialisées quand elles sont transmises à travers un lien physique à un taux r . Ainsi, des trames qui partagent un lien commun n'arrivent pas simultanément dans le nœud suivant mais séquentiellement. Leurs instants d'arrivée sont au moins séparés par leur temps de transmission à un taux maximal de r . Ce paradigme est communément appelé *effet de sérialisation* [Gri04].

Par exemple, Fig. 5.3, $k_i^h + 1$ trames arrivent simultanément sur h à l'instant 0, par le même lien physique (elles sont générées par le même flux). Ce scénario est un pire cas mais irréaliste car, au travers d'un lien physique, les arrivées de trames sont séquentielles et séparées par, au moins, C_i^h unités de temps. En tenant compte de la sérialisation et donc des caractéristiques physiques des liens, l'expression des RBF et donc de l'arriéré temporel $Bklg^h$ peut être grandement minimisée. À noter que la prise en compte de la sérialisation n'a de sens que pour les flux arrivant sur un nœud par un lien entrant et non pas, par définition, pour ceux générés par le nœud lui-même. Ainsi, la prise en compte de la sérialisation permet de calculer une autre borne supérieure de $W_x^h(t)$ de la formule (5.8), en utilisant la limitation physique du lien entrant par son débit (cf. lemme 5.1).

Lemme 5.1

Le temps de transmission cumulé par les trames entrantes via le lien d'entrée IP_x^h (avec $1 \leq x \leq n$) d'un nœud h durant un intervalle de longueur t , dénoté $W_x^h(t)$, est borné par :

$$W_x^h(t) \leq LinkRate_x^h(t) = \frac{r^{h_x}}{r^h} t + \max_{v_i \in \Gamma_x^h} C_i^h \quad (5.10)$$

Où h_x dénote le nœud prédécesseur de h via IP_x^h et r^h (resp. r^{h_x}) est le taux de transmission du nœud h (resp. h_x).

Démonstration : La démonstration est détaillée dans [Kem+15]; [Ben+17b]; [Ben18]. □

Le scénario modélisé par la fonction $LinkRate_x^h(t)$ du lemme 5.1 est dépeint Fig. 5.4. Les nœuds h_x et h sont directement connectés par un lien physique. Par conséquent, la quantité en bits, arrivant dans h via IP_x^h durant un intervalle de longueur t correspond au temps total de transmission sur le nœud h_x multiplié par son débit de sortie r^{h_x} . Il se compose :

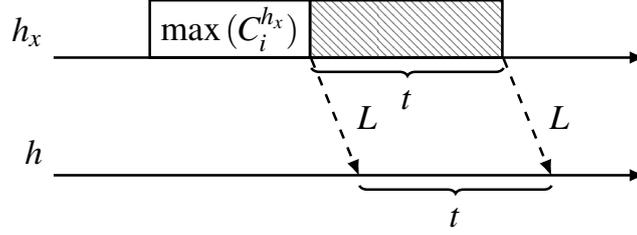


FIGURE 5.4 – La charge de travail maximale arrivant sur un nœud h via le lien d'entrée IP_x^h durant un intervalle de temps de longueur t .

- (i) d'un intervalle de longueur t transmis avec un débit de r^{h_x} , soit $r^{h_x} \times t$ (la partie hachurée de la Fig. 5.4);
- (ii) dans le pire cas, le temps de transmission de la trame de taille maximale qui arrive sur h vis IP_x^h , au tout début de l'intervalle : $r^{h_x} \times \max_{v_i \in \Gamma_x^h} C_i^{h_x}$.

Pour obtenir la formule (5.10), il reste à considérer la vitesse de transmission du nœud h , r^h .

La fonction $W_x^h(t)$ est optimisée en prenant le minimum des deux formules 5.9 et 5.10 :

$$W_x^h(t) = \min \left(\sum_{v_i \in \Gamma_x^h} r b f_i^h(t), \text{LinkRate}_x^h(t) \right) \quad (5.11)$$

Nous en déduisons, la formulation de $W^h(t)$, la fonction qui à chaque instant t ($t \geq 0$), associe la durée totale de transmission des trames qui sont arrivées sur h ou ont été générées par h durant l'intervalle $[0, t]$:

$$W^h(t) = \sum_{v_i \in \Gamma_0^h} r b f_i^h(t) + \sum_{x=1}^n \min \left(\sum_{v_i \in \Gamma_x^h} r b f_i^h(t), \text{LinkRate}_x^h(t) \right) \quad (5.12)$$

L'arriéré temporel pire cas $Bklg^h$ d'un nœud h peut ainsi être déterminé avec la formule (5.4). Cependant, le calcul de $W^h(t)$ doit être réalisé pour tout $t \geq 0$. Ainsi, une condition d'arrêt doit être déterminée.

5.4.4 Condition suffisante pour l'arrêt du calcul

Pour chaque nœud h , déterminer l'arriéré pire cas temporel, $Bklg^h$, nécessite de calculer (cf. formule (5.4)) la différence entre les fonctions $W^h(t)$ et t , pour tout $t \geq 0$. Cependant, ce calcul ne peut pas être fait indéfiniment. Notre condition d'arrêt se base sur les concepts d'instant oisif et de période d'activité définis en ordonnancement temps réel.

Définition 5.7

Sur un nœud h , un instant t est dit oisif [Leh90] si toutes les trames, générées par h ou qui sont

arrivées sur h avant t , ont toutes été complètement transmises par h à l'instant t .

Définition 5.8

Sur un nœud h , une période d'activité [Leh90] de h est définie par un intervalle de temps $[t_1, t_2)$ ($t_1 < t_2$) tel que t_1 et t_2 sont deux instants oisifs consécutifs.

Par construction, l'instant 0 est un instant oisif, à partir duquel, nous avons maximisé le nombre de trames à transmettre. Dès que pour un instant t ($t > 0$) analysé, un instant oisif est détecté, le calcul peut être stoppé. En effet, par définition, aucune trame arrivée sur un nœud avant un instant oisif ne peut impacter l'arriéré de travail après cet instant oisif. Par conséquent, le calcul est limité à la première période d'activité démarrant à l'instant 0 (instant de démarrage du nœud).

Remarque 5.5

Cette condition d'arrêt est valable quelle que soit la politique de service conservative. C'est-à-dire qui n'insère pas de temps creux si des trames sont prêtes à être transmises.

Définition 5.9

Une politique de service est conservative si elle n'insère pas de temps creux dans la transmission des trames dès lors qu'il y a des trames prêtes à être transmises.

Dans un nœud h , si le temps total de transmission des trames générées durant l'intervalle $[0, t]$ ($t > 0$), est inférieur ou égal à la quantité transmise par h durant ce même intervalle alors t est un instant oisif. En considérant les définitions des fonctions $W^h(t)$ et t , un instant oisif se produit à un instant t dans un nœud h si $W^h(t) \leq t$.

Nous définissons, maintenant, une condition suffisante (cf. théorème 5.3) garantissant l'existence d'un instant oisif $t > 0$ sur le nœud h , en se basant sur sa charge locale U^h .

Théorème 5.3

Soit un nœud FIFO h . Si les trames arrivent selon le scénario défini dans le théorème 5.1 alors l'existence d'un instant oisif sur h est garantie si la condition suivante sur sa charge locale (U^h) est vérifiée :

$$U^h = \sum_{v_i \in \Gamma^h} \frac{C_i^h}{T_i} < 1 \quad (5.13)$$

Démonstration : La démonstration est détaillée dans [Kem+15]; [Ben+17b]; [Ben18]. □

5.4.5 Algorithme de FA dans un contexte FIFO

Le délai de bout-en-bout R_i de chaque flux $v_i \in \Gamma$ est à déterminer. Pour ce calcul, tous les nœuds traversés par v_i sont analysés dans l'ordre de son chemin \mathcal{P}_i . Pour chaque nœud $h \in \mathcal{P}_i$, une borne supérieure du délai de traversée du réseau R_i^h entre sa génération sur son nœud source $first_i$ et sa sortie

du nœud h analysé est déterminé (cf. formule (5.1)). Pour finir, $R_i = R_i^{last_i}$.

L'arriéré temporel pire cas (cf. formule (5.4)) s'obtient en calculant la différence entre les fonctions $W^h(t)$ (cf. formule (5.8)) et t , jusqu'à ce que la condition d'arrêt définie paragraphe 5.4.4 soit vérifiée.

L'algorithme 1 détaille le pseudo-code du calcul par FA du délai de bout en bout pire cas d'un ensemble de flux Γ dans un réseau composé de S nœuds. Il est pseudo-polynomial en raison du calcul du point fixe.

Input: Un réseau est défini par S et Γ
 Pour chaque flux $v_i \in \Gamma$: $Fmax_i$, T_i et \mathcal{P}_i
 Pour chaque nœud $h \in S$: r^h avec $U^h < 1$ requis
 La latence technologique L

Result: R_i for each $v_i \in \Gamma$

```

1 begin
2   foreach  $v_i \in \Gamma$  do
3      $Smin_i^{first_i} \leftarrow 0$ 
4      $Smax_i^{first_i} \leftarrow 0$ 
5   foreach  $h \in S$  do
6     foreach Flux  $v_i \in \Gamma^h$  do
7        $C_i^h \leftarrow \frac{Fmax_i}{r^h}$ 
8        $J_i^h \leftarrow Smax_i^h - Smin_i^h$ 
9     foreach  $IP_x^h$  avec  $x \in \{1, \dots, n\}$  do
10       $W_x^h(t) \leftarrow \min \{rbf_x^h(t), LinkRate_x^h(t)\}$ 
11       $W^h(t) \leftarrow \sum_{v_i \in \Gamma_0^h} rbf_i^h(t) + \sum_{x=1}^n W_x^h(t)$ 
12       $Bklg^h \leftarrow 0$ 
13       $t \leftarrow 0$ 
14      do
15         $Bklg^h \leftarrow \max \{Bklg^h, W^h(t) - t\}$ 
16         $t \leftarrow \min(Set\_TestingTimes^h)$ 
17      while  $W^h(t) > t$ 
18      foreach  $v_i \in \Gamma^h$  do
19        if  $h \neq last_i$  then
20           $Smin_i^{h_i+1} \leftarrow Smin_i^h + C_i^h + L$ 
21           $Smax_i^{h_i+1} \leftarrow Smax_i^h + Bklg^h + L$ 
22        else
23           $R_i \leftarrow Smax_i^h + Bklg^h$ 

```

Algorithm 1: Calcul des délais de bout en bout pire cas, en contexte FIFO, avec la méthode FA.

L'algorithme 1 commence, par une initialisation, dans le nœud source de chaque flux v_i de la configuration, de $Smin_i^h$ et $Smax_i^h$ à zéro. Ensuite, à partir de la ligne 5, nous analysons les nœuds h , pour lesquels les caractéristiques des flux entrants ($Smin_i^h$ et $Smax_i^h$) sont calculés. Pour chacun de ses flux, nous calculons (lignes 9 – 11) respectivement de $W_x^h(t)$ et $W^h(t)$. Finalement, tant qu'un instant oisif n'a pas été détecté (ligne 17), nous calculons pour tout instant t à tester (ligne 16), la différence entre $W^h(t)$ et t et nous mettons $Bklg^h$ à jour (ligne 15).

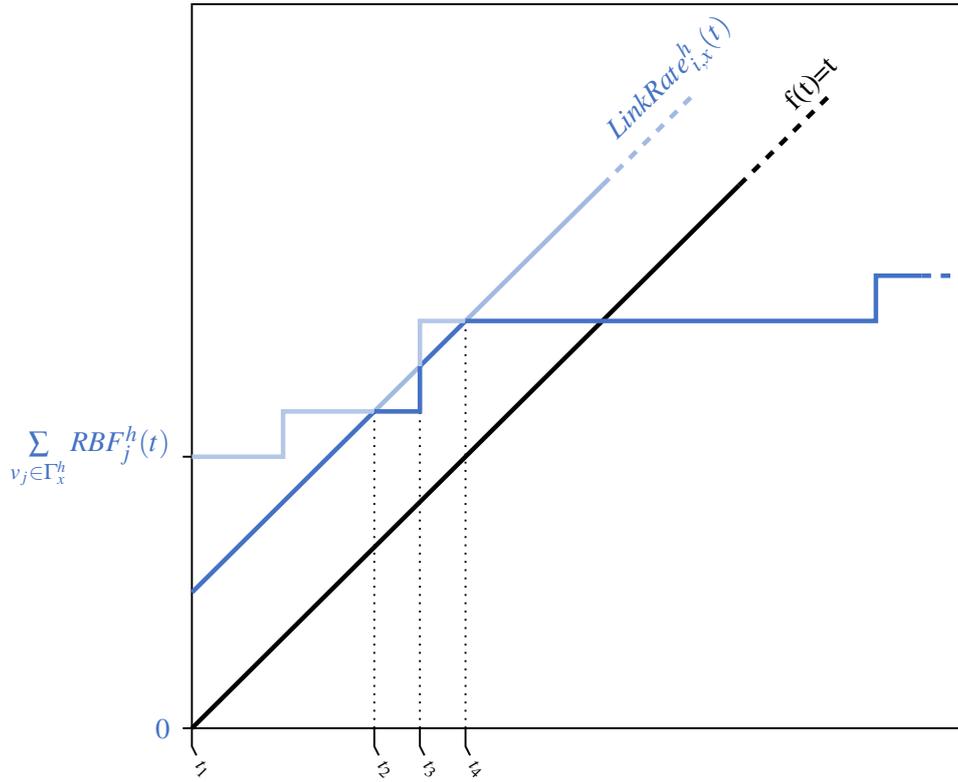


FIGURE 5.5 – Les instants t à tester.

Pour finir (lignes 20 – 21), tant que le dernier nœud n’a pas été atteint, nous calculons $Smin_i^{h_i}$ et $Smax_i^{h_i}$ dans le nœud suivant h dans le chemin \mathcal{P}_i de v_i . Sinon (ligne 23), nous calculons le délai en bout en bout de traversée du réseau R_i du flux v_i .

L’ensemble des instants t à tester, $Set_TestingTimes^h$ de la ligne 16, correspond à l’ensemble des points d’inflexion de la droite $W^h(t)$.

Pour déterminer $Set_TestingTimes^h$, prenons l’exemple de la Fig. 5.5. Soit IP_x^h le lien entrant d’un nœud h . Fig. 5.5, les représentations graphiques des fonctions $rbf_x^h(t)$ et $LinkRate_x^h(t)$ et t ont été tracées. Le tracé de $W_x^h(t)$ s’en déduit en prenant à chaque instant le minimum entre ces deux courbes. Ainsi, nous obtenons $Set_TestingTimes^h = \{t_1, t_2, t_3, t_4\}$. A noter que dès lors $W_x^h(t) < t$, le calcul s’arrête.

5.5 Expérimentations

5.5.1 Introduction

Dans ce paragraphe, nous présentons les expérimentations de la méthode FA en contexte FIFO. Nous nous plaçons dans un contexte AFDX avec la politique de service FIFO. Dans le paragraphe 3.2,

nous avons présenté l'ensemble des méthodes de calcul d'une borne supérieure du délai de bout-en-bout dont fait partie, maintenant, FA. Le tableau 4.1 résume les travaux et les domaines d'application. Ainsi en AFDX, FA peut être comparée à Network Calculus et Trajectory Approach.

Dans la suite de ce paragraphe, la version de NC est basée sur [Gri04], utilisée par Airbus pour la certification de AFDX : courbes d'arrivée de type "seau percé", courbes de services et application de l'effet de sérialisation (plutôt que le Pay-Burst-Only-Once (PBOO) paradigme [BT01]). Concernant TA, nous avons utilisé la version corrigée par Li *et al.* [LCG14] suite au problème d'optimisme.

Paragraphe 5.5.2, nous détaillons la transposition de l'AFDX à notre modèle réseau. Ensuite, paragraphe 5.5.3, nous appliquons FA sur une configuration industrielle tirée de l'A380 et nous comparons ses résultats avec ceux obtenus à l'aide du Network Calculus et de la méthode des Trajectoires.

5.5.2 Transposition de l'AFDX au modèle réseau de FA

La méthode FA a été développée initialement pour valider les délais de bout-en-bout dans l'AFDX. Ainsi, son modèle réseau en est largement inspiré. Les principales caractéristiques de l'AFDX ont été résumées dans paragraphe 2.3.1.

Les nœuds du modèle réseau de FA représentent les points de multiplexage du réseau à modéliser. Dans l'AFDX, ces points de multiplexage sont composés des ports de sortie des ES et des commutateurs. Un ES est composé d'un unique port de sortie. Dans notre modèle réseau, il est donc modélisé par un unique nœud qui génère l'ensemble des flux qui le traverse. Ainsi, il n'y a pas de lien entrant IP_x^h et donc $\Gamma^h = \Gamma_0^h$.

Les commutateurs AFDX sont composés d'un ensemble de liens entrants, des fonctions de contrôle et de routage basées sur une configuration statique et d'un ensemble de ports de sortie. Chaque port de sortie est un point de multiplexage et est modélisé par un nœud dans le modèle réseau de FA, avec comme liens entrants, ceux du commutateur. La latence technologique L de chacun de ces nœuds est fixé au temps de commutation des trames (analyse et routage) qui est égal à $L = 16 \mu s$ (cf. paragraphe 2.3.1 et [Gri04]).

Ainsi par exemple, le commutateur S_3 de la configuration illustrée Fig. 2.7 a deux ports de sortie (l'un à destination de ES_6 et l'autre à destination de S_5). S_3 est modélisé en utilisant FA par deux nœuds nommés respectivement S_{31} et S_{32} (cf. Fig. 5.6). Les ports d'entrée du commutateur sont transposés aux nœuds modélisant les ports de sortie. Les ESs ne disposent pas de port d'entrée mais d'un unique port de sortie modélisé, également, par un nœud. Le débit de chaque nœud est fixé à la vitesse du lien de sortie.

Les VLs AFDX sont définis (cf. paragraphe 2.3.1) par :

- F_{max} , la taille de trame maximale du VL qui se traduit directement en F_{max_i} , pour les flux v_i du modèle ;
- BAG, le délai minimum entre la génération de deux trames sur le nœud source qui se modélise directement par T_i , pour les flux v_i du modèle ;

Les VLs sont émis par un unique ES source et suivent un chemin statique vers un ou plusieurs

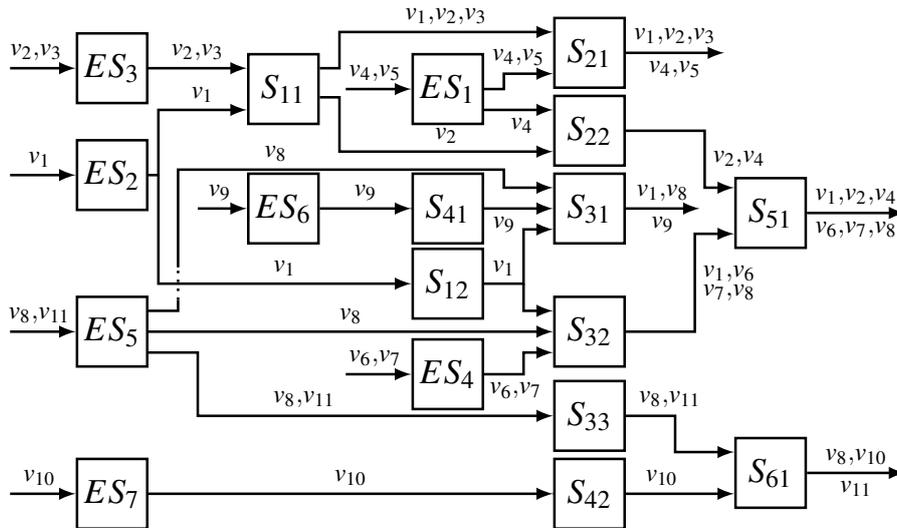


FIGURE 5.6 – Modélisation par FA de la configuration illustrée Fig. 2.7.

destinataires. Ce caractère *multicast* des VLs n'est pas compatible directement avec les flux *unicast* du modèle de FA. Ainsi, un VL est modélisé par autant de flux dans le modèle de FA qu'il n'a de destinataire. Le calcul commence sur le ES source et à chaque fois que le VL se subdivise vers différents nœuds, le calcul est dupliqué sur chaque chemin. Au final, un délai est déterminé pour chaque chemin de chaque VL.

Remarque 5.6

Dans le contexte AFDX, le calcul du délai de bout-en-bout de chaque flux s'arrête au port de sortie du dernier commutateur traversé. En effet, l'ES destination n'est pas modélisé dans le calcul car le délai qu'il induit est fortement dépendant de sa partie applicative. Ainsi, il n'est pas pris en compte dans le délai de traversée du réseau. La modélisation de la configuration AFDX illustrée Fig. 2.7 est représentée par la Fig. 5.6.

5.5.3 Configuration industrielle en FIFO

Nous appliquons les trois méthodes (FA, NC and TA) sur une configuration AFDX industrielle déployée dans l'A380 et présentée dans [Bau11]. Cette configuration est composée de 96 ESs reliés entre eux par 983 VLs *multicast* (soit 6412 flux) et 8 commutateurs de 16 ports chacun (soit 288 nœuds). Le temps de calculs des délais de bout-en-bout sur cette configuration par FA, est d'environ 2.12s avec un outil développé en Python et exécuté sur un processeur 3.3Ghz quad-core 64 bits. Par conséquent, FA est applicable sur de grandes configurations industrielles.

Nous comparons FA avec chacune des deux autres méthodes séparément. Dû au grand nombre de flux, les résultats sont présentés, sous la forme d'histogramme statistique de fréquences, dans la Fig. 5.7.

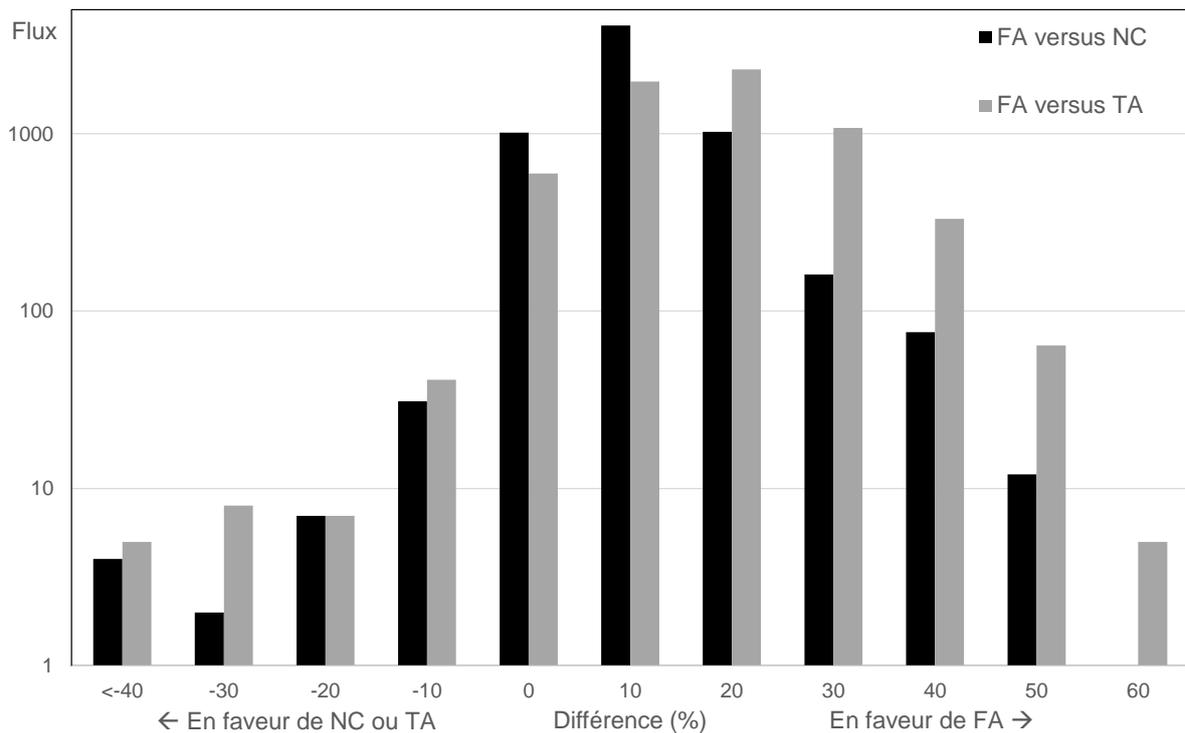


FIGURE 5.7 – Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA).

Ainsi, Pour chaque flux, nous déterminons la différence en pourcentage entre le délai de bout en bout pire cas obtenu par FA et par chacune des méthodes concurrentes (NC et TA). L’histogramme est tracé avec des intervalles de 10%. Pour chaque valeur de pourcentage, le nombre de flux correspondant est représenté verticalement par la hauteur de la barre. De plus, une valeur positive signifie que FA a obtenu une borne supérieure sur le délai inférieure à celle de l’approche concurrente, et inversement pour les colonnes négatives. Par exemple, la barre grise à 30% indique que, parmi les 6412 flux du réseau, un peu plus d’un millier de flux ont des délais pire cas calculés en utilisant FA, inférieurs entre 20 à 30% à ceux déterminés par TA.

Dans [BSF10], il a été prouvé qu’aucune des méthodes NC et TA n’est dominante. C’est-à-dire qu’aucune des deux n’obtient, pour tous les flux de toutes les configurations possibles, des bornes meilleures que l’autre approche. De même, FA n’est ni dominante, ni dominée vis-à-vis de TA et NC pour cette configuration dès lors que nous obtenons des valeurs positives et négatives dans l’histogramme de la Fig. 5.7. Ces variations de résultats entre les méthodes sont dues aux différentes hypothèses propres à chacune. Elles induisent, ainsi, du pessimisme dans les bornes calculées, mais pas nécessairement sous les mêmes contraintes, et donc pour les mêmes flux.

Cependant, il est toujours possible de combiner deux ou trois de ces approches en prenant le minimum entre les délais pire cas calculés par chacune. La borne sera toujours garantie mais réduira les effets néfastes d’un scénario défavorable à l’une d’entre elles.

Néanmoins, peu de barres dans le graphe, majoritairement basses, sont en faveur de TA ou NC. La

plupart des barres les plus hautes sont en faveur de FA. Les bornes calculées par FA sont en moyenne 4.74% moins pessimistes que celles de NC et 11.78% par rapport à celles de TA.

5.6 Conclusion

Dans ce chapitre, nous avons présenté la méthode *Forward end-to-end delay Analysis* (FA) dans un contexte FIFO. Cette approche permet de déterminer une borne supérieure garantie du délai de bout-en-bout de traversée du réseau pour chaque trame. Après avoir défini son modèle réseau en contexte FIFO, nous avons défini et démontré formellement chaque formule de FA.

Nous avons appliqué FA à une configuration industrielle issue de l'A380 d'Airbus. Nous avons comparé les délais de bout-en-bout calculés par FA avec ceux obtenus en utilisant le Network Calculus et l'approche par Trajectoires). Nous avons démontré que FA était applicable sur des configurations industrielles (temps de calculs de l'ordre de deux secondes). Nous avons démontré qu'il n'y a de domination entre FA, NC et TA. Cependant, les bornes garanties déterminées par FA sont globalement inférieures à celles de NC et TA. De plus, contrairement à TA, FA n'a pas de contrainte sur la charge globale des flux (*cf.* paragraphe 3.2.3).

Ces travaux ont donné lieu à deux conférences internationales [Kem+14]; [Kem+15] et à une revue internationale [Ben+17b]. Ils ont été réalisés principalement durant la thèse de Georges Arnaud Kemayo. Durant la thèse de Nassima Bennamar, il y a eu également un travail de réécriture du scénario (mais sans modification pour les formules). Ils ont, dans la suite été étendus à la politique de service FP/FIFO, détaillés dans le chapitre suivant.

La méthode Forward End-to-End Delay Analysis en FP/FIFO

Sommaire

6.1	Introduction	81
6.2	Modèle réseau en FP/FIFO	81
6.3	Méthode FA appliquée à FP/FIFO	83
6.3.1	Maximisation de l'arriéré de travail en fonction des caractéristiques des flux . . .	83
6.3.2	Maximisation de l'arriéré de travail en fonction des caractéristiques des liens . .	85
6.3.3	Algorithme de FA dans un contexte FP/FIFO	89
6.4	Expérimentations en FP/FIFO	89
6.5	Conclusion	92

6.1 Introduction

Nous avons défini dans le chapitre 5 une nouvelle approche pour le calcul du délai de bout-en-bout des trames traversant un réseau de paquet avec la politique de service FIFO. Cette approche FA a été démontrée formellement ainsi que son applicabilité sur des configurations industrielles. Cependant, dans les applications embarquées critiques, les industriels veulent intégrer des flux avec des contraintes temporelles différenciées. Pour reprendre l'exemple de l'avionique, un avionneur souhaite partager son réseau (*p.ex.* l'AFDX) entre des flux critiques avec des contraintes temporelles strictes (*p.ex.* les consignes issues des calculateurs de commandes de vol) et des flux sans contrainte temporelle ou avec peu de contraintes (*p.ex.* les informations cabine).

La politique de service FIFO ne permet pas de différencier le service en fonction des flux. En effet avec FIFO, le seul paramètre de priorité d'une trame est sa date d'arrivée dans le nœud. La criticité d'un flux doit être gérée statiquement afin que toutes les trames générées par un même flux aient la même priorité. La politique de service *Fixed Priority/FIFO* (FP/FIFO) permet la mise en place d'un tel mécanisme statique. Avec FP/FIFO, les flux sont répartis en classes de trafic. A chaque classe correspond un niveau de priorité. A l'intérieur d'une classe, tous les flux ont la même priorité et les trames sont transmises selon la politique FIFO. Ainsi dans un nœud FP/FIFO, si plusieurs trames sont en attente alors la trame transmise sera la première trame arrivée sur le nœud, parmi celles appartenant à la classe de plus haute priorité. Airbus a déjà prévu l'implémentation de la politique de service FP/FIFO. En effet, le réseau AFDX de l'A380 implémente deux classes de trafic : *HIGH* (*resp.* *LOW*) pour les flux plus (*resp.* moins) prioritaires. Le cas où la politique de service FP/FIFO n'est appliquée qu'avec une seule classe de trafic correspond au cas FIFO.

Dans la suite, nous présentons le modèle réseau de FA étendu à la politique de service FP/FIFO (*cf.* paragraphe 6.2). Les formules de FA sont étendues à FP/FIFO dans le paragraphe 6.3. Finalement avant de conclure, nous présentons une comparaison des résultats obtenus par FA avec ceux déterminés par deux approches (Network Calculus et la méthode des Trajectoires) qui ont déjà implémenté la politique FP/FIFO (*cf.* paragraphe 6.4).

6.2 Modèle réseau en FP/FIFO

Un réseau est toujours composé d'un ensemble de flux Γ et d'un ensemble de nœuds S . Les principales caractéristiques du réseau détaillées paragraphe 5.2 sont identiques en FP/FIFO. Nous complétons ce modèle afin de l'étendre à la politique de service FP/FIFO.

Nous considérons la politique de service FP/FIFO avec $m \in \mathbb{N}^*$ classes de trafic. Chaque classe de trafic représente un niveau de priorité $p \in \{1, \dots, m\}$ et est modélisée dans un nœud par une file FIFO. Par convention, les priorités sont rangées par ordre décroissant sur le numéro de la classe, de la plus prioritaire 1 à la moins prioritaire m .

La priorité P_i d'un flux $v_i \in \Gamma$ étant statique pour toutes ses trames, c'est une caractéristique qui s'ajoute à ses autres paramètres : $Fmax_i$, T_i et \mathcal{P}_i .

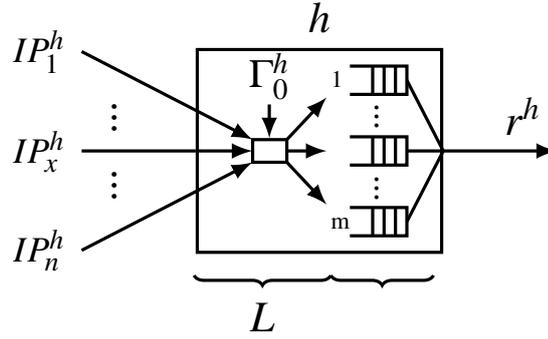


FIGURE 6.1 – Modèle nœud en FP/FIFO.

Le modèle du nœud (cf. Fig. 5.1) est modifié. En effet, l'unique file FIFO est remplacée par $m \in \mathbb{N}^*$ files FIFO de priorité. Ce modèle est résumé Fig. 6.1.

Considérons un nœud h et un flux $v_i \in \Gamma^h$. En fonction de la priorité P_i du flux v_i , les différents ensembles de flux en fonction de leurs priorités sont notés :

hp_i^h : ensemble des flux de priorité strictement supérieure : $\{v_j \in \Gamma^h | P_j < P_i\}$;

sp_i^h : ensemble des flux de même priorité : $\{v_j \in \Gamma^h | P_j = P_i\}$;

shp_i^h : ensemble des flux de priorité supérieure ou égale à P_i : $\{v_j \in \Gamma^h | P_j \leq P_i\} = sp_i^h \cup hp_i^h$;

lp_i^h : ensemble des flux de priorité strictement inférieure : $\{v_j \in \Gamma^h | P_j > P_i\}$.

De plus, si le flux v_i atteint le nœud h via le lien IP_x^h ($v_i \in \Gamma_x^h$ avec $1 \leq x \leq n$) alors les différents ensembles de flux atteignant h via IP_x^h sont notés :

$hp_{x,i}^h$: ensemble des flux de priorité strictement supérieure à P_i et atteignant le nœud h via le lien entrant IP_x^h ;

$sp_{x,i}^h$: ensemble des flux de priorité P_i et atteignant le nœud h via le lien entrant IP_x^h ;

$shp_{x,i}^h$: ensemble des flux entrants dans h via IP_x^h et de priorité supérieure ou égale à P_i :
 $\{v_j \in \Gamma_x^h | P_j \leq P_i\} = sp_{x,i}^h \cup hp_{x,i}^h$;

$lp_{x,i}^h$: ensemble des flux de priorité strictement inférieure à P_i et atteignant le nœud h via le lien entrant IP_x^h .

Par extension, $sp_{0,i}^h$ (resp. $hp_{0,i}^h$) désigne l'ensemble des flux de priorité P_i (resp. supérieures strictement à P_i) générés par le nœud h : $sp_{i,0}^h = sp_i^h \cap \Gamma_0^h$ (resp. $hp_{i,0}^h = hp_i^h \cap \Gamma_0^h$).

Dans le chapitre 5, la fonction cumulative des temps de transmission de tous les flux traversant un nœud h durant un intervalle de temps de longueur t , $W^h(t)$, est identique pour tous les flux $v_i \in \Gamma^h$. Cependant, avec la politique de service FP/FIFO, cette fonction doit être individualisée pour chaque flux $v_i \in \Gamma^h$ (et de même pour l'arrière total $Bklg^h$).

Définition 6.1

Soit h un nœud. La fonction positive et cumulative $W_i^h(t)$ qui à chaque instant t ($t \geq 0$), associe la durée totale de transmission des trames qui sont arrivées sur h ou ont été générées par h retardant la trame f_i générée par le flux $v_i \in \Gamma^h$ et arrivant sur h à l'instant t .

Définition 6.2

Soit h un nœud, l'arriéré temporel pire cas retardant la trame f_i générée $v_i \in \Gamma^h$ en traversant le nœud h est égal à

$$Bklg_i^h = \max_{t \geq 0} (W_i^h(t) - t)$$

6.3 Méthode FA appliquée à FP/FIFO

Pour un flux v_i traversant un nœud h ($v_i \in \Gamma^h$), nous allons déterminer pour chaque ensemble de flux (lp_i^h , sp_i^h et hp_i^h), l'arriéré pire cas généré pendant n'importe quel intervalle $[0, t]$ avec $t \geq 0$. Ainsi, l'expression de $W_i^h(t)$ s'écrit :

$$W_i^h(t) = WLP_i^h(t) + WSP_i^h(t) + WHP_i^h(t) \quad (6.1)$$

Où $WLP_i^h(t)$ (resp. $WSP_i^h(t)$ et $WHP_i^h(t)$) représente l'arriéré pire cas généré par les flux de lp_i^h (resp. sp_i^h et hp_i^h) généré durant l'intervalle $[0, t]$ avec $t \geq 0$.

Pour étendre les formules de FA à la politique de service FP/FIFO, nous allons procéder comme dans le cas FIFO. Tout d'abord en considérant uniquement les caractéristiques des flux (contrat de trafic) paragraphe 6.3.1, puis en considérant les caractéristiques des liens physiques (effet de sérialisation) dans le paragraphe 6.3.2.

6.3.1 Maximisation de l'arriéré de travail en fonction des caractéristiques des flux

Pour maximiser l'arriéré de travail pire cas interférant avec un flux v_i traversant un nœud h , nous adaptons les définitions 5.7 et 5.8 en tenant compte du niveau de priorité.

Définition 6.3

Pour un nœud h , un instant oisif de niveau P_i , est un instant pour lequel toutes les trames générées par les flux $v_j \in shp_i^h$ et arrivées (ou générées) avant cet instant, ont été complètement servies. Autrement dit, il n'y a pas de trame de priorité d'au moins P_i en attente dans une des files du nœud h .

Définition 6.4

Pour un nœud h , une période d'activité de niveau P_i est un intervalle de temps entre deux instants oisifs consécutifs de niveau P_i .

Pour une trame f_i générée par un flux v_i et arrivant sur un nœud h à l'instant t , FA détermine la plus longue période d'activité commençant à $0 \mu s$, instant oisif de niveau P_i par définition. En FIFO, toutes les trames doivent arriver au plus tard sur h à l'instant t pour retarder f_i . Cependant en FP/FIFO les trames plus prioritaires (générées par les flux hp_i^h) peuvent arriver après t en retardant toujours f_i .

Définition 6.5

Soit f_i une trame générée par un flux v_i et arrivant sur (ou en étant générée par) un nœud FP/FIFO h à l'instant t . Le nœud h commence la transmission de la trame f_i , dans le pire cas à l'instant $dep_i^h(t) = W_i^h(t) - C_i^h$.

Le scénario permettant de maximiser l'interférence des flux traversant un nœud h durant n'importe quel intervalle $[0, t]$ (avec $t \geq 0$) (cf. théorème 5.1) est adapté au cas FP/FIFO, en considérant notamment que t correspond à la date d'arrivée de la trame sous analyse.

Théorème 6.1

Soit f_i une trame générée par un flux v_i arrivant dans un FP/FIFO nœud h , à l'instant t ($t \geq 0$). Une borne supérieure sur la plus longue période d'activité de niveau P_i démarrant en 0 est définie si :

- (i) les trames générées par tous les flux $v_j \in \Gamma^h$ sont générées périodiquement chaque T_j dans leur nœud source ($first_j$);
- (ii) la plus longue trame générée par un des flux de priorité inférieure $\max_{v_j \in lp_i^h}(C_j^h)$, arrive sur h juste avant l'instant 0 (à l'instant $0 - \varepsilon$, avec $\varepsilon \rightarrow 0^+$);
- (iii) l'arriéré de travail généré par les flux de même priorité (sp_i^h) est maximisé durant l'intervalle de temps $[0, t]$;
- (iv) l'arriéré de travail généré par les flux de priorité supérieure (hp_i^h) est maximisé durant l'intervalle de temps $[0, dep_i^h(t)]$ [avec $dep_i^h(t) = W_i^h(t) - C_i^h$, l'instant de début de transmission de f_i par h .

Où $W_i^h(t)$ est une borne supérieure sur la charge de travail générée par tous les flux qui retardent la transmission de f_i dans h depuis l'instant 0.

Démonstration : Le scénario pire cas de génération des trames de n'importe quel flux reste identique à FIFO. Pour les flux de lp_i^h , une seule trame peut retarder f_i . De plus en FP/FIFO, toutes les trames générées par les flux de même priorité (resp. de priorité supérieure) retardent la trame f_i si elles arrivent avant elle sur le nœud h , t (resp. son départ du nœud h , $dep_i^h(t) = W_i^h(t) - C_i^h$). Ce théorème a été démontré dans [Ben+17b]; [Ben18]. □

Ce pire scénario se traduit en interférences spécifiques selon le niveau de priorité des flux.

Théorème 6.2

La plus longue période d'activité rencontrée par une trame f_i générée par un flux v_i , arrivant dans un FP/FIFO nœud h à l'instant t , est bornée par le plus petit point fixe calculé par la formule (6.1) avec :

$$\begin{cases} W_{t,i}^h(0) = C_i^h \\ W_{t,i}^h(k+1) = WLP_i^h + WSP_i^h(t) + WHP_{t,i}^h(k) \end{cases} \quad (6.2)$$

Où

$$\begin{aligned}
WLP_i^h(t) &= \max_{v_j \in IP_i^h} C_j^h \\
WSP_i^h(t) &= \sum_{v_j \in SP_i^h} rbf_j^h(t) \\
WHP_{t,i}^h(k) &= \sum_{v_j \in HP_i^h} rbf_j^h \left(W_{t,i}^h(k) - C_i^h \right).
\end{aligned}$$

La calcul est effectué itérativement jusqu'au premier point fixe, $W_{t,i}^h(k+1) = W_{t,i}^h(k)$ et $W_i^h(t)$ dénote, alors, ce plus petit point fixe.

Démonstration : Le calcul itératif du point fixe est nécessaire. En effet, plus il y a des trames de flux plus prioritaires qui arrivent, plus l'instant de départ $dep_i^h(t)$ est retardé. Ce théorème a été démontré dans [Ben+17b]; [Ben18]. \square

Remarque 6.1

Le calcul de point fixe du théorème 6.2 converge si la condition sur la charge du nœud h , $\sum_{v_j \in \Gamma^h} \frac{C_j}{T_j} < 1$, est respectée.

Le théorème 6.2 permet d'assurer une borne supérieure garantie du délai de bout-en-bout pour les trames de chaque flux $v_i \in \Gamma$. Cependant, cette borne est pessimiste (au moins pour des petites valeurs de t). Dans le paragraphe 6.3.2, nous fournissons une autre borne garantie basée sur le débit des liens physiques et la prise en compte de la sérialisation.

6.3.2 Maximisation de l'arriéré de travail en fonction des caractéristiques des liens

Cette borne supérieure n'a, comme pour le cas FIFO, de sens que pour les flux qui ne sont générés par le nœud h analysé. Ainsi, dans la suite, nous nous intéresserons à borner les interférences dues aux flux entrants via le lien d'entrée IP_x^h (avec $1 \leq x \leq n$).

En FIFO, FA se base sur les fonctions RBF afin de déterminer l'interférence de chaque flux. Cependant, une RBF associée à chaque t ($t \geq 0$) le nombre de trames pire cas arrivées (générées) sur un nœud h . En particulier, il n'y a aucune notion sur les instants d'arrivée de ces trames. Ainsi, pour un même nombre de trames déterminé par la fonction $rbf_i^h(t)$ (cf. théorème 5.2), il est possible de définir plusieurs scénarios d'arrivée de trames, comme illustrés Fig. 6.2. Ils correspondent, pour n'importe quel intervalle $[0, t]$ (avec $t \geq 0$) à la même quantité de trames mais avec une rafale de trames arrivant à l'instant 0 dans la Fig. 6.2.a et à l'instant t dans la Fig. 6.2.b. En effet, si une trame f_i arrive dans un nœud h FIFO à l'instant t , peu importe l'instant d'arrivée de la rafale de trames durant l'intervalle $[0, t]$, toute trame arrivant avant t sera, dans le pire cas, transmise avant f_i . Le scénario de la Fig. 6.2.a été formellement démontré théorème 5.1. Celui de la Fig. 6.2.b a, lui, été démontré dans [Kem+14].

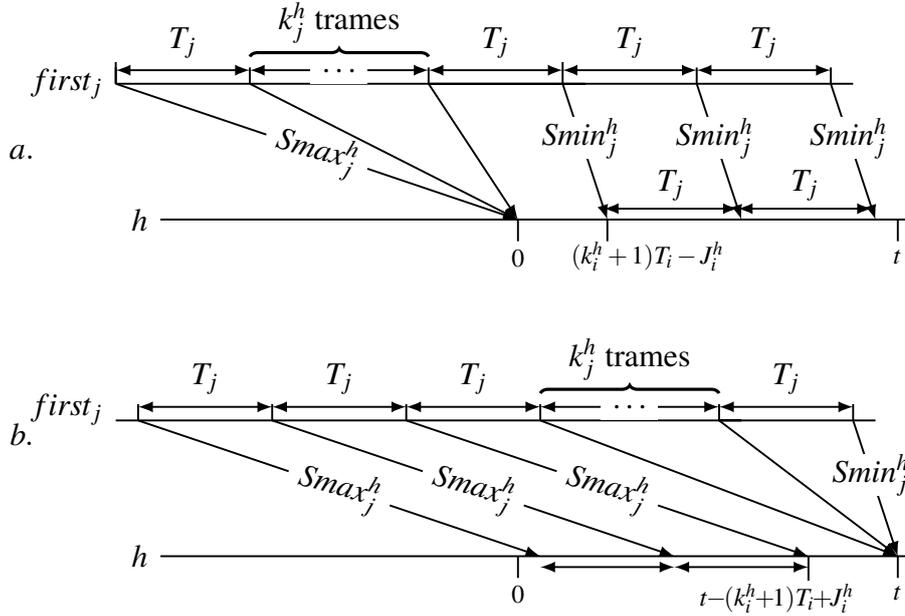


FIGURE 6.2 – Les scénarios d’arrivée pire cas des trames (tel que $v_j \in \Gamma^h$) où la rafale de $(1 + k_j^h)$ trames arrive à l’instant 0 (a) ou à l’instant t (b).

Néanmoins, avec la politique FP/FIFO et en fonction de la priorité des flux, les dates d’arrivées des trames et l’instant d’arrivée de la rafale de trames a une incidence forte sur la construction du pire cas. En considérant les caractéristiques physiques et notamment l’arrivée séquentielle par un lien physique (effet de sérialisation), nous définissons un scénario pire cas :

Théorème 6.3

Pour une trame f_i générée par un flux v_i arrivant dans un nœud FP/FIFO h à l’instant t ($t \geq 0$), la plus longue période d’activité de niveau P_i , démarrant en 0, est définie par :

- (i) les trames générées par tous les flux $v_j \in \Gamma^h$ sont générées périodiquement chaque T_j dans leur nœud source ($first_j$);
- (ii) l’arrivée (ou génération) de la trame, de priorité inférieure lp_i^h , avec le plus long temps de transmission $\max_{v_j \in lp_i^h}(C_j^h)$, se produit juste avant l’instant 0 (à l’instant $0 - \varepsilon$, avec $\varepsilon \rightarrow 0^+$);
- (iii) l’arriéré de travail généré par les flux de même priorité (sp_i^h) est maximisé durant l’intervalle de temps $[0, t]$;
- (iv) l’arriéré de travail engendré par les flux de priorité supérieure et générés par h ($hp_{0,i}^h$) est maximisé durant l’intervalle de temps $[0, dep_i^h(t)[$ avec $dep_i^h(t) = W_i^h(t) - C_i^h$;
- (v) l’arriéré de travail généré par les flux de priorité supérieure et arrivant par les liens IP_x^h ($hp_{x,i}^h$) avec $1 \leq x \leq n$ est maximisé durant l’intervalle de temps $[0, dep_i^h(t)[$ avec $dep_i^h(t) = W_i^h(t) - C_i^h$ mais, pour autant de trames que possible, les arrivées sont repoussées dans le sous-intervalle $[t, dep_i^h(t)[$.

Démonstration : Les points (i), (ii) et (iii) ont déjà été démontrés dans le théorème 6.1. Pour les

flux de plus haute priorité, hp_i^h , il faut différencier les flux générés par h et ceux qui arrivent par h via un lien entrant. En effet, toute trame de priorité strictement supérieure retarde la trame f_i étudiée du moment qu'elle arrive avant que celle-ci commence sa transmission en $dep_i^h(t) = W_i^h(t) - C_i^h$.

Les trames de même priorité ne peuvent interférer avec f_i que si elles arrivent avant elle (instant t) sur h (dû à la gestion FIFO entre flux de sp_i^h). Cependant, dû à la sérialisation et l'arrivée séquentielle des trames via un même lien, toute trame de priorité strictement supérieure qui atteint h via un lien IP_x^h avec $1 \leq x \leq n$ durant l'intervalle $[0, t]$ interfère avec les trames de $sp_{x,i}^h$ et pourraient retarder leur arrivée sur h après t et donc limiter leur impact sur f_i .

Ainsi, pour maximiser les trames atteignant h via IP_x^h durant l'intervalle $[0, t]$, les trames de $hp_{x,i}^h$ sont bien maximisées sur $[0, dep_i^h(t)]$ [avec $dep_i^h(t) = W_i^h(t) - C_i^h$], mais retardées pour autant de trames que possible après t . Ce dernier point a été prouvé dans [Ben+17b]; [Ben18].

□

Afin de pouvoir déterminer la quantité de trames de $sp_{x,i}^h$ entrant par le lien physique IP_x^h (avec $1 \leq x \leq n$), nous avons besoin de connaître le plus petit nombre de trames générées par chaque flux $v_j \in hp_{x,i}^h$ qui peuvent arriver durant l'intervalle $[0, t]$ tout en maximisant ce nombre sur $[0, dep_i^h(t)]$.

Lemme 6.1

Soit $v_j \in \Gamma^h$, dont l'arriéré de travail est maximisé selon la fonction rbf_j^h . Si la rafale de trames générée par v_j arrive à l'instant t (avec $t \geq 0$) alors le nombre de trames générées dans l'intervalle $[0, t]$, dénoté $BHP_i^h(t)$ est égal à :

$$rbf_j^h(t) - rbf_j^h(0) = \left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor\right) C_j^h$$

Démonstration : Le résultat correspond à un régime strictement périodique des trames, dès l'instant 0. Ce lemme a été démontré dans [Ben+17b]; [Ben18].

□

Nous modifions, maintenant, le calcul de point fixe du théorème 6.2 pour introduire la sérialisation au niveau des trames appartenant à $sp_{x,i}^h$ et $hp_{x,i}^h$ pour chaque lien entrant IP_x^h (avec $1 \leq x \leq n$). Ainsi, les termes $WSP_i^h(t)$ et $WHP_i^h(t)$ peuvent se réécrire respectivement :

$$WSP_i^h(t) = \sum_{v_j \in sp_{0,i}^h} rbf_j^h(t) + \sum_{x=1}^n \sum_{v_j \in sp_{x,i}^h} rbf_j^h(t) \quad (6.3)$$

$$WHP_{t,i}^h(k) = \sum_{v_j \in hp_{0,i}^h} rbf_j^h(W_{t,i}^h(k) - C_i^h) + \sum_{x=1}^n \sum_{v_j \in hp_{x,i}^h} rbf_j^h(W_{t,i}^h(k) - C_i^h) \quad (6.4)$$

Théorème 6.4

L'arriéré de travail pire cas retardant une trame f_i générée par le flux v_i , arrivée dans un FP/FIFO nœud h à l'instant t , est borné par le plus petit point fixe déterminé par la formule (6.5) si la condition sur la charge $\sum_{v_j \in \Gamma^h} \frac{C_j}{T_j} < 1$ est respectée.

$$\begin{cases} W_{t,i}^h(0) = C_i^h \\ W_{t,i}^h(k+1) = WLP_i^h + WSP_i^h(t) + WHP_{t,i}^h(k) \end{cases} \quad (6.5)$$

Où

$$\begin{aligned} WLP_i^h(t) &= \max_{v_j \in lp_i^h} C_j^h \\ WSP_i^h(t) &= \sum_{v_j \in sp_{i,0}^h} rbf_j^h(t) + \sum_{x=1}^n \min \left(\sum_{v_j \in sp_{x,i}^h} rbf_j^h(t), LinkRate_{x,i}^h(t) \right) \\ WHP_{t,i}^h(k) &= \sum_{v_j \in hp_i^h} rbf_j^h \left(W_{t,i}^h(k) - C_i^h \right). \end{aligned}$$

avec

$$LinkRate_{x,i}^h(t) = \frac{r^{hx}}{r^h} t + \max_{v_j \in shp_{x,i}^h} C_j^h - \sum_{v_j \in hp_{x,i}^h} \left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor \right) C_j^h$$

La calcul est effectué itérativement jusqu'au premier point fixe, $W_{t,i}^h(k+1) = W_{t,i}^h(k)$ et $W_i^h(t)$ dénote alors ce plus petit point fixe.

Démonstration :

Les flux de $hp_{x,i}^h$ ne sont pas concernés par la sérialisation. En effet, tant qu'il y a des trames de plus haute priorité dans un nœud précédent h , quelle que soit la vitesse du lien, dans le pire cas toutes ces trames auront le temps d'atteindre h avant le départ de la trame sous analyse. La fonction $LinkRate_x^h(t)$ est semblable à celle définie lemme 5.1. Cependant, nous devons soustraire le nombre de trames des flux de $hp_{x,i}^h$ arrivées durant l'intervalle $[0, t]$ (cf. lemme 6.1).

Le signe négatif dans l'expression de $LinkRate_x^h(t)$ n'entraîne pas de problème d'optimisme. En effet, la contribution des flux de priorité P_i et de priorités supérieures doit être étudiée séparément dans les intervalles $[0, t[$ et $[t, dep_i^h(t)[$ (avec $dep_i^h(t) = W_i^h(t) - C_i^h$). Selon le théorème 6.4 et le lemme 6.1, l'interférence des flux plus prioritaires peut être reformulées sous la forme suivante :

$$\sum_{v_j \in hp_i^h} \left(\underbrace{\left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor \right) C_j^h}_{[0,t[} + \underbrace{rbf_j^h \left(W_i^h(t) - C_i^h \right) - \left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor \right) C_j^h}_{[t, dep_i^h(t)[} \right) = \sum_{v_j \in hp_i^h} rbf_j^h \left(W_i^h(t) - C_i^h \right)$$

Quand à la sérialisation des flux sur le lien entrant IP_x^h (avec $1 \leq x \leq n$), la fonction $LinkRate_x^h(t)$

est semblable à celle définie lemme 5.1. Cependant, cette fonction $LinkRate_x^h(t)$ limite la quantité de données générées par les flux de priorité au moins égale à P_i durant l'intervalle $[0, t]$ (avec $t \geq 0$) :

$$\sum_{v_j \in sp_{x,i}^h} rb f_j^h(t) + \sum_{v_j \in hp_{x,i}^h} \left(1 + \left\lfloor \frac{t}{T_j} \right\rfloor\right) C_j^h$$

L'équation de $LinkRate_x^h(t)$ est alors adaptée aux seuls flux de P_i pour lesquels la sérialisation a un sens. Nous avons démontré ce théorème dans [Ben+17b]; [Ben18]. \square

Corollaire 6.1

Il est à noter que dans le cas particulier où nous avons une unique classe de priorité ($m = 1$) alors la simplification des formules du théorème 6.4 coïncide avec la formule (5.12) du cas FIFO.

La condition suffisante pour l'arrêt des calculs (cf. paragraphe 5.4.4) présentée en FIFO, reste valable en FP/FIFO.

6.3.3 Algorithme de FA dans un contexte FP/FIFO

Le calcul du délai de bout en bout en FP/FIFO avec FA est semblable à celui défini pour la politique d'ordonnancement FIFO. Cependant, l'ensemble des points à tester dépend du niveau de priorité du flux analysé, en raison du terme constant de $LinkRate_{i,x}^h$. De plus, pour chaque instant à tester t , le calcul de point fixe doit être effectué :

6.4 Expérimentations en FP/FIFO

Nous appliquons FA sur la configuration AFDX industrielle, présentée paragraphe 5.5.3 pour le cas FIFO. Cependant, nous avons modifié cette configuration afin d'introduire, de manière aléatoire sur les flux, 6 niveaux de priorité. Sur cette configuration, nous comparons les résultats de FA avec ceux obtenus avec TA et NC.

Le temps d'exécution de notre approche FA sur cette configuration est d'environ 7s avec un outil développé en Python et exécuté sur un processeur 3.3Ghz quad-core 64 bits. Par conséquent, FA est applicable sur des configurations industrielles sous la politique d'ordonnancement FP/FIFO.

Le comparatif entre FA et les approches TA et NC a fait l'objet d'une étude statistique sur les délais (semblable à celle présentée paragraphe 5.5.3) sous la forme d'un histogramme de fréquence (cf. Fig. 6.3).

Comme pour l'analyse de la configuration industrielle détaillée dans le paragraphe 5.5.3, l'histogramme est construit par intervalle de 10%. Chaque barre verticale noire (resp. grise) correspond au nombre de flux dont la différence entre les délais déterminés par FA et NC (resp. TA) est en

Input: Un réseau est défini par S et Γ
 Pour chaque flux $v_i \in \Gamma$: $Fmax_i$, T_i et \mathcal{P}_i
 Pour chaque nœud $h \in S$: r^h avec $U^h < 1$ requis
 La latence technologique L

Result: R_i for each $v_i \in \Gamma$

```

1 begin
2   foreach  $v_i \in \Gamma$  do
3      $Smin_i^{first_i} \leftarrow 0$ 
4      $Smax_i^{first_i} \leftarrow 0$ 
5   foreach  $h \in S$  do
6     foreach Flux  $v_i \in \Gamma^h$  do
7        $C_i^h \leftarrow \frac{Fmax_i}{r^h}$ 
8        $J_i^h \leftarrow Smax_i^h - Smin_i^h$ 
9        $Bklg_i^h \leftarrow 0$ 
10       $t \leftarrow 0$ 
11      do
12         $W_{t,i}^h(0) \leftarrow C_i^h$ 
13         $k \leftarrow 1$ 
14        do
15           $W_{t,i}^h(k+1) \leftarrow WLP_i^h + WSP_i^h(t) + WHP_{t,i}^h(k)$ 
16           $k \leftarrow k+1$ 
17        while  $W_{t,i}^h(k) \neq W_{t,i}^h(k+1)$ 
18           $W_i^h(t) \leftarrow W_{t,i}^h(k+1)$ 
19           $Bklg_i^h \leftarrow \max \{ Bklg_i^h, W_i^h(t) - t \}$ 
20           $t \leftarrow \min( Set\_TestingTimes_i^h )$ 
21        while  $W_i^h(t) > t$ 
22        if  $h \neq last_i$  then
23           $Smin_i^{h_i+1} \leftarrow Smin_i^h + C_i^h + L$ 
24           $Smax_i^{h_i+1} \leftarrow Smax_i^h + Bklg_i^h + L$ 
25        else
26           $R_i \leftarrow Smax_i^h + Bklg_i^h$ 

```

Algorithm 2: Calcul des délais de bout en bout pire cas, en contexte FP/FIFO, avec la méthode FA.

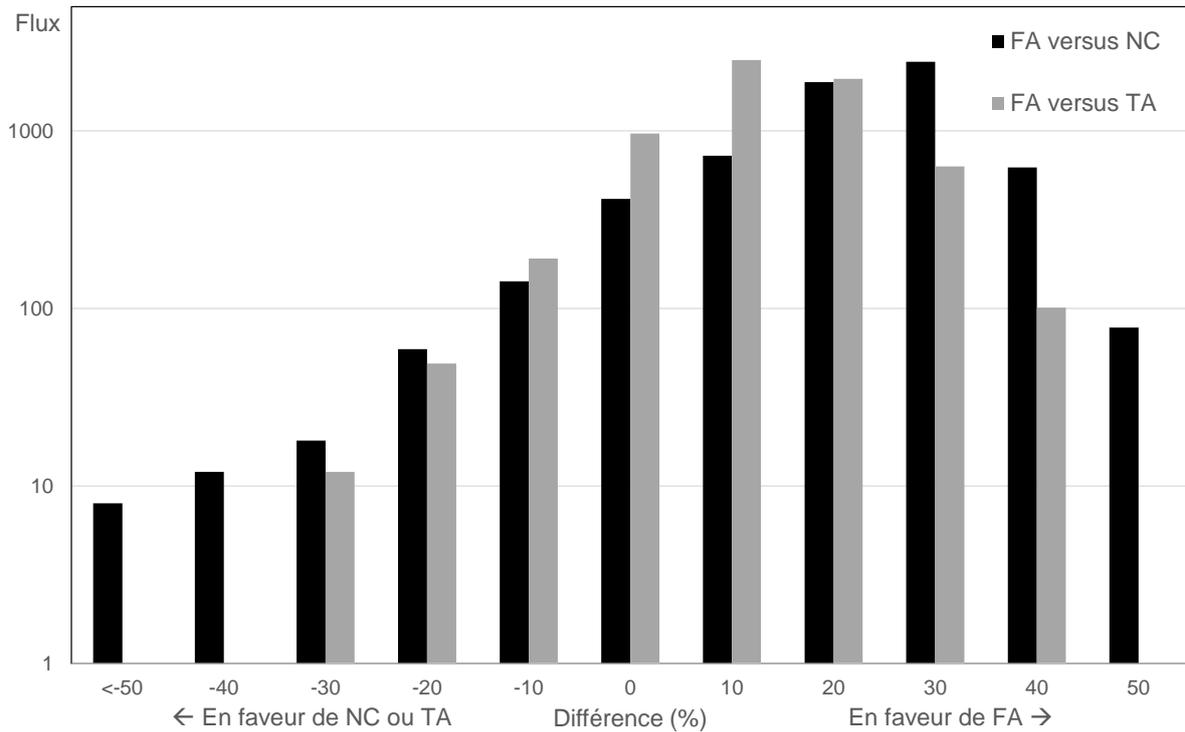


FIGURE 6.3 – Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA) en FP/FIFO.

pourcentage donnée par la valeur inscrite sur l'axe des abscisses. Une barre positive correspond à un délai inférieur déterminé par FA par rapport à l'autre approche et inversement pour une barre négative. Ainsi, la barre grise correspondant à 40% signifie que pour environ 100 flux, le délai calculé en utilisant FA est de 30 à 40% plus petit que celui obtenu en appliquant TA.

En FP/FIFO, nous pouvons tirer la même conclusion que celle effectuée avec la politique de service FIFO concernant la domination entre FA et TA ou NC. Ainsi comme prévu, FA n'est ni dominée, ni ne domine TA ou NC. Cependant, sur les 6412 flux de la configuration, FA obtient un délai inférieur à celui de NC (*resp.* TA) pour 5758 (*resp.* 5104) flux. De plus, en moyenne, les délais déterminés en utilisant FA sont inférieurs de l'ordre de 17.55% par rapport à ceux calculés avec NC et de 8.01% par rapport à ceux obtenus en utilisant TA.

6.5 Conclusion

La différenciation de service dans les commutateurs est de plus en plus demandée et implémentée par les industriels. Par exemple, l'AFDX d'Airbus admet deux niveaux de priorités statiques en FP/FIFO. Au cours de la thèse de Nassima, nous avons étendu FA à la politique de service FP/FIFO pour un nombre quelconque $m \geq 1$ de classe de trafic. Cette extension inclut le modèle réseau et les formules de calculs. Tous les travaux de Nassima ont été formellement démontrés dans sa thèse.

L'expérimentation menée sur une configuration industrielle nous a permis de prouver le passage à l'échelle de FA, pour des configurations FP/FIFO industrielles composées de milliers de flux avec des temps de calculs de l'ordre de quelques secondes.

Nous avons également démontré avec cette configuration industrielle qu'il n'y a pas de relation de dominance entre FA et NC ou TA. Cependant, les délais de bout-en-bout calculés par FA sont globalement meilleurs que ceux de NC et TA.

Ces travaux ont donné lieu à une conférence internationale [Ben+17a].

La méthode Forward End-to-End Delay Analysis en CBS

Sommaire

7.1	Introduction	95
7.2	Modèle réseau en CBS	95
7.3	Le calcul du délai pire cas d'une trame en CBS	97
7.3.1	Maximisation de l'arriéré de travail généré par la classe X_i en fonction du crédit	99
7.3.2	Interférence maximum générée par les classes plus prioritaires	102
7.3.3	Le temps d'attente maximum dû au réapprovisionnement de crédit	102
7.3.4	Interférence maximum générée par les classes moins prioritaires	103
7.3.5	Formules sur le délai calculé par FA en CBS	104
7.4	Expérimentations	105
7.4.1	Introduction	105
7.4.2	Transposition de l'AVB au modèle réseau de FA	105
7.4.3	Configurations réseau AVB	107
7.5	Conclusion	108

7.1 Introduction

FA a été développée, dans les chapitres 5 et 6, exclusivement dans un contexte avionique et plus particulièrement pour le protocole AFDX. Le modèle réseau défini dans ces chapitres est cependant plus généraliste et ne se limite pas nécessairement à l'avionique.

Durant la thèse de Nassima, nous nous sommes intéressés au protocole réseau *Audio/Vidéo Bridging Switched Ethernet* (AVB). L'AVB a été développé initialement pour le transport de flux audio et vidéo dans le domaine de l'automobile. L'AVB est un réseau de paquet basé sur Ethernet, il a été standardisé [80211].

L'AVB applique également de la différenciation de service au niveau des flux. Sa politique de service est basée sur FP/FIFO. Cependant, FP/FIFO favorise les flux de plus hautes priorités, à contrario des flux de plus petites priorités. Pour limiter l'impact des flux de plus hautes priorités, AVB ajoute à FP/FIFO, un crédit pour chaque classe de trafic (niveau de priorité). Ce crédit suit le protocole *Credit Based Shaping* (CBS). Selon la valeur du crédit, une trame est transmise ou retardée même si aucune trame de plus haute priorité n'est en attente de transmission.

Paragraphe 7.2, nous complétons le modèle détaillé paragraphe 6.2 afin d'introduire l'algorithme CBS et ainsi l'adapter à l'AVB. Les formules de FA sont étendues à la politique de service CBS dans le paragraphe 7.3. Des expérimentations sont menées paragraphe 7.4 avant de conclure, paragraphe 7.5.

7.2 Modèle réseau en CBS

Le modèle réseau détaillé paragraphe 6.2 est toujours valable pour AVB mais il doit être complété afin de modéliser l'algorithme basé sur le crédit, CBS. En particulier, une priorité P_i comprise entre 1 et m (avec $m \in \mathbb{N}^*$) est attribuée à chaque flux $v_i \in \Gamma$. Tous les flux de même priorité forment une classe de trafic. Comme en FP/FIFO, les flux de chaque classe de trafic X_i (avec $1 \leq i \leq m$) sont transmis selon la politique FIFO. À chaque classe de trafic X_i , un crédit est associé dans chaque nœud h . Le modèle de nœud en CBS est illustré Fig. 7.1.

Afin de modéliser les contraintes liées à l'évolution du crédit, nous introduisons HPX_i^h , l'ensemble des classes de trafic de priorité strictement supérieure à P_i , la priorité de X_i traversant le nœud h . Les termes HPX_i^h et hp_i^h (défini paragraphe 6.2) sont fortement liés : $hp_i^h = \{v_j \in Y \mid Y \in HPX_i^h\}$.

Le crédit est initialisé à 0 au démarrage du système. Si son crédit est supérieur ou égal à zéro, une classe de trafic peut transmettre une trame (selon la politique de service FP/FIFO). Lorsqu'une classe de trafic X_i transmet une trame sur un nœud h , son crédit diminue linéairement sur le temps avec un coefficient constant de $\sigma_{X_i}^{h,-}$ appelé aussi *send slope*. À contrario dans un nœud h , si une classe de trafic X_i dispose de trames en attente dans sa file sans pouvoir transmettre, son crédit augmente linéairement avec le temps à un rythme constant de $\sigma_{X_i}^{h,+}$ appelé *idle slope*.

À noter que le crédit d'une classe de trafic ne peut être strictement positif que si elle a encore des trames prêtes à être transmises dans un nœud h . Ainsi, dès lors que la file est vide, le crédit repasse

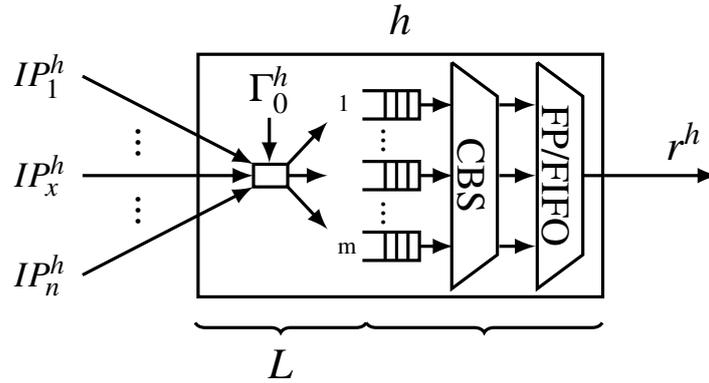


FIGURE 7.1 – Modèle de nœud CBS.

automatique à zéro.

En résumé à un instant t , une classe de trafic X_i dans un nœud h peut transmettre une trame que si elle est la file de plus haute priorité telle que :

- (i) il n'y a pas d'autres trames en cours de transmission (ordonnancement non-préemptif) ;
- (ii) elle a au moins une trame en attente ;
- (iii) elle a un crédit supérieur ou égal à zéro.

Pour chaque classe X_i (avec $1 \leq i \leq m$) et chaque nœud traversé h , le choix des valeurs des *send slope* $\sigma_{X_i}^{h,-}$ et des *idle slope* $\sigma_{X_i}^{h,+}$ est fait statiquement, avant le démarrage du système. Ces valeurs doivent respecter deux contraintes :

- Un *idle slope* d'une classe trafic du nœud h a un impact sur la vitesse de transmission des trames de sa classe de trafic. Ainsi, un *idle slope* s s'exprime en bits/s et correspond à une réservation de la bande passante du nœud h . Si on note r^h son débit du port de sortie du nœud h , alors la condition suivante doit être respectée :

$$\sum_{i=1}^m \sigma_{X_i}^{h,+} \leq r^h \quad (7.1)$$

- Pour une classe X_i ($1 \leq i \leq m$), le *send slope* impacte le temps nécessaire au rechargement de crédit, et donc le temps où les autres classes peuvent transmettre. Ainsi, il définit la bande passante (en bits/s) allouée aux autres classes :

$$\sigma_{X_i}^{h,-} = r^h - \sigma_{X_i}^{h,+} \quad (7.2)$$

Un exemple de la transmission des trames avec la politique de service CBS est illustré Fig. 7.2. Il est composé de trois classes de trafic avec par ordre de priorités décroissantes (A, B, C) dont les flux traversent un unique nœud h . Afin de simplifier l'exemple, nous n'avons pas représenté la courbe de crédit de la classe C (pas nécessaire car il n'y a qu'une seule trame de classe C qui arrive seule dès l'instant 0).

La trame C_1 est transmise dès son arrivée à l'instant 0. Deux trames du flux A (A_1 et A_2) et une trame du flux B (B_1) arrivent sur le nœud h à l'instant 3. Ainsi, des trames des classes A et B sont

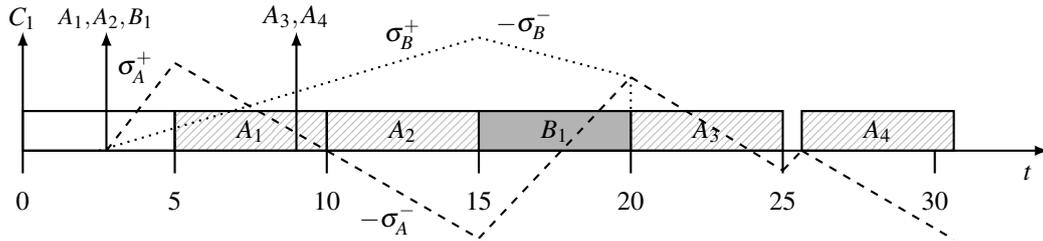


FIGURE 7.2 – Un exemple d’ordonnement suivant le régulateur CBS.

prêtes mais en attente de transmission. Leurs crédits respectifs augmentent au rythme de leurs *idle slope*. A l’instant 5, après la fin de transmission de C_1 , la trame A_1 commence sa transmission et son crédit baisse selon son *send slope*. En $t = 10$, la classe A (plus prioritaire) a un crédit nul donc la trame A_2 est aussitôt transmise. Cependant, à l’instant 15, même s’il lui reste deux trames à transmettre (A_3 et A_4 arrivées sur h à $t = 9$), la classe A perd la main car son crédit est négatif. La trame B_1 est ainsi transmise et avec un crédit redevenu positif, la classe A peut transmettre A_3 à l’instant 20. Cependant, à $t = 25$, il reste une seule trame à transmettre, A_4 mais un délai supplémentaire est nécessaire pour que son crédit revienne à zéro (effectif à l’instant $t = 26.25$).

7.3 Le calcul du délai pire cas d’une trame en CBS

Dans la définition même de FA, la politique de service est nécessairement conservatrice (cf. définition 5.9). C’est-à-dire qu’elle n’insère aucun temps creux (ne transmet pas de trame) tant qu’il y a des trames en attente. Cependant, avec la politique CBS, même si une trame est prête à être transmise et qu’il n’y a pas de trame plus prioritaire en attente de transmission, elle peut ne pas être transmise si son crédit est insuffisant (négatif) (cf. la trame A_4 en $t = 25$, Fig. 7.2). Ainsi, la définition de l’instant oisif (cf. définition 6.3) doit être étendue au contexte CBS afin que la notion de période d’activité (cf. définition 6.4) reste consistante.

Définition 7.1

Pour un nœud CBS h , un instant oisif de niveau P_i (pour la classe X_i), est un instant tel que toutes les trames de $v_j \in shp_i^h$ (c-à-d. $P_j \leq P_i$), arrivées avant cet instant, ont été totalement transmises et que le crédit est revenu à 0.

Le théorème 7.1 définit le pire scénario retardant une trame f_i de la classe X_i , générée flux v_i et arrivant sur un nœud CBS h ($v_i \in \Gamma^h$) à l’instant t (avec $t \geq 0$). Ce scénario se base sur le terme $dep_i^h(t)$ (cf. définition 6.5) qui représente en FP/FIFO, l’instant de départ pire cas de la trame f_i qui est arrivée sur le nœud h à l’instant t . Cet instant correspond au dernier instant auquel une trame de plus haute priorité peut arriver pour retarder f_i . Avec la politique de service CBS, cet instant de départ doit être modifié pour d’inclure le délai supplémentaire dû au rechargement de crédit afin qu’à l’instant $dep_i^h(t)$, le crédit soit suffisant (≥ 0) pour transmettre f_i :

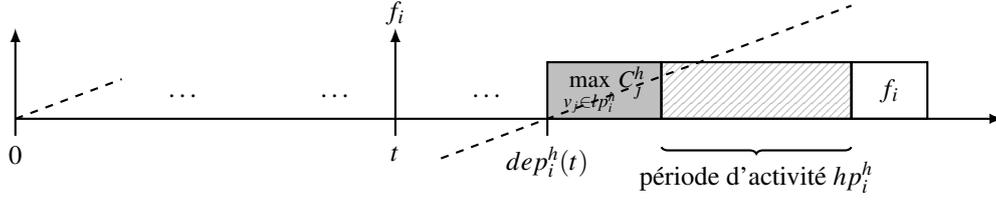


FIGURE 7.3 – Scénario pire cas pour la politique de service CBS.

Lemme 7.1

Soit f_i une trame générée par un flux v_i et arrivant sur (ou en étant générée par) un nœud CBS h à l'instant t . Le nœud h commence la transmission de la trame f_i , dans le pire cas à l'instant :

$$dep_i^h(t) = W_i^h(t) - \left(1 + \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} \right) C_i^h$$

Démonstration : Le calcul de $W_i^h(t)$ inclut, par définition, les temps d'attente nécessaires à l'algorithme CBS afin de réapprovisionner le crédit utilisé pour chaque trame. Cependant, le crédit que la trame f_i va consommer n'a pas besoin d'être récupéré. Ainsi, ce crédit consommé par f_i dans le nœud h est égal à $\sigma_{X_i}^{h,-} \cdot C_i^h$ et le temps de réapprovisionnement de crédit correspondant est alors $\frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} C_i^h$. La démonstration est détaillée dans [Ben+18] ; [Ben18]. \square

Nous définissons maintenant le scénario pire cas :

Théorème 7.1

Dans un nœud CBS h , le scénario maximisant l'arriéré de travail retardant une trame f_i de classe X_i , générée par le flux v_i et arrivant sur h à l'instant t ($t \geq 0$) est défini par :

- (i) un crédit de la classe X_i nul au démarrage du système (instant 0) ;
- (ii) un crédit qui fluctue entre les émissions de trames : le crédit diminue quand une trame de X_i est en cours de transmission et augmente dans les autres cas, selon respectivement $\sigma_{X_i}^{h,-}$ et $\sigma_{X_i}^{h,+}$;
- (iii) à l'instant $dep_i^h(t)$, le crédit est nul (par définition) et la trame f_i est la prochaine trame de la classe X_i qui sera transmise mais à l'instant $dep_i^h(t) - \varepsilon$ (avec $\varepsilon \rightarrow 0^+$) une trame de priorité inférieure arrive et est transmise directement ;
- (iv) suite à cette transmission toutes les trames de hp_i^h qui arrivent sont transmises ;
- (v) finalement dès qu'il n'y a plus de trame de hp_i^h en attente, la trame f_i est transmise.

Démonstration : Ce scénario est illustré Fig. 7.3. La démonstration est détaillée dans [Ben+18] ; [Ben18]. \square

Par rapport à l'expression de $W_i^h(t)$, la prise en compte du temps de rechargement implique une

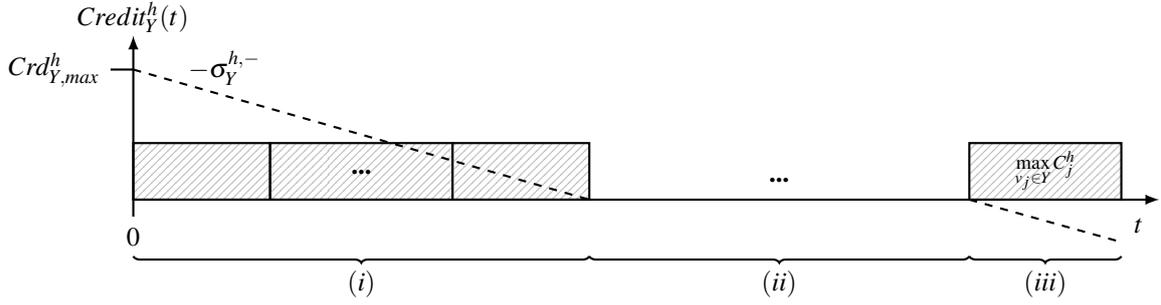


FIGURE 7.4 – Scénario pire cas défini et prouvé par Bordoloi *et al.* [Bor+14] maximisant le temps de transmission total généré par une classe Y sur un nœud CBS durant un intervalle de temps de durée t .

modification de la formule (6.1) :

$$W_i^h(t) = WLP_i^h(t) + WSP_i^h(t) + WHP_i^h(t) + WNIL_i^h(t) \quad (7.3)$$

Le terme $WNIL_i^h(t)$ définit pour la trame f_i générée par le flux v_i analysé, le temps supplémentaire nécessaire au rechargement du crédit pour l'envoi de f_i . Par exemple Fig. 7.2, le terme $WNIL_i^h(t)$ de la trame A_4 correspond au délai entre la fin de la transmission de A_3 et le début de celle de A_4 .

Pour les flux de même priorité (*cf.* paragraphe 7.3.1), nous définissons une troisième borne supérieure en se basant uniquement sur le crédit. Ces résultats sont en partie réutilisés pour les flux de plus hautes priorités dans le paragraphe 7.3.2. Un nouveau délai lié au rechargement est démontré paragraphe 7.3.3 et nous démontrons, paragraphe 7.3.4, que pour les flux de plus basse priorité, le terme $WLP_i^h(t)$ (*cf.* théorème 6.4) reste inchangé. Finalement, les formules de FA étendues à CBS sont résumées paragraphe 7.3.5.

7.3.1 Maximisation de l'arriéré de travail généré par la classe X_i en fonction du crédit

Afin de définir cette nouvelle borne supérieure garantie, nous nous focalisons comme précédemment (*cf.* théorème 6.4) sur un lien entrant IP_x^h (avec $1 \leq x \leq n$). L'objectif est d'analyser l'algorithme CBS dans le nœud h_x précédant h (via IP_x^h) afin de limiter l'arriéré de travail maximum, $W_{x,i}^h(t)$, entrant sur h par ce lien.

Pour procéder au calcul de cette nouvelle borne, nous nous basons sur les travaux de Bordoloi *et al.* [Bor+14]. Dans ce papier, les auteurs se sont focalisés sur l'algorithme CBS et la classe de trafic de plus haute priorité. Ils ont construit le pire scénario maximisant l'arriéré de travail émis par un nœud h , durant un intervalle de longueur t (avec $t \geq 0$) pour la classe de trafic de plus haute priorité, dénotée Y . Ce scénario est décrit Fig. 7.4.

Dans ce scénario, le début et la fin de l'intervalle $[0, t]$ sont toujours identiques (quelle que soit la valeur de t) :

- À l'instant 0 (i), le crédit de la classe de plus haute priorité Y est à son maximum et la classe Y

transmet jusqu'à ce que le crédit retombe à 0. À noter que cet instant ne correspond pas nécessairement à une fin de transmission d'une trame. L'objectif est d'avoir consommé exactement tout le crédit pour obtenir 0, soit une longueur de $\frac{Crd_{Y,max}^h}{\sigma_Y^{h,-}}$;

- En fin d'intervalle, le pire cas est défini (iii) quand la plus longue trame de Y est émise en dernier, à partir d'un crédit nul et dont la transmission se termine en t , $\max_{v_j \in Y} C_j^h$.

Entre les deux, (ii), le crédit démarre à zéro et finit à zéro. Il est possible de déterminer l'interférence maximale générée par Y sur cet intervalle.

Lemme 7.2

Pour tout intervalle de longueur t ($t \geq 0$), l'interférence maximale, générée par une classe de trafic Y dont le crédit est nul au début et à la fin de l'intervalle, est égale à :

$$t \cdot \frac{\sigma_Y^{h,+}}{\sigma_Y^{h,+} + \sigma_Y^{h,-}}$$

Démonstration : La démonstration est directe dès lors que le crédit diminue de $\sigma_{X_i}^{h,-}$ et doit être compensé à la fin de l'intervalle [Ben+18]. \square

Ce scénario (cf. Fig. 7.4) n'a été démontré par ses auteurs [Bor+14] que pour la classe de trafic la plus prioritaire. En effet, pour la classe de trafic de plus haute priorité Y , le crédit augmente uniquement pendant la transmission d'une trame de plus petite priorité. Une seule ne peut retarder un flux de Y donc le crédit maximum ($Crd_{Y,max}^h$) est égal à la durée de transmission de la plus longue trame fois $\sigma_Y^{h,+}$.

Nous avons généralisé ce scénario pire cas, illustré Fig. 7.4, à n'importe quelle classe de trafic en déterminant son crédit maximum.

Lemme 7.3

Pour une classe de trafic X_i avec $1 \leq i \leq m$, le crédit maximum accumulé dans le nœud h est :

$$Crd_{X_i,max}^h = \left(\max_{v_j \in I_p_i^h} C_j^h + BP^{X_i} \right) \sigma_{X_i}^{h,+} \quad (7.4)$$

Où BP^{X_i} désigne la plus longue période d'activité (cf. définition 6.4) générée par l'ensemble des classes plus prioritaires HPX_i^h (ou l'ensemble des flux de hp_i^h), obtenu par le plus petit point fixe de :

$$BP^{X_i}(0) = \sum_{j \in hp_i^h} C_j^h \quad \text{et} \quad BP^{X_i}(k+1) = \sum_{j \in hp_i^h} \left\lceil \frac{BP^{X_i}(k)}{T_j} \right\rceil C_j^h$$

Démonstration : Le crédit d'une classe augmente dès lors qu'une de ces trames est en attente de transmission et que d'autres trames sont transmises. Pour obtenir le crédit maximal de la classe X_i , il faut supposer que son crédit est à zéro quand une trame f_i de X_i arrive sur le nœud h . Dans le pire cas,

- la plus longue trame de plus petite priorité atteint h juste avant f_i et est donc transmise avant f_i .

- De plus, pour retarder au maximum f_i (et donc maximiser son crédit) cette transmission est suivie par la plus longue période d'activité des flux de priorité supérieure à la classe X_i .

Ce calcul de la plus longue période d'activité de transmission de trames est une application de celle déterminée pour les tâches non-préemptives sporadiques ou périodiques dans un système mono-processeur [RCM96]; [Spu96]. \square

Remarque 7.1

Le point fixe du lemme 7.3 existe si, et seulement si, $\sum_{v_j \in hp_i^h} \frac{C_j^h}{T_j} \leq 1$ (assurée par construction).

À partir du scénario pire cas de Bordoloi et al [Bor+14], illustré Fig. 7.4 et du calcul du crédit maximum (cf. lemme 7.3), nous pouvons en déduire l'interférence maximale générée par une classe de trafic X_i avec ($1 \leq i \leq n$).

Théorème 7.2

Le temps de transmission total généré par une classe de trafic X_i sur un nœud CBS durant un intervalle de durée t est borné par $Bord_{X_i}^h(t)$:

$$Bord_{X_i}^h(t) = \underbrace{\frac{Crd_{X_i,max}^h}{\sigma_{X_i}^{h,-}}}_{(i)} + \underbrace{\max_{v_j \in X_i^h} C_j^h}_{(iii)} + \underbrace{\left[\left(t - \frac{Crd_{X_i,max}^h}{\sigma_{X_i}^{h,-}} - \max_{v_j \in X_i} C_j^h \right) \frac{\sigma_{X_i}^{h,+}}{\sigma_{X_i}^{h,+} + \sigma_{X_i}^{h,-}} \right]^+}_{(ii)}$$

Où $[Expr.]^+ = \max(0, Expr.)$

Démonstration : Ce Théorème a été démontré [Ben+18] et est une directe application du lemme 7.3 au scénario dépeint Fig. 7.4. \square

Nous appliquons le théorème 7.2 pour borner l'arriéré temporel généré par les flux de la classe X_i entrant sur le nœud h via le lien IP_x^h (avec $1 \leq x \leq n$) depuis le nœud h_x . Nous complétons ainsi l'expression de $W_{x,i}^h(t)$.

Théorème 7.3

L'interférence maximale générée par les flux de la classe X_i entrant un nœud CBS h via le lien IP_x^h (avec $1 \leq x \leq n$) et retardant une trame f_i générée par le flux v_i (de la classe X_i) arrivant sur h à l'instant t ($t \geq 0$) est :

$$\min \left[\sum_{v_j \in sp_{x,i}^h} rbf_j^h(t), LinkRate_{x,i}^h(t), ShaperRate_{x,i}^h(t) \right] \quad (7.5)$$

Avec $ShaperRate_{x,i}^h(t) = Bord_{X_i}^{h_x} \left(\frac{r^{h_x}}{r^h} t + \max_{v_j \in shp_{x,i}^h} C_j^h \right)$.

Démonstration : La démonstration est détaillée [Ben+18] et est une application de la fonction $Bord_{X_i}^h(t)$ sur le nœud h_x sur le plus grand intervalle qui peut interférer avec la trame f_i sur le nœud h (cf. lemme 5.1). \square

7.3.2 Interférence maximum générée par les classes plus prioritaires

En nous appuyant sur la fonction $Bord_{X_i}^h(t)$ du théorème 7.2, nous allons, comme pour les flux de même priorité, fournir une seconde borne garantie pour l'interférence générée par les flux plus prioritaires entrant par chaque lien physique.

Théorème 7.4

L'interférence maximale générée par les flux des classes plus prioritaires que la classe X_i , entrant un nœud CBS h via le lien IP_x^h (avec $1 \leq x \leq n$) et retardant une trame f_i générée par le flux v_i (de la classe X_i) arrivant sur h à l'instant t ($t \geq 0$) est :

$$\min \left[\sum_{v_j \in hp_{x,i}^h} rbf_j^h \left(dep_i^h(t) \right), \sum_{Y \in HPP_Y^h} Bord_Y^{h_x-1} \left(\frac{r^{h_x-1}}{r^h} dep_i^h(t) + \max_{v_j \in SHP_Y^h} C_j^h \right) \right] \quad (7.6)$$

Démonstration : Le premier terme se déduit directement du théorème 6.4. Le second terme est une application du théorème 7.2 pour chaque classe de trafic Y de priorité supérieure à celle de X_i sur l'intervalle le plus long sur le nœud $h_x - 1$, et pouvant retarder f_i , $\frac{r^{h_x-1}}{r^h} dep_i^h(t) + \max_{v_j \in SHP_Y^h} C_j^h$ (cf. lemme 5.1). \square

Remarque 7.2

Pour les flux plus prioritaires générés directement par le nœud CBS h ($v_j \in hp_{0,i}^h$), le théorème 7.4 n'est pas applicable. Ainsi, pour ces flux, nous conservons la formule (6.4) reste inchangée.

7.3.3 Le temps d'attente maximum dû au réapprovisionnement de crédit

L'algorithme *Credit Based Shaping* est une politique de service non-conservative (cf. définition 5.9). Ainsi, si la crédit d'une classe X_i est négatif, une trame de la classe X_i peut être retardée dans une file d'attente, même si aucune autre trame n'est en cours de transmission. Le terme $WNIL_i^h(t)$ de la formule (7.3) correspond au délai supplémentaire nécessaire à la classe X_i pour recharger son crédit afin de pouvoir transmettre la trame f_i analysée.

Théorème 7.5

Le temps d'attente maximum dû au réapprovisionnement de crédit d'une trame f_i générée par un flux

v_i appartenant à la classe X_i arrivant dans un nœud h à l'instant t est :

$$WNIL_i^h(t) = \left[\left(WSP_i^h(t) - C_i^h \right) \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} - \sum_{v_j \in hp_i^h} \left\lfloor \frac{t}{T_j} \right\rfloor C_j^h \right]^+$$

Où $[Expr.]^+ = \max(0, Expr.)$

Démonstration : $WNIL_i^h(t)$ se détermine en

- (i) calculant le temps nécessaire pour recréditer le crédit consommé par l'ensemble des flux de la classe X_i (à l'exception de la trame f_i) :

$$\left(WSP_i^h(t) - C_i^h \right) \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}}$$

- (ii) soustrayant le temps pendant lequel le crédit a déjà été réapprovisionné. Ce temps est au minimum composé par le temps de transmission accumulé des trames des flux de plus haute priorité durant l'intervalle $[0, t[$:

$$\sum_{v_j \in hp_i^h} \left\lfloor \frac{t}{T_j} \right\rfloor C_j^h$$

Pour finir, le maximum de cette expression avec 0 est pris afin de s'assurer que le terme $WNIL_i^h(t)$ ne soit jamais négatif. \square

7.3.4 Interférence maximum générée par les classes moins prioritaires

Concernant les flux de priorités inférieures dans le délai subi par la trame analysée avec CBS, le scénario est différent de celui défini en FP/FIFO (cf. théorème 6.1) mais la contribution définie pour ces flux par le théorème 6.4 reste valide.

Théorème 7.6

Le délai dû aux trames de priorités inférieures subi par une trame f_i générée par un flux v_i appartenant à la classe X_i arrivant dans un nœud h à l'instant t (≥ 0) est :

$$WLP_i^h(t) = \max_{v_j \in lp_i^h} C_j^h$$

Démonstration :

Si une trame de lp_i^h est transmise par h durant l'intervalle $[0, t[$, le crédit de la classe X_i augmente durant cette transmission avec un ratio de $\sigma_{X_i}^{h,+}$. Mais ce temps de réapprovisionnement a déjà été comptabilisé dans le terme $WNIL_i^h(t)$.

Afin de retarder la trame f_i , seule une trame de priorité inférieure de longueur maximale arrivant

à l'instant $dep_i^h(t) - \varepsilon$, avec $\varepsilon \rightarrow 0^+$ peut arriver et donc être comptabilisée. \square

7.3.5 Formules sur le délai calculé par FA en CBS

Une borne supérieure sur le délai pire cas subi par une trame en traversant un nœud CBS, peut être déterminée à l'aide du théorème suivant :

Théorème 7.7

L'arriéré de travail pire cas retardant une trame f_i générée par le flux v_i appartenant à la classe X_i , arrivée dans un nœud CBS h à l'instant t , est borné par le plus petit point fixe déterminé par la formule (7.7) si la condition sur la charge $\sum_{v_j \in \Gamma^h} \frac{C_j}{T_j} < 1$ est respectée.

$$\begin{cases} W_{t,i}^h(0) = C_i^h \\ W_{t,i}^h(k+1) = WLP_i^h + WSP_i^h(t) + WHP_{t,i}^h(k) + WNIL_i^h(t) \end{cases} \quad (7.7)$$

Où

$$\begin{aligned} WLP_i^h(t) &= \max_{v_j \in lp_i^h} C_j^h \\ WSP_i^h(t) &= \sum_{v_j \in \Gamma_{0,i}^h} rbf_j^h(t) + \sum_{x=1}^n \min \left[\sum_{v_j \in sp_{x,i}^h} rbf_j^h(t), LinkRate_{x,i}^h(t), ShaperRate_{x,i}^h(t) \right] \\ WHP_{t,i}^h(k) &= \sum_{v_j \in hp_{0,i}^h} rbf_j^h \left(dep_{t,i}^h(k) \right) \\ &+ \sum_{x=1}^n \min \left[\sum_{v_j \in hp_{x,i}^h} rbf_j^h \left(dep_{t,i}^h(k) \right), \sum_{Y \in HP_Y^h} Bord_Y^{h_x-1} \left(\frac{r^{h_x-1}}{r^h} dep_{t,i}^h(k) + \max_{v_j \in SHP_Y^h} C_j^h \right) \right] \\ WNIL_i^h(t) &= \left[\left(WSP_i^h(t) - C_i^h \right) \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} - \sum_{v_j \in hp_i^h} \left\lfloor \frac{t}{T_j} \right\rfloor C_j^h \right]^+ \end{aligned}$$

$$\text{Avec } dep_{t,i}^h(k) = W_{t,i}^h(k) - \left(1 + \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} \right) C_i^h$$

Le calcul est effectué itérativement jusqu'au premier point fixe, $W_{t,i}^h(k+1) = W_{t,i}^h(k)$ et $W_i^h(t)$ dénote, alors, ce plus petit point fixe.

Démonstration : Le résultat est une application directe des Théorèmes 7.3, 7.4 et 7.5. \square

Remarque 7.3

L'algorithme de calcul du délai de bout en bout de trames dans un réseau de paquets utilisant la

politique de service CBS en utilisant FA est similaire à celui détaillé en FP/FIFO, algorithme 2 du chapitre 6.

Remarque 7.4

La condition d'arrêt du paragraphe 5.4.4 est toujours valide avec la nouvelle définition de l'instant oisif en CBS (cf. définition 7.1).

7.4 Expérimentations

7.4.1 Introduction

Pour les expérimentations en CBS, la principale difficulté est de trouver des résultats auxquels nous pouvons confronter ceux déterminés avec FA. Dans la littérature, nous n'avons trouvé que peu de configurations suffisamment détaillées et/ou d'implémentations disponibles pour les approches concurrentes.

Par exemple, dans [LG17], les auteurs appliquent la méthode des trajectoires sur une configuration détaillée dans [Ald+12]; [Ste+12] mais pour laquelle les valeurs des *idle slot* et *send slot* ne sont pas données. Ainsi, après application à la main de leurs formules (de TA) pour un flux ne traversant qu'un seul commutateur et ne croisant qu'un seul autre flux, il nous est impossible de retrouver leurs résultats. Ainsi, dans ce chapitre, les seules expérimentations menées le seront sur la configuration détaillée dans [DRE12] et en comparant les résultats de FA avec ceux de *Compositional Performance Analysis* (CPA).

Nous présentons dans le paragraphe 7.4.2 la transposition d'une configuration réseau AVB (cf. paragraphe 2.3.2) au modèle FA. Ensuite paragraphe 7.4.3, nous présentons la configuration étudiée et nous comparons nos résultats à ceux obtenus avec CPA.

7.4.2 Transposition de l'AVB au modèle réseau de FA

Dans une configuration AVB (cf. paragraphe 2.3.2) l'ensemble des flux est réparti en plusieurs classes FIFO de priorités fixes. Une configuration AVB a deux types de classes de trafic : les classes *Stream Reservation* (SR) et le trafic *Best Effort* (BE). Selon le standard AVB, jusqu'à huit classes SR sont possibles. Il y a néanmoins classiquement deux classes SR : A et B (classe A avec une priorité supérieure à celle de la classe B). Le niveau de priorité de la classe BE est toujours plus petit que celle de l'ensemble des classes SR.

En AVB, les entrées/sorties (cf. ES) du réseau correspondent aux *Electronic Control Unit* (ECU). Les nœuds de FA représentent les points de multiplexage du réseau. Ils modélisent ainsi les ports de sortie des ECUs et des commutateurs AVB.

Le protocole IEEE 802.1Qat [Gro10] définit pour chaque flux AVB à son entrée sur le réseau, trois paramètres : *Class Measurement Interval* (CMI), *Maximum Interval Frames* (MIF) et *Maximum*

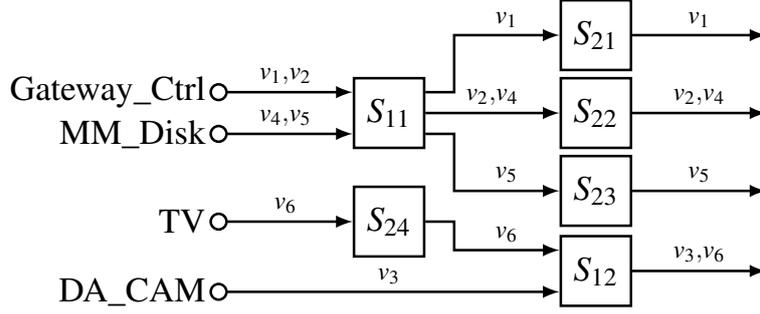


FIGURE 7.5 – Modélisation par FA de la configuration illustrée Fig. 2.9.

Frame Size (MFS). Selon ce modèle, un flux AVB génère au plus MIF trames, de taille maximale MFS bits, durant un intervalle de temps de longueur CMI. La bande passante allouée à chaque flux est donc limitée par $\frac{MIF \times MFS}{CMI}$.

Le contrat de trafic défini par un flux v_i dans le modèle de FA (C_i^h, T_i, P_i) peut être calculé à partir des paramètres de chaque flux AVB (CMI, MIF, MFS, paragraphe 2.3.2) à son entrée dans le réseau :

- C_i^h , la durée de transmission maximum d'une trame du flux v_i sur le nœud h , en divisant MFS par le débit en sortie du nœud h , r^h : $C_i^h = \frac{MFS}{r^h}$;
- T_i : la période est égale CMI divisé par MIF : $T_i = \frac{CMI}{MIF}$;
- P_i , tous les flux d'une classe SR X ont la même priorité : ceux de la classe SR A sont plus prioritaires que ceux de la classe SR B qui sont plus prioritaires que les flux BE.

Les flux d'un réseau AVB sont similaires aux VLs de l'AFDX : ils sont émis par un unique ES source, suivent un chemin statique jusque, par définition, un ou plusieurs destinataires. Par conséquent, comme pour les VLs de l'AFDX (cf. paragraphe 5.5.2), un délai est calculé par flux et dès lors que le flux se subdivise, le délai est propagé vers chaque destinataire. Similairement à ce qui a été fait pour l'AFDX (cf. remarque 5.6), pour chaque flux, le calcul s'arrête dans le port de sortie du dernier commutateur traversé. La Fig. 7.5 illustre la configuration de la Fig. 2.9 modélisée par FA.

Dans une configuration AVB classique, 75% de la bande passante totale est réservée aux classes SR. Ainsi nous obtenons (cf. formule (7.1)) :

$$\sum_{i=1}^m \sigma_{X_i}^{h,+} \simeq 0.75 \times r^h$$

Les paramètres *idle slope* ($\sigma_{X_i}^{h,+}$) et *send slope* ($\sigma_{X_i}^{h,-}$) de chaque flux peuvent être configurés manuellement (sous réserve que les contraintes des équations 7.1 et 7.2 soient respectées) ou automatiquement via le *Stream Reservation Protocol* IEEE 802.1Qat [Gro10].

La Fig. 7.6 définit le fonctionnement d'un port AVB. Ainsi, l'algorithme *credit based traffic shaper* (CBS) ne s'applique qu'aux classes SR et la classe *Best Effort* (BE) est connectée en direct à l'algorithme FP/FIFO. Ainsi, la classe BE étant de plus petite priorité, elle n'a pas de paramètre CBS et ses flux sont toujours transmis dès lors qu'aucun autre flux n'est en attente avec un crédit suffisant.

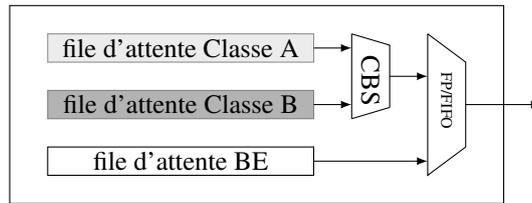


FIGURE 7.6 – Port de sortie d’un nœud CBS.

Cependant, pour être conforme au modèle FA (cf. Fig. 7.1), nous lui attribuons des paramètres *idle slope* et *send slope* afin qu’elle soit, comme avec une classe SR, en entrée de l’algorithme CBS avec la plus petite priorité, tout en n’interférant pas avec les autres flux :

$$\begin{aligned}\sigma_{BE}^{h,+} &= r^h \\ \sigma_{BE}^{h,-} &= 0\end{aligned}$$

Remarque 7.5

Les équations 7.1 et 7.2 sont définies dans la norme pour les classe SR. Ne pas les respecter pour les flux BE n’a aucun impact négatif sur le réseau.

À noter que les paramètres CMI, MIF et MFS pour les flux BE ne sont pas définis. Ainsi, pour chaque flux BE v_i et chaque nœud h , nous considérons que :

- $C_i^h = 1518$ octets, la plus longue trame Ethernet ;
- $T_i = +\infty$, le trafic BE est purement sporadique, il n’a pas de période. Cette hypothèse est valide puisque l’impact d’une classe BE est limité à une seule trame de taille maximale sur les classes de plus haute priorité (cf. théorème 7.7) ;
- P_i , la priorité la plus faible par définition.

7.4.3 Configurations réseau AVB

Dans ce paragraphe, nous appliquons FA sur deux configurations détaillées dans [DRE12] et nous comparons les résultats avec ceux obtenus par l’approche CPA détaillés dans le même article. Ces deux configurations sont composées des mêmes flux avec les mêmes caractéristiques, seule la topologie change : linéaire pour la Fig. 7.7 et en *cluster* Fig. 7.8.

Dans les deux configurations, le réseau est composé d’un unique contrôleur (Ctrl) qui envoie un message de classe A à un ensemble de nœuds destinataires (I/O). Pour ces flux de classe A, la taille maximale des trames est fixée à 336 bits et la période est de $250 \mu s$. En plus de ce flux de classe A, chaque nœud intermédiaire S_i émet une trame de trafic *Best-effort* (BE) de la taille de trame Ethernet maximale 12144 bits, à destination de tous les nœuds I/O (rejoignant le flux v_1 vers les nœuds I/O restants dans la suite de son chemin). Le débit de tous les ports de sortie h est fixé à $r^h = 1$ Gbps et la latence technologique à $L = 0.33 \mu s$.

Pour la configuration linéaire, Fig. 7.7, le flux v_1 de classe A possède 16 nœuds destinataires (I/O).

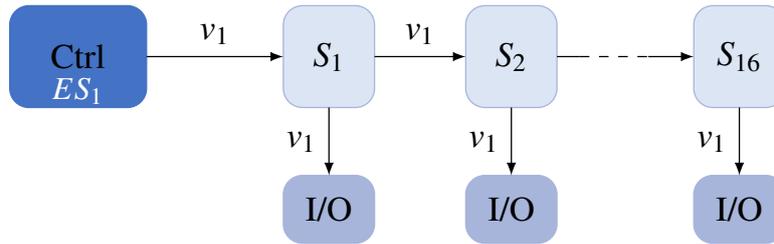


FIGURE 7.7 – Topologie linéaire.

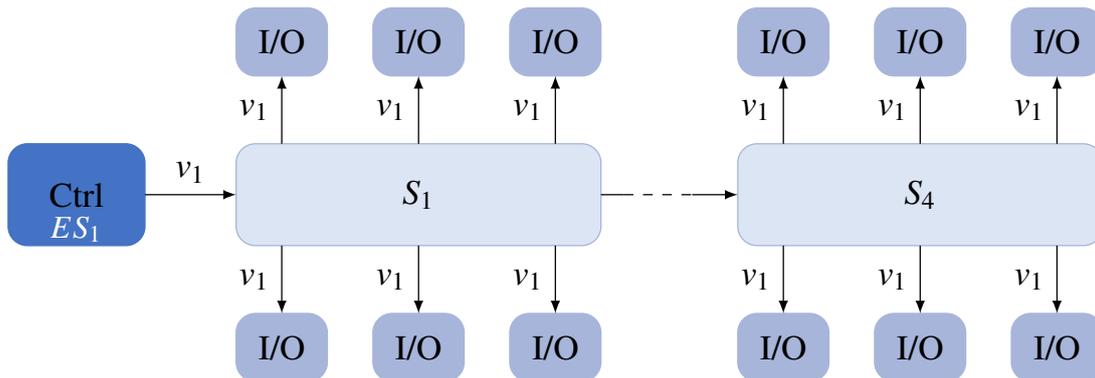


FIGURE 7.8 – Topologie en *cluster*.

Ainsi, à chaque commutateur traversé (16 au total), le flux se duplique vers un nœud de sortie et vers le commutateur suivant. Pour la topologie en étoile ou *cluster* (cf. Fig. 7.8), le flux v_1 a 24 destinataires regroupés en étoile sur quatre commutateurs consécutifs (six nœuds I/O par commutateur).

Concernant la topologie linéaire, Fig. 7.7, les délais de bout-en-bout pire cas (en μs) déterminés par FA et CPA pour le flux v_1 sont résumés Fig. 7.9. Contrairement à CPA, le délai augmente linéairement avec le nombre de commutateurs traversés. Les raisons principales tiennent à une prise en compte de l’algorithme de régulation CBS dans les nœuds précédents. Ainsi, à chaque commutateur traversé, le délai en FA ne subit que l’interférence d’une nouvelle trame BE de plus faible priorité d’une longueur de 12144 bits avec un débit de sortie des nœuds de 1 Gbps (*c-à-d.* 12.144 μs).

Pour la topologie en étoile, les délais de bout-en-bout des six destinataires reliés au même commutateur sont identiques. Ainsi, la Fig. 7.10 regroupe les quatre délais de bout-en-bout obtenus (en μs) par FA et CPA pour le flux v_1 de classe A. L’écart entre les résultats des deux approches est important et la raison principale est une nouvelle fois la prise en compte de la régulation opérée par l’algorithme CBS dans les nœuds précédents.

7.5 Conclusion

Dans ce chapitre, nous avons proposé une extension de notre approche FA au réseau AVB et à l’algorithme de régulation de crédit CBS. Le réseau AVB est un standard basé sur Ethernet utilisé

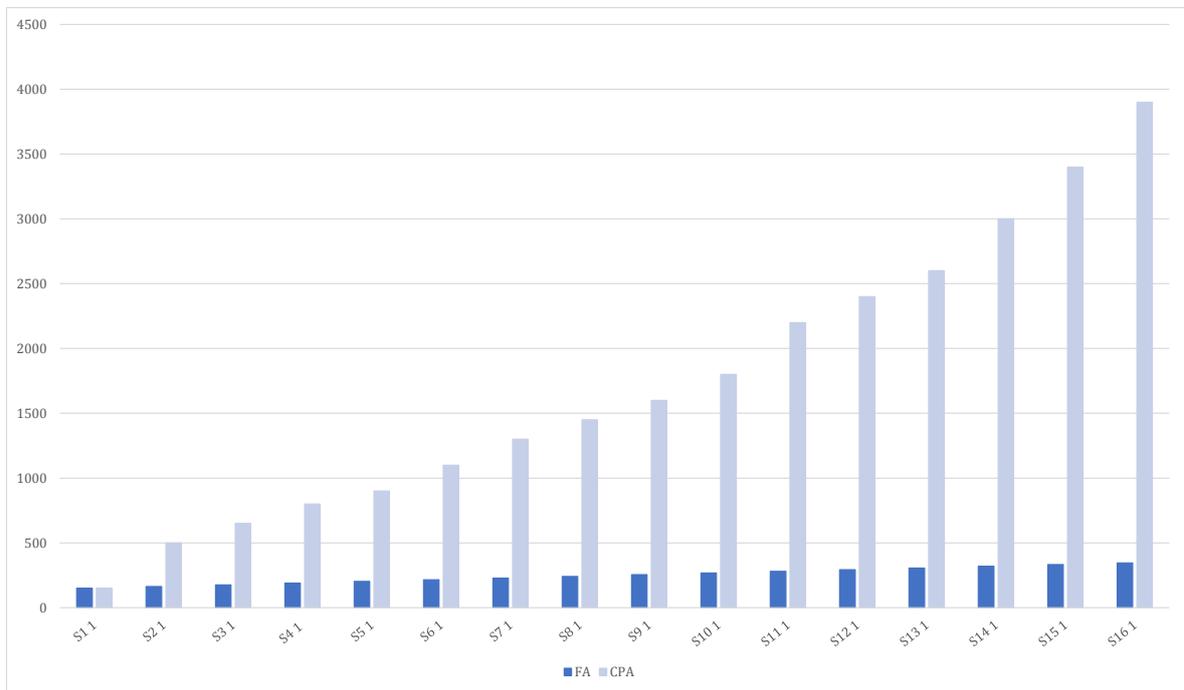


FIGURE 7.9 – Comparaison des résultats (en μs) obtenus par FA et CPA pour la configuration de la Fig. 7.7.

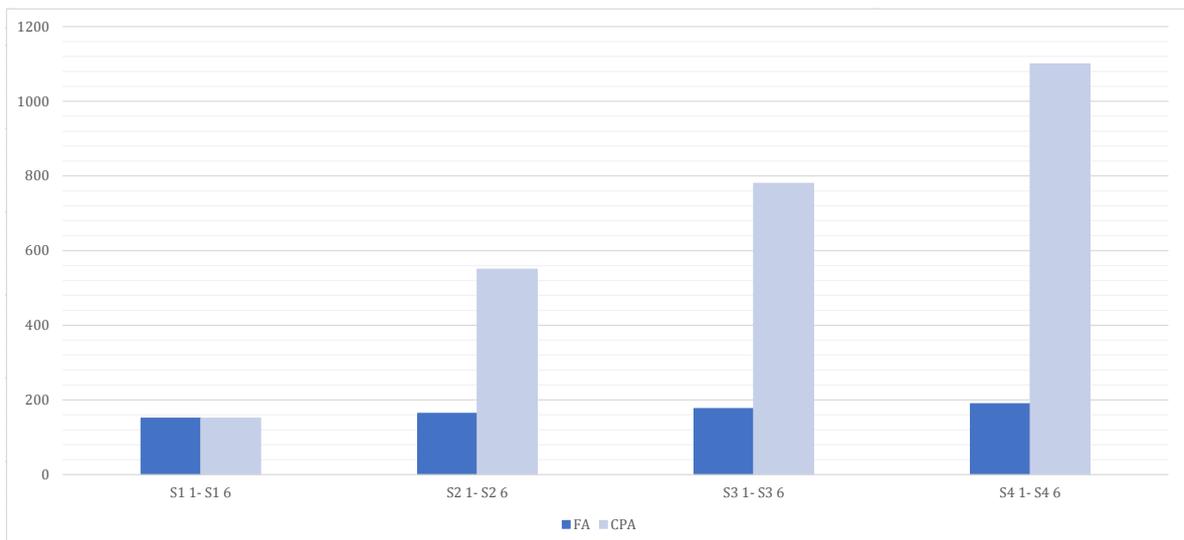


FIGURE 7.10 – Comparaison des résultats (en μs) obtenus par FA et CPA pour la configuration représentée par la Fig. 7.8.

principalement dans les applications automobiles. Ses principales caractéristiques sont sa bande passante élevée et sa garantie de qualité de service. Ce protocole afin de pouvoir être déployé dans un système temps réel critique a besoin d'une démonstration du déterminisme et donc dans notre cas, du calcul d'une borne supérieure du délai de bout en bout de traversée de l'AVB pour chaque trame.

Les travaux de ce chapitre se basent sur le modèle réseau de FA, et ses formules de calculs en FP/FIFO détaillés chapitre 6. En considérant FP/FIFO comme politique de service, l'impact des flux entre eux dépend uniquement de leurs niveaux de priorité respectifs. Le réseau AVB, en plus de FP/FIFO, utilise l'algorithme CBS basé sur une consommation de crédit afin de réguler l'interférence des classes de plus hautes priorités. Ainsi, notre extension de FA, inclut une prise en compte de CBS afin de limiter l'interférence des flux plus prioritaires mais aussi pour comptabiliser le temps supplémentaire induit pour la classe sous analyse afin de réapprovisionner son crédit pour transmettre une trame en attente.

Ces travaux ont donné lieu à une publication dans la conférence internationale [Ben+18]. Ces travaux ont également donné lieu à un stage d'ingénieur (A2 ISAE-ENSMA) et d'un stage de projet de fin d'études (A3 ISAE-ENSMA). Les objectifs de ces deux stages ont été d'analyser l'approche CPA et de leur outil python *PyCPA* afin de les intégrer à nos outils.

Dimensionnement de files d'attente

Sommaire

8.1	Introduction	113
8.2	Modèle réseau de BDF	114
8.3	BDF : Dimensionnement d'une file d'attente en nombre de trames	115
8.3.1	Principe général de l'approche BDF	115
8.3.2	Maximisation des trames entrantes	116
8.3.3	Minimisation des trames transmises par un nœud h	118
8.3.4	Calcul du dimensionnement pire cas en nombre de trames par BDF	119
8.4	Expérimentations	121
8.5	Conclusion	122

8.1 Introduction

Démontrer le déterminisme d'un réseau ne se résume pas seulement à garantir une borne supérieure sur le délai de traversée du réseau pour chaque trame mais aussi, entre autre, à garantir la non-perte de trame. La nature même des liens *full-duplex* assure l'absence de collision et donc la non perte de trame sur les liens physiques. Ainsi, les trames peuvent seulement être perdues aux points de multiplexage du réseau lorsqu'une rafle de trames arrive dans un port de sortie et entraîne le dépassement de la capacité de sa file d'attente (*buffer*). En particulier dans les réseaux asynchrones, le trafic peut être très variable menant à de grandes variations dans l'occupation des files d'attente. Pour des raisons de certification, une borne supérieure garantie sur la taille de chaque file d'attente doit être déterminée.

Le dimensionnement pire cas d'une file d'attente peut être exprimé en nombre soit de bits, soit de trames. Un dimensionnement en terme de bits est utilisé dans des applications à allocation dynamique de la mémoire où les demandes mémoire sont de tailles très différentes et où seul le nombre exact de bits nécessaires est alloué.

Dans bon nombre d'applications critiques telles que dans l'avionique, l'allocation de la mémoire est statique. La mémoire est divisée en un nombre de *slots* de taille fixe. L'objectif est de simplifier l'algorithme d'allocation et donc de réduire la complexité de la certification.

Naïvement, le pire cas en nombre de trames peut être déduit de celui en nombre de bits en le divisant par la taille de trame minimale. Cette approche naïve peut conduire à des dimensionnements en nombre de trames pessimistes. Ainsi, la plupart des approches de calculs de délais de bout-en-bout déduit une borne supérieure de l'occupation d'une file d'attente en nombre de bits directement du pire cas d'attente dans le port de sortie. Cependant, à l'aide d'un contre-exemple, il est possible de démontrer que les pire cas en termes de bits et de trames ne coïncident pas. Considérons la configuration définie Fig. 8.1.a composée d'un seul nœud h et de trois flux périodiques v_1 , v_2 et v_3 entrant sur h par trois liens différents. Les caractéristiques des flux sont :

- Durées de transmission : $C_1^h = C_2^h = 10 \mu s$ et $C_3^h = 30 \mu s$;
- Périodes : $T_1 = T_2 = 30 \mu s$ et $T_3 = 100 \mu s$.

Sans perte de généralité, nous fixons $r^h = 1 \text{ bit}/\mu s$. Ainsi, l'arriéré présent dans la file d'attente en terme de bits est équivalent au temps d'attente en μs .

À l'instant $t = 0 \mu s$ (cf. Fig. 8.1.b), une trame de chaque flux arrive sur h . L'arriéré présent dans la file d'attente est alors composé de trois trames et 50 bits. La trame 3 est transmise en premier. À l'instant $30 \mu s$, dû à leurs périodes, une nouvelle trame des flux v_1 et v_2 arrivent sur h . L'arriéré pire cas dans la file d'attente du nœud h est alors de seulement 40 bits mais de quatre trames. Sur cet exemple l'arriéré pire cas en terme de trames (instant $30 \mu s$) ne coïncide pas avec celui en nombre de bits ($0 \mu s$).

Ainsi, la recherche d'une borne supérieure sur l'occupation d'une file d'attente en nombre de trames nécessite une méthode dédiée. Dans ce chapitre, nous présentons une nouvelle approche *Buffer Dimensioning per Frame* (BDF) dédiée spécifiquement au calcul de meilleures bornes supérieures garanties sur les tailles des files d'attente.

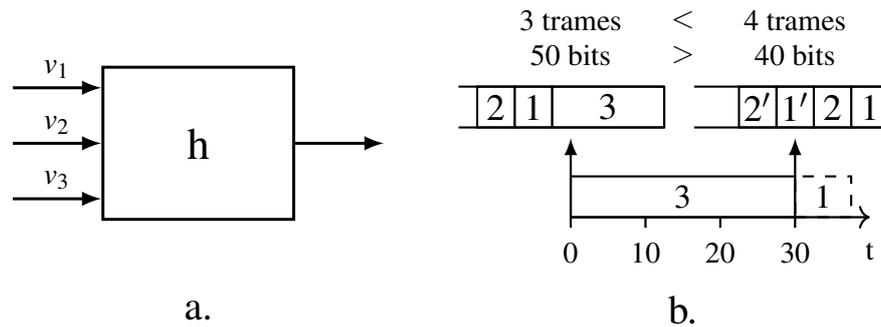


FIGURE 8.1 – Non-concordance des pires dimensionnement en termes de bits et de trames.

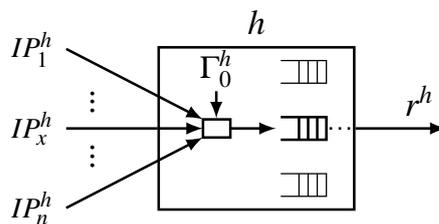


FIGURE 8.2 – Modèle du nœud h utilisé par BDF.

Paragraphe 8.2, nous présentons le modèle réseau de BDF. L'approche BDF est détaillée et démontrée paragraphe 8.3. Nous présentons les expérimentations dans le paragraphe 8.4 avant la conclusion paragraphe 8.5.

8.2 Modèle réseau de BDF

BDF peut s'appliquer sur n'importe quel réseau de paquets utilisant une politique de service conservative (*cf.* définition 5.9). Le modèle réseau est similaire à celui défini paragraphe 5.2 et la seule contrainte sur la politique de service implémentée dans les nœuds est qu'elle soit conservative et non-préemptive. Ainsi le modèle des nœuds utilisé par BDF est résumé Fig. 8.2.

Ainsi, BDF a juste besoin en entrée de :

- (i) la topologie du réseau ;
- (ii) un contrat de trafic à l'entrée du réseau pour chaque flux. Ce contrat de trafic définit une enveloppe maximale pour chaque flux v_i à son point d'entrée du réseau. Il est composé d'une taille maximale de trame $Fmax_i$ et une période T_i au sens apériodique (durée minimale entre l'émission de deux trames par v_i) ;
- (iii) une méthode de calcul des délais de bout-en-bout des trames. N'importe quelle approche, qui détermine pour chaque flux v_i et chaque nœud h du réseau traversé, une borne supérieure ($Smax_i^h$) et une borne inférieure ($Smin_i^h$) garanties sur les délais de bout-en-bout, peut être utilisée avec BDF. À noter que la politique de service n'est pas mentionnée car elle est intégrée dans les calculs par la méthode d'analyse des délais utilisés.

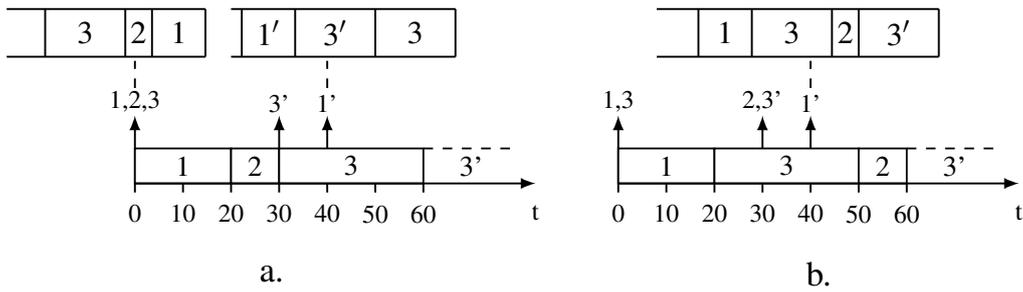


FIGURE 8.3 – Scénarios d'arrivée dans une file d'attente FIFO.

8.3 BDF : Dimensionnement d'une file d'attente en nombre de trames

8.3.1 Principe général de l'approche BDF

À notre connaissance, il n'existe pas dans la littérature de scénario pire cas pour les politiques de service conservatives et maximisant le nombre de trames présentes dans une file d'attente. Pour illustrer la complexité de ce problème, analysons de nouveau la configuration de la Fig. 8.1.a. La politique de service est FIFO et nous modifions seulement les caractéristiques des flux :

- Durées de transmission : $C_1^h = 20 \mu s$, $C_2^h = 10 \mu s$ et $C_3^h = 30 \mu s$
- Périodes : $T_1 = T_2 = T_3 = 100 \mu s$

La Fig. 8.3.a. dépeint un scénario possible d'arrivée dans lequel une trame de chaque flux arrive sur h dès l'instant 0 (pour des arrivées simultanées, la trame avec l'indice le plus petit est la première transmise). Nous considérons qu'une nouvelle trame du flux v_3 (*resp.* v_1) arrive dès l'instant $t = 30 \mu s$ (*resp.* $40 \mu s$). Sur leur nœud source, les trames sont générées avec un délai minimum de la période. Mais en fonction des trames croisées, il est possible que le délai entre deux trames consécutives soit inférieur à la période (*cf.* gigue sur activation $J_i^h = Smax_i^h - Smin_i^h$, Fig. 5.3). Pour ce scénario, l'arriéré pire cas en nombre de trames est d'au moins 3 en $t = 0 \mu s$ et $t = 40 \mu s$. Si maintenant, l'arrivée de la trame 2 est repoussée à l'instant $t = 30 \mu s$ (*cf.* Fig. 8.3.b), l'arriéré en terme de trames est de 4 en $t = 40 \mu s$.

Cet exemple illustre que la construction d'un instant critique à l'instant $t = 0 \mu s$ avec l'arrivée d'une trame de chaque flux ne maximise pas toujours le nombre de trames présentes dans un nœud simultanément. Par contre, analyser tous les scénarios d'arrivée de trames afin de trouver l'arriéré pire cas en nombre de trames mène à une explosion combinatoire. Nous allons donc nous concentrer sur la recherche d'un scénario potentiellement pessimiste afin de déterminer une borne supérieure sur le dimensionnement pire cas d'une file d'attente en nombre de trames.

Le principe de BDF consiste à :

- Maximiser le nombre de trames entrant sur un nœud, en les comptabilisant dès que possible, en utilisant les fonctions RBF (*cf.* formule (5.6));
- tout en minimisant le nombre de trames transmises par le nœud en privilégiant les trames les plus longues..

À chaque instant le nombre de trames présentes simultanément dans une file d'attente est égal à la différence de ces deux comptages.

Le scénario d'arrivée pire cas maximisant le nombre de trames entrant dans un nœud est détaillé paragraphe 8.3.2. La stratégie qui minimise le nombre de trames transmises par un nœud est définie paragraphe 8.3.3. Pour finir, l'algorithme de BDF, pour le calcul du nombre de trames pire cas présent dans une file d'attente, est résumé paragraphe 8.3.4.

8.3.2 Maximisation des trames entrantes

Afin de maximiser les trames entrantes sur un nœud, nous utilisons, paragraphe 8.3.2.1 les fonctions RBF. Ces résultats sont affinés avec la prise en compte de la sérialisation (cf. paragraphe 8.3.2.2).

8.3.2.1 En fonction des caractéristiques des flux (RBF)

À partir des fonctions RBF (*Request Bound Functions*), il est possible de déterminer pour chaque flux $v_i \in \Gamma^h$, le nombre pire cas de trame entrant sur un nœud h durant n'importe quel intervalle de longueur t (avec $t \geq 0$). Les fonctions RBF ont été introduites dans le contexte FIFO [Ben+17b]. Elles sont généralisables à n'importe quelle politique de service non-préemptive. En effet, à partir du contrat de trafic d'un flux v_i , le nombre maximum de trames générées par v_i par son nœud source ($first_i$) durant n'importe quel intervalle fermé $[t_0, t_1]$ est égal à $\left(1 + \left\lfloor \frac{t_1 - t_0}{T_i} \right\rfloor\right) C_i^{first_i}$.

Sur les nœuds suivants $first_i$ sur le chemin \mathcal{P}_i de v_i , les trames atteignant h durant l'intervalle $[t_0, t_1]$ ont été générées pendant l'intervalle étendu $[t_0 - Smax_i^h, t_1 - Smin_i^h]$ par $first_i$. Ainsi, la quantité de données générée par le flux v_i et entrant sur h durant l'intervalle $[0, t]$ est égale à :

$$rbf_i^h(t) = \left(1 + \left\lfloor \frac{t + J_i^h}{T_i} \right\rfloor\right) C_i^h$$

Par construction, les fonctions $rbf_i^h(t)$ maximisent l'arrivée des trames, en fournissant également, la date d'arrivée au plus tôt de chaque trame sur le nœud h [Ben+17b]. Comme il a été démontré paragraphe 5.4.2, en considérant les gignes maximales dues aux nœuds précédents, à l'instant 0, $1 + k_i^h$ trames générées par le flux v_i peuvent arriver en h :

$$rbf_i^h(0) = \left(1 + \left\lfloor \frac{J_i^h}{T_i} \right\rfloor\right) C_i^h = (1 + k_i^h) C_i^h$$

avec $k_i^h = \left\lfloor \frac{J_i^h}{T_i} \right\rfloor$ (cf. formule (5.5)).

La prochaine trame arrive à l'instant $t = (1 + k_i^h)T_i - J_i^h$ et les suivantes arrivent périodiquement toutes les T_i unités de temps. Ce scénario maximise dès l'instant 0 (cf. Fig. 8.4) le nombre de trames générées par le flux $v_i \in \Gamma^h$ arrivant sur h .

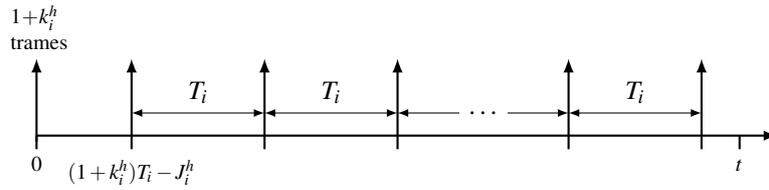


FIGURE 8.4 – Arrivée au plus tôt des trames d'un flux $v_i \in \Gamma^h$ dans une file d'attente conservatrice h .

8.3.2.2 En fonction des caractéristiques des liens (sérialisation)

Dans un réseau de paquets, les trames qui partagent un même lien physique sont sérialisées, elles ne peuvent jamais arriver simultanément dans le nœud suivant. Ainsi, l'arrivée d'une nouvelle trame est distante d'au moins la durée de transmission de la précédente. Cette contrainte liée au lien physique, appelée effet de sérialisation, n'est pas prise en compte dans les fonctions RBF du paragraphe 8.3.2.1 impliquant une surestimation des trames entrant dans un nœud h .

Les flux $v_i \in \Gamma_0^h$ ne sont, par définition, pas sérialisés dans le nœud h car ils sont générés directement par h . Seuls les flux qui arrivent depuis le nœud précédent sur h_x via le lien entrant IP_x^h avec $1 \leq x \leq n$ (cf. Fig. 8.2) sont sérialisés. Il n'y a aucune garantie que deux trames qui sont prêtes dans le nœud h_x soient toujours transmises dans le même ordre. Ainsi, l'analyse pire cas nécessite d'explorer tous les scénarios possibles menant à une explosion combinatoire. Pour résoudre ce problème, nous définissons une solution basée sur les notions de période d'activité et d'instant oisif (cf. Définitions 5.7 et 5.8) adaptées au lien IP_x^h .

Définition 8.1

Dans un nœud h , un x -instant oisif (avec $1 \leq x \leq n$) est un instant oisif (cf. définition 5.7) pour les trames entrant par le lien entrant IP_x^h et une x -période d'activité (avec $1 \leq x \leq n$), la période d'activité (cf. définition 5.8) entre deux x -instants oisifs consécutifs.

Nous définissons, maintenant, la trame à envoyer par le nœud h_x vers h (via IP_x^h) durant une x -période d'activité, afin de maximiser le nombre de trames entrant sur h . Nous considérons dans un premier temps, le x -instant oisif de début de la période d'activité (lemme 8.1) puis le reste de la période d'activité (lemme 8.2).

Lemme 8.1

Soient h un nœud et IP_x^h un lien entrant dans le nœud h depuis h_x . À chaque x -instant oisif, l'application de l'algorithme Largest Processing Time First (LPT) [Gra69] qui choisit la trame avec le plus long temps de transmission pour être transmise vers h , maximise le nombre de trames entrant dans h .

Démonstration :

La démonstration se fait par un contre-exemple [Gar+22].

Le scénario résultant peut mener à une sur-estimation de nombre de trames entrant sur h . Ce

scénario pessimiste peut être utilisé pour l'évaluation pire cas.

□

Nous prouvons (cf. lemme 8.2) maintenant que l'algorithme *Shortest Processing Time First* (SPT) [Gra69] maximise le nombre de trames qui atteignent h via le lien sérialisé IP_x^h depuis h_x durant n'importe quelle x -période d'activité $]t_1, t_2]$ ($t_1 < t_2$) où l' x -instant oisif t_1 a déjà été analysé dans le lemme 8.1.

Lemme 8.2

La politique de service SPT est optimale pour maximiser le nombre de trames transmises par un nœud durant n'importe quelle période d'activité $]t_1, t_2]$ ($t_1 < t_2$) parmi toutes les politiques de service non-préemptive et conservative.

Démonstration : La politique de service SPT maximise le plus le nombre de trames transmises par un nœud durant n'importe quelle période d'activité $]t_1, t_2]$ ($t_1 < t_2$) parmi toutes les politiques de service non-préemptives et conservatives.

Ce lemme a été démontré dans [Gar+22] avec un argument d'interchange similaire à celui utilisé par Dertouzos [Der74] pour démontrer l'optimalité d'*Earliest Deadline First*, pour définir un ordonnancement faisable sur un système monoprocesseur.

□

Nous définissons maintenant l'algorithme SPT^* qui est une variation de SPT et qui maximise dès l'instant 0, le nombre de trames arrivant sur un nœud h .

Théorème 8.1

Soit h un nœud avec une politique de service conservative et non-préemptive. Le nombre de trames atteignant le nœud h via le lien IP_x^h (avec $1 \leq x \leq n$) durant l'intervalle $[0, t]$ (with $t > 0$), est maximisé si parmi les trames disponibles selon les fonctions RBF (cf. paragraphe 8.3.2.1) l'algorithme LPT est appliqué au début de l'intervalle (à l'instant 0) et à chaque x -instant oisif et SPT pour le reste du temps. Cette stratégie est nommée SPT^ .*

Proof : Directement des lemmes 8.1 and 8.2.

□

Nous analysons maintenant les trames transmises par le nœud h .

8.3.3 Minimisation des trames transmises par un nœud h

Définir pour chaque politique de service, le scénario exact de départ des trames qui minimise le nombre de trames transmises par un nœud est un problème avec une grande complexité combinatoire. Nous définissons dans la suite, une borne inférieure pire cas, du nombre de trames transmises par un nœud, garantie pour n'importe quelle politique de service. Cette borne inférieure garantie est obtenue

en appliquant l'algorithme *Largest Processing Time First* (LPT) [Gra69] à chaque instant de décision et donc, en sélectionnant la trame avec la plus longue durée de transmission parmi celles en attente dans le nœud. Nous démontrons que LPT est optimal pour ce problème (cf. théorème 8.2).

Théorème 8.2

Considérant un ensemble de trames présentes dans la file d'attente d'un nœud h à des instants arbitraires, la politique de service LPT est optimale parmi toutes les politiques de service conservatives et non-préemptives pour minimiser le nombre de trames transmises par h durant n'importe quel intervalle $[0, t]$ (avec $t \geq 0$).

Démonstration : La preuve est similaire à celle du lemme 8.2 en utilisant le même argument d'échange et qui a été initialement utilisé par Dertouzos [Der74].

□

Nous détaillons, dans la suite, le calcul de BDF afin de déterminer une borne supérieure du nombre de trames qui peuvent être présentes simultanément dans la file d'attente d'un nœud.

8.3.4 Calcul du dimensionnement pire cas en nombre de trames par BDF

Les fonctions RBF, présentées paragraphe 8.3.2.1, définissent pour n'importe quelle politique de service conservative et non-préemptive, pour chaque nœud et pour tout instant t , une borne supérieure du nombre de trames en attente dans la file d'attente du nœud. Les fonctions RBF sont utilisées par l'approche BDF afin de déterminer NBF^h , le pire dimensionnement en nombre de trames de la file d'attente de chaque nœud h . Ainsi, les fonctions $rbf_i^h(t)$ sont utilisées :

- pour chaque lien entrant IP_x^h (avec $1 \leq x \leq n$) pour maximiser le nombre de trames atteignant h selon l'algorithme SPT^* du théorème 8.1 ;
- afin de maximiser le nombre de trames générées par les flux $v_i \in \Gamma_0^h$ en utilisant SPT (cf. paragraphe 8.3.2.1) ;
- pour l'ensemble des flux $v_i \in \Gamma^h$ (et indépendamment de ce qui a été fait précédemment) afin de minimiser les trames transmises par h selon LPT, théorème 8.2.

Nous présentons l'algorithme 3 de l'approche BDF permettant de déterminer le pire dimensionnement d'une file d'attente d'un nœud h . Cet algorithme est basé sur SPT^* pour comptabiliser les trames entrantes et LPT pour celles transmises par h .

L'algorithme 3 prend en entrée l'ensemble de flux Γ^h qui traverse le nœud h , et pour chaque flux v_i , sa durée de transmission C_i^h , sa période sporadique initiale T_i et la gigue pire cas accumulée tout au long du chemin de v_i , depuis son nœud source jusqu'à son arrivée sur h . Les paramètres C_i^h et T_i peuvent être dérivés directement du contrat de trafic du flux. La gigue J_i^h peut être calculée par n'importe quelle méthode d'analyse des délais de bout-en-bout pire cas telle que NC ou FA.

L'algorithme 3 initialise le nombre de trames arrivées sur h , le nombre de trames transmises par h et NBF^h à zéro (ligne 2). L'algorithme de BDF définit un ensemble de fonctions génératrices (*func-*

Input: Γ^h , l'ensemble de flux traversant le nœud h

$$\forall v_i \in \Gamma^h : C_i^h, T_i, J_i^h$$

Result: NBF^h

```

1 begin
2   (Arrivées, Transmises,  $NBF^h$ )  $\leftarrow$  (0, 0, 0)
3   foreach  $v_i \in \Gamma^h$  do
4      $\lfloor$   $rbf_i^h(t) \leftarrow \text{groupBy}([( \max(0, kT_i - J_i^h), C_i^h ) | k > 0])$ 
5      $RBF^h \leftarrow \text{merge}(rbf_i^h(t), \forall v_i \in \Gamma^h)$ 
6      $RBF_x^h \leftarrow \text{merge}(rbf_i^h(t), \forall v_i \in \Gamma_x^h)$ 
7      $in^h \leftarrow \text{merge}(RBF_0^h, SPT^*(RBF_{x \neq 0}^h))$ 
8      $out^h \leftarrow LPT(RBF^h)$ 
9     foreach  $(t, CList) \in \text{merge}(in^h, out^h)$  do
10      foreach  $C \in CList$  do
11        if  $C \in in^h$  then
12          Arrivées  $\leftarrow$  Arrivées + 1
13        else if  $C \in out^h$  then
14          Transmises  $\leftarrow$  Transmises + 1
15       $NBF^h \leftarrow \max(NBF^h, \text{Arrivées} - \text{Transmises})$ 
16      if  $\text{Arrivées} - \text{Transmises} = 0$  then return  $NBF^h$ 

```

Algorithm 3: Calcul de NBF^h par BDF.

tional streams) $rbf_i^h(t)$ pour chaque flux $v_i \in \Gamma^h$ (ligne 4). Nous définissons un flux infini de couple (t, C_i^h) ordonné par ordre croissant sur t , où t désigne l'instant d'arrivée dans le nœud h d'une trame f_i de durée de transmission C_i^h . Les trames avec le même instant d'arrivée t sur h sont agrégées dans une même liste via `groupBy`. Ainsi, ce générateur définit des couples $(t, [C_i^h, \dots])$ par ordre croissant sur les t .

Ces flux sont combinés pour former un nouveau flux appelé RBF^h (l. 5) avec la fonction *merge* qui préserve l'ordre croissant sur t , tout en fusionnant les listes de C_i^h avec un même instant d'arrivée t . De même, nous construisons les flux RBF_x^h pour chaque lien entrant IP_x^h de h (l. 6).

Ces différents flux (RBF^h et RBF_x^h) permettent de comptabiliser le nombre de trames qui théoriquement arrivent sur h à chaque instant t en fonction des contrats de trafic. Ces fonctions peuvent potentiellement mener à des arrivées massives de trames via un lien entrant IP_x^h avec $1 \leq x \leq n$. Ainsi, les arrivées potentielles de trames sont régulées (voir lignes 7-8) via l'algorithme *SPT**. De même pour les trames émises par h qui sont régulées par LPT. Les seules trames qui ne sont pas régulées, sont celles qui ont pour nœud source h (Γ_0^h) représentées par le flux RBF_0^h .

La fin de l'algorithme parcourt les instants t par ordre croissant depuis 0. Pour chaque nouvelle arrivée (*resp.* transmission) de trame, le nombre de trames arrivées (*resp.* transmises) est incrémenté (l. 9-14). Dès que la file d'attente est vide, le maximum de $\text{Arrivées} - \text{Transmises}$ est retourné (l. 15-16). En effet, le nombre de trames présentes dans le nœud h est maximisé dès l'instant 0. Le pire dimensionnement ne pourra pas être déterminé après que la file soit de nouveau vide. La validité de cette condition d'arrêt est démontrée théorème 8.3.

Théorème 8.3

Soit h un nœud utilisant une politique de service conservatrice et non-préemptive. Dès lors que la charge locale $U^h = \sum_{v_i \in \Gamma^h} \frac{C_i^h}{T_i}$ est inférieure ou égale à 1, la condition d'arrêt de l'algorithme 3 est nécessairement atteinte.

Démonstration : Ce théorème a été démontré dans [Gar+22]. □

8.4 Expérimentations

Dans ce paragraphe, nous appliquons l'approche BDF sur une configuration industrielle et nous comparons les résultats obtenus avec BDF, avec la seule approche de dimensionnement de files d'attente, pire cas en nombre de trames, disponible pour des configurations avec des tailles de trames différentes, l'approche naïve (*cf.* paragraphe 8.1).

La configuration industrielle est basée sur celle de l'AFDX présentée paragraphe 5.5.3, déployée dans l'A380 [Ben+17b]. Elle est composée huit commutateurs (avec 16 ports de sortie) et de 96 *End Systems* (ES), soit 288 nœuds. Les ES s'échangent des données (trames) générées par 983 flux. Sur cette configuration, nous appliquons les approches BDF et naïve à partir des délais pire cas calculés par notre approche FA.

Pour cette configuration industrielle, les temps de calculs nécessaires à une version de BDF, implémentée dans un unique processus Python sur un processeur Intel Core i5-10500 de 3.1 Ghz et est de 117 ms avec une utilisation mémoire de 27 MB.

Une présentation statistique des résultats est résumée Fig. 8.5. Ce graphique regroupe pour chaque intervalle de 5%, le nombre de nœuds pour lesquels BDF améliore le dimensionnement pire cas de la méthode naïve. Par exemple, BDF réduit de 40 – 45% la taille de la file d'attente obtenue par la méthode naïve. Cette amélioration est, en moyenne, de 76.3%. BDF détermine toujours des bornes plus serrées que celles calculées avec la méthode naïve. Pour seulement deux nœuds, les dimensionnements pire cas de la file d'attente calculés avec BDF et la méthode naïve sont égaux. Ces deux nœuds sont peu chargés et sont tels que toutes les trames ont la même taille.

La Fig. 8.6 représente pour chacun des huit commutateurs (SW), la mémoire totale allouée pour l'ensemble de ses files d'attente. En moyenne, avec BDF, la mémoire totale allouée est divisée par 3.16, comparée à la méthode naïve. La mémoire ainsi économisée est de 6.36 Mb. Cette réduction de la mémoire utilisée peut améliorer les performances du système. Par exemple, les trames pourraient être stockées dans la mémoire cache du processeur au lieu de l'être dans la mémoire principale.

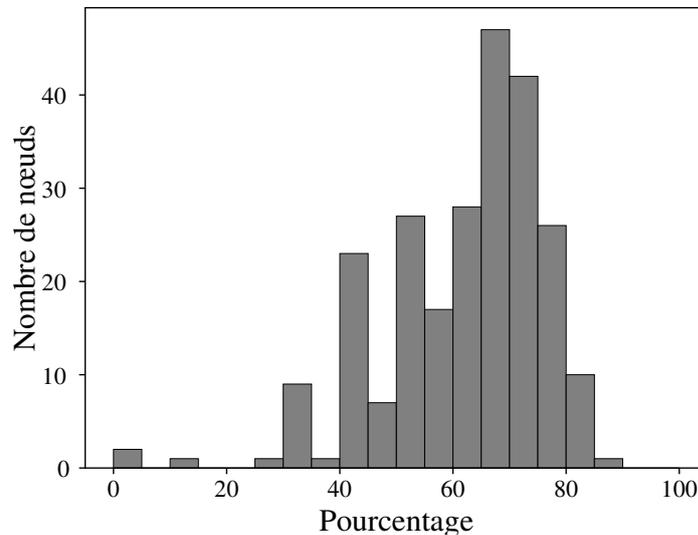


FIGURE 8.5 – Amélioration du dimensionnement des files d’attente par BDF en comparaison avec la méthode naïve sur une configuration industrielle.

8.5 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche, BDF, pour déterminer le dimensionnement pire cas d’une file d’attente. Cette approche est applicable dans le contexte des systèmes embarqués critiques pour lesquels l’allocation dynamique de mémoire n’est pas permise et lorsque la certification est requise. BDF détermine une borne supérieure garantie pire cas des files d’attente en nombre de trames. BDF s’applique à tout type de réseau de paquets utilisant une politique de service conservative et non-préemptive. Pour son calcul de dimensionnement pire cas, BDF requière : la topologie du réseau, les contrats de trafic des flux à l’entrée du réseau et une méthode d’analyse des délais de bout en bout pire cas telle que NC ou FA.

BDF n’est pas contrainte comme les autres approches de la littérature par une taille constante pour toutes les trames du réseau. Nous l’avons appliqué sur une configuration réseau industriel et, ainsi démontré, son passage à l’échelle. Nous avons pu démontrer un bénéfice significatif par rapport à une approche naïve.

Ces travaux ont été initiés par Nassima durant sa thèse [Ben18] avec notamment un workshop [Ben+16] et une présentation. Ces travaux ont été finalisés durant la thèse de Richard [Gar+22].

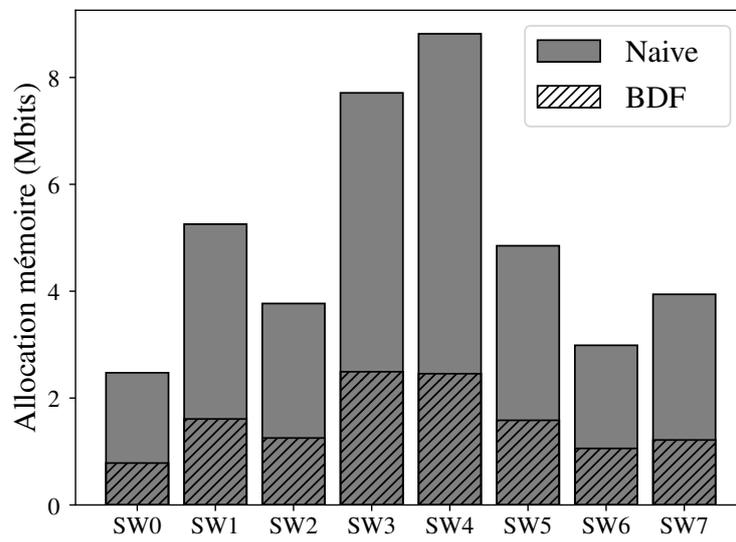


FIGURE 8.6 – Comparaison de la mémoire allouée pour chaque commutateur (SW) entre BDF et la méthode naïve.

Quatrième partie

Bilan et perspectives

Bilan des travaux

Depuis l'obtention de mon doctorat en 2006, j'ai encadré cinq thèses dont trois sont en cours. J'ai également encadré trois stagiaires de M1 (ou équivalent) et quatre M2 (ou équivalent). Mes travaux ont donné lieu à la publication de cinq chapitres d'ouvrage, huit revues internationales et 23 conférences internationales. J'ai participé à trois projets (nationaux et internationaux).

Ces travaux ont été menés dans deux laboratoires, l'équipe *Ingénierie Réseaux et Télécommunications* de l'*Institut en Recherche Informatique de Toulouse* de 2006 à 2009 puis le *Laboratoire d'Informatique et d'Automatique pour les Systèmes* et son équipe *Systèmes Embarqués Temps Réel* depuis 2009.

Mes travaux de recherches se sont focalisés principalement sur l'analyse de performance dans les réseaux temps réel embarqués dans les domaines initialement de l'avionique puis de l'automobile et de l'industrie 4.0. Mes travaux consistent à la démonstration du déterminisme des réseaux embarqués pour deux points :

- Délai de bout-en-bout de traversée du réseau par les trames ;
- Dimensionnement des files d'attente.

9.1 Analyse du délai de bout-en-bout de traversée du réseau

La méthode *Forward end-to-end delay Analysis* (FA) a été définie, initialement, dans le contexte du réseau AFDX avec la politique de service FIFO. FA se base sur des résultats d'ordonnancement temps réel monoprocesseur. En effet, la transmission d'une trame par un port de sortie est similaire à l'exécution d'une tâche non-préemptive sur un système monoprocesseur. FA a été développée avec un modèle de réseau proche de l'AFDX. Son modèle a été généralisé afin de pouvoir appliquer FA à d'autres réseaux de paquets. L'ensemble des formules de FA a été démontré formellement. Nous avons comparé FA avec les deux approches concurrentes, le *Network Calculus* et la méthode des trajectoires. Nous avons démontré que FA est applicable sur des configurations industrielles. Nous avons démontré qu'il n'y a pas de domination entre FA, NC et TA. Cependant, FA détermine globalement des délais de bout en bout inférieurs à ceux obtenus par NC et TA.

Pour répondre aux besoins des industriels d'introduire de la différenciation de service dans les commutateurs, FA a été étendue à un modèle similaire à celui défini en FIFO mais avec la politique de service FP/FIFO. Le passage à l'échelle industrielle a été démontré sur une configuration issue de l'A380.

Après le domaine de l’avionique, nous avons travaillé pour étendre FA à un réseau embarqué de l’automobile, AVB. Cette extension a demandé une évolution du modèle réseau afin d’intégrer l’algorithme de gestion de crédit CBS. Les formules ont été démontrées formellement.

9.2 Dimensionnement des files d’attente

Dans la littérature, il n’existe pas à notre connaissance de méthode définie spécifiquement pour le dimensionnement pire cas par trame. Après avoir démontré que ce problème (allocation statique) est différent de celui en nombre de bits (allocation dynamique), nous avons défini l’approche *Buffer Dimensioning per Frame* (BDF). Cette approche peut s’appliquer pour tout réseau de paquets utilisant n’importe quelle politique d’ordonnement conservatrice et non-préemptive. Ainsi, pour être appliquée, l’approche BDF a besoin de :

- la topologie du réseau ;
- un contrat de trafic à l’entrée du réseau pour chaque flux ;
- une méthode de calcul des délais de bout-en-bout des trames.

Nous avons appliqué BDF sur une configuration réseau industriel et, ainsi, démontré, son passage à l’échelle. Nous avons pu démontrer une amélioration significative des dimensionnements en nombre de trames des files d’attente en comparaison avec une approche naïve qui divise le dimensionnement pire cas en bits par la taille de la trame la plus petite.

9.3 Bilan

Pour situer nos contributions, nous reprenons le tableau 4.1 et le complétons avec nos deux approches FA et BDF.

	Preuve formelle	Modèle généraliste	AFDX FIFO	AFDX FP/FIFO	AVB	File d’attente bits	trames
Méthode Holistique	-	✓	-	-	-	-	-
Network Calculus (NC)	✓	✓	✓	✓	✓	✓	~
Méthode des Trajectoires (TA)	-	✓	✓	✓	✓	✓	~
<i>Computational Performance Analysis</i> (CPA)	✓	✓	-	-	✓	-	-
Travaux de Bordoloi <i>et al.</i>	✓	-	-	-	✓	-	-
La méthode des intervalles éligibles	✓	-	-	-	✓	-	-
<i>Forward end-to-end Analysis</i> (FA)	✓	✓	✓	✓	✓	✓	-
<i>Buffer Dimensioning per Frame</i> (BDF)	✓	✓	-	-	-	-	✓

TABLE 9.1 – Contributions à l’analyse de performances dans les réseaux AFDX et AVB.

Ainsi, tableau 9.1, FA et BDF ont été démontrées formellement et proposent chacune un modèle réseau. FA détermine les délais de bout-en-bout dans des réseaux de paquets avec différentes politiques de service (FA, FP/FIFO et CBS). Par définition, FA permet le calcul de la pire occupation

d'une file d'attente en nombre de bits (problème connexe aux délais pire cas). Pour le calcul du pire dimensionnement en nombre de trames, nous avons défini une approche spécifique BDF, qui n'est pas limitée par une taille de trames commune pour toutes les trames.

Perspectives

10.1 *Time-Sensitive Networking* (TSN)

TSN est développé par un groupe de travail 802.1 de l'*Institute of Electrical and Electronics Engineers* (IEEE). TSN se compose d'un ensemble de standards développés pour normaliser les caractéristiques temps réel, regroupées en quatre catégories : synchronisation, fiabilité, latences garanties et gestion de ressources. TSN est une évolution de l'AVB, il se base ainsi sur les politiques de service FP/FIFO et CBS. Cependant, TSN ajoute en plus, un mécanisme de fenêtres de transmission appelé *Time Aware Shaping* (TAS) [Qbv] contrôlé par ouverture/fermeture de portes. TAS requière une exigence forte en synchronisation. TAS supprime (même pour les flux les plus prioritaires) le coût de la non préemption d'une trame moins prioritaire en cours de transmission, contrairement à FP/FIFO et CBS. Le TAS, utilisé pour des flux planifiés (*p.ex.* en fonction des temps de production des données) dans un réseau entièrement synchronisé, minimise les latences.

L'extension de FA au réseau TSN est complexe car TAS est une politique de service non conservative par nature mais c'est aussi une étape indispensable. En effet, TSN est conçu pour plusieurs domaines, en particulier pour l'industrie 4.0. La difficulté du TSN réside aussi bien dans la diversité des différentes normes et la grande flexibilité qui est souhaitée par ses concepteurs. Ainsi, l'extension de FA se fera forcément dans un premier temps, en partant de sa version CBS mais aussi en procédant avec un modèle incrémental.

10.2 Ordonnancement hiérarchisé et chaînes fonctionnelles

Les calculateurs utilisés par Airbus dans ses avions sont standardisés par la norme ARINC 653 [GZP16]. Dans un calculateur, l'ensemble des tâches (applications) est réparti dans les partitions selon leurs niveaux de criticité. Le calculateur garantit à chaque partition un accès aux ressources. Ainsi, le temps processeur est composé d'un cycle qui est répété à l'infini. Ce cycle résulte d'un ordonnancement hors-ligne entre les partitions. Lorsqu'une partition accède au processeur, ses applications sont exécutées selon un ordonnancement local. Ainsi lorsqu'une tâche se réveille, son ordonnancement par le calculateur est à deux niveaux (ordonnancement de la tâche à l'intérieur de la partition et de la partition sur le calculateur).

Il n'existe dans la littérature que peu de travaux sur l'ordonnancement temps réel avec les spécificités de l'ARINC 653. Ce problème a été peu analysé dans la littérature. Ainsi dans le cadre du

projet CORAC PANDA (*cf.* paragraphe 1.4.4.1), nous avons abordé cette problématique mais avec des temps de réponse surévalués. *Rate Monotonic* (RM) est un bon point de départ pour l'ordonnancement local à chaque partition. En effet, l'aéronautique est un domaine pour lequel l'ordonnancement est, généralement, statique. Mais une des difficultés de ce problème est liée aux paramètres temporels des tâches (temps d'exécution, période, échéance, *etc.*) qui sont exprimés comme des contraintes globales sur le calculateur. Cependant, pour analyser le pire temps de réponse d'une tâche, il faut transformer ses paramètres en données locales à l'intérieur de la partition. Le pire temps de réponse (en fonction des autres tâches) est alors déterminé localement avant de les calculer globalement sur le calculateur. L'objectif est d'analyser finement les deux niveaux d'ordonnancement afin de déterminer des temps de réponse avec un minimum de pessimisme.

Ce travail pourra ensuite être intégré au processus de validation des délais pire-cas de traversée du réseau (avec FA) par les trames. L'objectif est d'analyser le comportement temporel des chaînes fonctionnelles. Une chaîne fonctionnelle se définit par une tâche qui s'exécute sur un calculateur, la tâche envoie le résultat de son travail, via une trame, vers une tâche d'un autre calculateur. Le temps réponse temporel de cette chaîne est le délai de bout-en-bout depuis le réveil de la première tâche jusqu'à la fin d'exécution de la seconde.

10.3 Ordonnancement temps réel

L'ordonnancement temps réel constitue ma thématique développée, initialement, durant ma thèse et qui est toujours la thématique centrale de l'équipe *Systèmes Embarqués Temps Réel* (SETR). En parallèle de mes travaux sur les réseaux embarqués, je vais reprendre mes recherches sur l'ordonnancement, par exemple sur des plateformes hétérogènes. Je n'ai pas de sujet en particulier et cette perspective est à plus long terme. En effet, je n'ai plus contribué sur cette thématique depuis plus de 17 ans maintenant et j'ai un travail important de bibliographie à effectuer afin de remonter en compétences. Pourtant c'est un objectif important pour moi et pour l'équipe SETR. Les deux sujets (réseaux et ordonnancement) sont connexes et source de sujets croisés comme le démontre la définition de l'approche FA, basée sur l'ordonnancement temps réel, pour les réseaux de paquets.

Nos travaux sur les réseaux embarqués temps réel conjugués aux synergies de l'équipe SETR sur l'ordonnancement et l'ingénierie des modèles doivent permettre de développer les collaborations avec des partenaires privés et/ou publics. Ainsi comme pour le projet CORAC PANDA (*cf.* paragraphe 1.4.4.1), notre expertise pour la validation temporelle de systèmes distribués doit être valorisée. Ces travaux doivent aussi être l'occasion de développer des collaborations de recherche tant au niveau national qu'international.

Bibliographie

- [66402] ARINC 664. *ARINC 664, Aircraft Data Network, Parts 1,2 7. Technical report, ARINC specification 664*. 2002-2005 (cf. p. 37).
- [74994] No. 7498-1 :1994. *Information Technology — Open Systems Interconnection — Basic Reference Model : The Basic Model*. 1994 (cf. p. 44).
- [80206] 802.1AS. *802.1AS - Timing and Synchronization for Time-Sensitive Applications*. Rapp. tech. IEEE, 2006 (cf. p. 43).
- [80209] IEEE 802.1QAV. « IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12 : Forwarding and Queuing Enhancements for Time-Sensitive Streams ». In : *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)* (jan. 2009), p. C1-72 (cf. p. 43).
- [80211] 802.1BA. *IEEE Standard for Local and metropolitan area networks—Audio Video Bridging (AVB) Systems*. Rapp. tech. IEEE, 2011 (cf. p. 95).
- [80299] IEEE 802. *Time-Sensitive Networking (TSN) Task Group*. 1999. URL : <https://1.ieee802.org/tsn/> (visité le 03/09/2019) (cf. p. 43).
- [AD94] Rajeev ALUR et David L DILL. « A theory of timed automata ». In : *Theoretical computer science* 126.2 (1994), p. 183-235 (cf. p. 49).
- [Adn+10] Muhammad ADNAN et al. « Model for worst case delay analysis of an AFDX network using timed automata ». In : *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)* (2010), p. 1-4 (cf. p. 49).
- [Adn+12] Muhammad ADNAN et al. « An improved timed automata approach for computing exact worst-case delays of AFDX sporadic flows ». In : *Emerging Technologies and Factory Automation (ETFA), Krakow*. IEEE, sept. 2012, p. 1-8 (cf. p. 49).
- [Adn13] Muhammad ADNAN. « Exact worst-case communication delay analysis of AFDX network ». Thèse de doct. Institut National Polytechnique de Toulouse, nov. 2013 (cf. p. 49).
- [Ald+12] Giuliana ALDERISI et al. « Simulative assessments of IEEE 802.1 Ethernet AVB and Time-Triggered Ethernet for Advanced Driver Assistance Systems and in-car infotainment ». In : *2012 IEEE Vehicular Networking Conference, VNC 2012, Seoul, Korea (South), November 14-16, 2012*. 2012, p. 187-194 (cf. p. 43, 44, 105).
- [Alu91] Rajeev ALUR. « Techniques for automatic verification of real-time systems ». Thèse de doct. stanford university, 1991 (cf. p. 49).
- [Ana+06] Madhukar ANAND et al. « Formal Modeling and Analysis of the AFDX Frame Management Design. » In : avr. 2006, p. 393-399 (cf. p. 49).
- [ARI97] ARINC. *ARINC 651, Aeronautical Radio Inc. ARINC specification 651. Design Guidance for Integrated Modular Avionics*. Nov. 1997 (cf. p. 37).

- [ASF11] Muhammad ADNAN, Jean-Luc SCHARBARG et Christian FRABOUL. « Minimizing the search space for computing exact worst-case delays of AFDX periodic flows (regular paper) ». In : *IEEE International Symposium on Industrial Embedded Systems (SIES), Västerås, 15/06/2011-17/06/2011*. IEEE, juin 2011, p. 294-301 (cf. p. 49).
- [Ash+17] Mohammad ASHJAEI et al. « Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support ». In : *Real-Time Systems* 53.7 (juill. 2017), p. 526-577 (cf. p. 49, 54).
- [Aut] *What Engineers Need to Know When Selecting an Automotive-Qualified MCU for Vehicle Applications*. <https://www.digikey.dk/en/articles/techzone/2014/jul/what-engineers-need-to-know-when-selecting-an-automotive-qualified-mcu-for-vehicle-applications>. Accessed : 2019-03-22 (cf. p. 39).
- [Axe+14] P. AXER et al. « Exploiting shaper context to improve performance bounds of Ethernet AVB networks ». In : *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*. 2014, p. 1-6 (cf. p. 54).
- [Bac+92] F. BACCELLI et al. *Synchronization and linearity : an algebra for discrete event systems*. John Wiley & Sons Ltd. John Wiley & Sons Ltd, 1992 (cf. p. 51).
- [Bas+10] Ananda BASU et al. « Verification of an AFDX Infrastructure Using Simulations and Probabilities ». In : *Runtime Verification*. Sous la dir. d'Howard BARRINGER et al. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, p. 330-344 (cf. p. 49).
- [Bau11] Henri BAUER. « Analyse pire cas de flux hétérogènes dans un réseau embarqué avion ». Thèse de doct. Institut National Polytechnique de Toulouse, 2011 (cf. p. 42, 76).
- [BBLC18] Anne BOUILLARD, Marc BOYER et Euriell LE CORRONC. *Deterministic Network Calculus : From Theory to Practical Implementation*. Oct. 2018 (cf. p. 50).
- [BD12] Marc BOYER et David DOOSE. « Combining Network Calculus and Scheduling Theory to Improve Delay Bounds ». In : *Proceedings of the 20th International Conference on Real-Time and Network Systems*. Association for Computing Machinery, 2012, p. 51-60 (cf. p. 52).
- [Ben+16] Nassima BENAMMAR et al. « Tighter buffer dimensioning in AFDX networks ». In : *SIGBED Review* 13.4 (2016), p. 37-42 (cf. p. 122).
- [Ben+17a] Nassima BENAMMAR et al. « Forward end-to-end delay analysis extension for FP/FIFO policy in AFDX networks ». In : *22nd IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2017, Limassol, Cyprus, September 12-15, 2017*. 2017, p. 1-8 (cf. p. 92).
- [Ben+17b] Nassima BENAMMAR et al. « Forward End-to-End Delay for AFDX Networks ». In : *IEEE Transactions on Industrial Informatics* 2017 (2017), p. 858-865 (cf. p. 63, 68-70, 72, 78, 84, 85, 87, 89, 116, 121).
- [Ben18] Nassima BENAMMAR. « Modélisation, évaluation et validation des systèmes temps réel distribués ». Thèse de doct. Université de Poitiers, 2018 (cf. p. 68-70, 72, 84, 85, 87, 89, 98, 122).

- [Ben+18] Nassima BENAMMAR et al. « Timing analysis of AVB Ethernet network using the Forward end-to-end Delay Analysis ». In : *Proceedings of the 26th International Conference on Real-Time Networks and Systems, RTNS 2018, Chasseneuil-du-Poitou, France, October 10-12, 2018*. 2018, p. 223-233 (cf. p. 98, 100-102, 110).
- [BF08] M. BOYER et C. FRABOUL. « Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus ». In : *2008 IEEE International Workshop on Factory Communication Systems*. 2008, p. 11-20 (cf. p. 52).
- [BFG11] Anne BOUILLARD, Nadir FARHI et Bruno GAUJAL. « Packetization and Aggregate Scheduling ». In : *Research Report - INRIA* (juill. 2011) (cf. p. 52, 57).
- [BJ11] Anne BOUILLARD et Aurore JUNIER. « Worst-Case Delay Bounds with Fixed Priorities Using Network Calculus ». In : *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*. VALUETOOLS '11. ICST (Institute for Computer Sciences, Social-Informatics et Telecommunications Engineering), 2011, p. 381-390 (cf. p. 52).
- [BMN11] Marc BOYER, Jörn MIGGE et Nicolas NAVET. « An Efficient and Simple Class of Functions to Model Arrival Curve of Packetised Flows ». In : *Proceedings of the 1st International Workshop on Worst-Case Traversal Time*. Association for Computing Machinery, 2011, p. 43-50 (cf. p. 52).
- [Bor+14] Unmesh D. BORDOLOI et al. « Schedulability analysis of Ethernet AVB switches ». In : *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications, Chongqing, China, August 20-22, 2014*. 2014, p. 1-10 (cf. p. 54, 99-101).
- [Bou+08] Anne BOUILLARD et al. « Optimal routing for end-to-end guarantees using Network Calculus ». In : *Performance Evaluation* 65 (nov. 2008) (cf. p. 52).
- [Bou+09] Anne BOUILLARD et al. « COINC Library : A toolbox for Network Calculus ». In : *Proceedings of the 4th International Conference on Performance Evaluation Methodologies and Tools, ValueTools 9* (jan. 2009) (cf. p. 51).
- [Boy10] Marc BOYER. « Half-modeling of shaping in FIFO net with network calculus ». In : (nov. 2010) (cf. p. 52).
- [Boy+10] Marc BOYER et al. « The PEGASE project : precise and scalable temporal analysis for aerospace communication systems with network calculus ». In : *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*. Springer. 2010, p. 122-136 (cf. p. 51).
- [BS13] Michael A. BECK et Jens SCHMITT. « The DISCO Stochastic Network Calculator Version 1.0 : When Waiting Comes to an End ». In : *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2013, Torino, Italy*. ICST (Institute for Computer Sciences, Social-Informatics et Telecommunications Engineering), 2013, p. 282-285 (cf. p. 51, 52).
- [BS15a] A. BOUILLARD et G. STEA. « Exact Worst-Case Delay in FIFO-Multiplexing Feed-Forward Networks ». In : *IEEE/ACM Transactions on Networking* 23.5 (2015), p. 1387-1400 (cf. p. 52).

- [BS15b] Anne BOUILLARD et Giovanni STEA. « Quantitative Assessments of Distributed Systems : Methodologies and Techniques ». In : avr. 2015. Chap. Worst-Case Analysis of Tandem Queueing Systems Using Network Calculus, p. 129-173 (cf. p. 52).
- [BSF09a] Henri BAUER, Jean-Luc SCHARBARG et Christian FRABOUL. « Applying and optimizing Trajectory approach for performance evaluation of AFDX avionics network (regular paper) ». In : *Emerging Technologies and Factory Automation (ETFA), Palma de Mallorca*. IEEE, sept. 2009, p. 1-8 (cf. p. 53).
- [BSF09b] Henri BAUER, Jean-Luc SCHARBARG et Christian FRABOUL. « Applying trajectory approach to AFDX avionics network (short paper) ». In : *Euromicro Conference on Real-Time Systems (session Work in Progress) (ECRTS09), Dublin*. IEEE, juill. 2009, p. 57-60 (cf. p. 53).
- [BSF10] Henri BAUER, Jean-Luc SCHARBARG et Christian FRABOUL. « Improving the worst-case delay analysis of an AFDX network using and optimized trajectory approach ». In : *IEEE Transactions on Industrial Informatics* 6.4 (nov. 2010), p. 521-533 (cf. p. 53, 77).
- [BSF12] Henri BAUER, Jean-Luc SCHARBARG et Christian FRABOUL. « Applying trajectory approach with static priority queueing for improving the use of available AFDX resources ». In : *Real-Time Systems, Springer-Verlag* 48.1 (jan. 2012), p. 101-133 (cf. p. 53).
- [BT01] Jean-Yves Le BOUDEC et P. THIRAN. *Network calculus : A Theory of Deterministic Queueing Systems for the Internet*. T. 2050. ISBN : 3-540-42184-X. Springer Verlag, 2001 (cf. p. 50, 52, 54, 75).
- [Cao+16] J. CAO et al. « Tight worst-case response-time analysis for ethernet AVB using eligible intervals ». In : *2016 IEEE World Conference on Factory Communication Systems (WFCS)*. 2016, p. 1-8 (cf. p. 55).
- [Cao+16] Jingyue CAO et al. « Independent yet Tight WCRT Analysis for Individual Priority Classes in Ethernet AVB ». In : *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. RTNS '16. Brest, France : Association for Computing Machinery, 2016, p. 55-64 (cf. p. 55).
- [Cao+18] Jingyue CAO et al. « Independent WCRT Analysis for Individual Priority Classes in Ethernet AVB ». In : *Real-Time Syst.* 54.4 (oct. 2018), p. 861-911 (cf. p. 55).
- [CFS15] Rodrigo COELHO, Gerhard FOHLER et Jean-Luc SCHARBARG. « Dimensioning buffers for AFDX networks with multiple priorities virtual links ». In : *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th*. IEEE. 2015, 10A5-1 (cf. p. 53).
- [Cha00] Cheng-Shang CHANG. *Performance Guarantees in Communication Networks*. Berlin, Heidelberg : Springer-Verlag, 2000 (cf. p. 50, 52, 54).
- [Cha+06] Hussein CHARARA et al. « Methods for bounding end-to-end delays on an AFDX network ». In : *Real-Time Systems, 2006. 18th Euromicro Conference on*. IEEE. 2006, 10-pp (cf. p. 49, 52).
- [Cha07] Hussein CHARARA. « Évaluation des performances temps réel de réseaux embarqués avioniques ». Thèse de doct. Institut National Polytechnique de Toulouse, 2007 (cf. p. 49).

- [Com18] C/LM LAN/MAN Standards COMMITTEE. *IEEE 802.3-2018 - IEEE Standard for Ethernet*. Août 2018 (cf. p. 39).
- [Cru91a] Rene L. CRUZ. « A Calculus for NetWork Delay, Part I : Network Elements in Isolation ». In : *IEEE Transactions on Information Theory* 37.1 (jan. 1991), p. 114-131 (cf. p. 50).
- [Cru91b] Rene L. CRUZ. « A Calculus for NetWork Delay, Part II : Network Analysis ». In : *IEEE Transactions on Information Theory* 37.1 (jan. 1991), p. 132-141 (cf. p. 50).
- [DAB14] Joan Adrià Ruiz DE AZUA et Marc BOYER. « Complete Modelling of AVB in Network Calculus Framework ». In : *22Nd International Conference on Real-Time Networks and Systems*. RTNS '14. Versaille, France : ACM, 2014, p. 55-64 (cf. p. 53).
- [DAE12] Jonas DIEMER, Philip AXER et Rolf ERNST. « Compositional Performance Analysis in Python with pyCPA ». In : *In Proc. of WATERS*. 2012 (cf. p. 54).
- [DBZ18] Hugo DAIGMORTE, Marc BOYER et Luxi ZHAO. « Modelling in network calculus a TSN architecture mixing Time-Triggered, Credit Based Shaper and Best-Effort queues ». In : (2018) (cf. p. 53).
- [Der74] Michael Leonidas DERTOUZOS. « Control robotics :The procedural control of physical processors ». In : *Proceedings of the IFIP congress* (1974), p. 807-813 (cf. p. 118, 119).
- [Din+10] L. DING et al. « The Research of AFDX System Simulation Model ». In : *International Conference on Multimedia Technology* (2010), p. 1-4 (cf. p. 49).
- [DRE12] Jonas DIEMER, Jonas ROX et Rolf ERNST. « Modeling of Ethernet AVB Networks for Worst-Case Timing Analysis ». In : *IFAC Proceedings Volumes* 45.2 (2012). 7th Vienna International Conference on Mathematical Modelling, p. 848-853 (cf. p. 54, 105, 107).
- [DTE12] J. DIEMER, D. THIELE et R. ERNST. « Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching ». In : *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*. 2012, p. 1-10 (cf. p. 54).
- [Edm+18] Eds. EDMUND et al. *Handbook of Model Checking*. Springer. Springer, 2018 (cf. p. 49).
- [Emm+14] Grolleau EMMANUEL et al. *Systèmes temps réel embarqués : Conception et implémentation*. Dunod. Dunod, 2014, p. 592 (cf. p. 35).
- [FFG06] Fabrice FRANCES, Christian FRABOUL et Jérôme GRIEU. « Using network calculus to optimize the AFDX network ». In : *Embedded Real Time Systems (ERTS)* (2006) (cf. p. 52).
- [Gar+22] Richard GARREAU et al. *Buffer Dimensioning per Frames in packet networks*. Rapp. tech. LIAS/ISAE-ENSMA, 2022 (cf. p. 117, 118, 121, 122).
- [Gey+13] Fabien GEYER et al. « Evaluation of Audio/Video Bridging forwarding method in an avionics switched ethernet context ». In : *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE. 2013, p. 711-716 (cf. p. 49).
- [GPGH12] J. Javier GUTIÉRREZ, J. Carlos PALENCIA et Michael GONZÁLEZ HARBOUR. « Response time analysis in AFDX networks with sub-virtual links and prioritized switches ». In : *XV Jornadas de Tiempo Real* (fév. 2012). Santander (cf. p. 50).

- [Gra69] R. L. GRAHAM. « Bounds on Multiprocessing Timing Anomalies ». In : *SIAM Journal on Applied Mathematics* 17.2 (1969), p. 416-429 (cf. p. 117-119).
- [Gri04] Jérôme GRIEUX. « Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques ». Thèse de doct. Institut National Polytechnique de Toulouse, 2004 (cf. p. 40, 52, 54, 70, 75).
- [Gro10] IEEE 802.1 Working GROUP. « IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks Amendment 14 : Stream Reservation Protocol (SRP) ». In : *IEEE Std 802.1Qat-2010* (2010), p. 1-149 (cf. p. 105, 106).
- [Gro14] Emmanuel GROLLEAU. « Introduction to real-time scheduling ». In : Maryline Chetto. Wiley, 2014 (cf. p. 35).
- [GRS96] Laurent GEORGE, Nicolas RIVIERRE et Marco SPURI. « Preemptive and non-preemptive real-time uniprocessor scheduling ». Thèse de doct. Inria, 1996 (cf. p. 50).
- [GZP16] Jorge GARRIDO, Juan ZAMORANO et Juan A. de la PUENTE. « ARINC-653 Inter-Partition Communications and the Ravenscar Profile ». In : *Ada Lett.* 35.1 (2016), 38–45 (cf. p. 131).
- [Hea20] IEEE (Adam HEALEY). *IEEE 802.3 ETHERNET WORKING GROUP*. 2020. URL : <http://www.ieee802.org/3/> (visité le 20/04/2020) (cf. p. 54).
- [Hot+17] Oana HOTESCU et al. « Towards Quality of Service Provision with Avionics Full Duplex Switching (student paper) ». In : *Euromicro Conference on Real-Time Systems (ECRTS 2017), Dubrovnik, Croatia*. LIPIcs - Leibniz International Proceedings in Informatics, juin 2017, p. 1-4 (cf. p. 49).
- [HP01] J. HUYSSSEUME et P. PALMER. *Nevada - pamela - victoria*. Aeronautics days, 2001 (cf. p. 37).
- [HP85] David HAREL et Amir PNUELI. « On the development of reactive systems ». In : *Logics and models of concurrent systems*. Springer, 1985, p. 477-498 (cf. p. 35).
- [HZL17] Feng HE, Lin ZHAO et Ershuai LI. « Impact Analysis of Flow Shaping in Ethernet-AVB/TSN and AFDX from Network Calculus and Simulation Perspective ». In : *Sensors (Basel, Switzerland)* 17 (mai 2017) (cf. p. 53).
- [IJH09] Jahanzaib IMTIAZ, Jürgen JASPERNEITE et Lixue HAN. « A Performance Study of Ethernet Audio Video Bridging (AVB) for Industrial Real-time Communication. » In : *Emerging Technologies and Factory Automation (ETFA)*. 2009, p. 1-8 (cf. p. 49).
- [Isi10] Yasemin ISIK. « ARINC 629 data bus standard on aircrafts ». In : (jan. 2010) (cf. p. 37).
- [JPMV95] Thomesse JEAN-PIERRE, Zoubir MAMMERI et L. VEGA. « Time in distributed systems cooperation and communication ». In : *5th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS 1995)*. Août 1995, p. 41-49 (cf. p. 44).
- [Kem+13] Georges KEMAYO et al. « Optimistic problems in the trajectory approach in fifo context ». In : *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*. IEEE. 2013, p. 1-8 (cf. p. 53).
- [Kem+14] Georges KEMAYO et al. « A Forward end-to-end delays Analysis for packet switched networks ». In : *22nd International Conference on Real-Time Networks and Systems*. Versailles, France : RTNS 2014, oct. 2014 (cf. p. 63, 65, 78, 85).

- [Kem+15] Georges KEMAYO et al. « Improving AFDX end-to-end delays analysis ». In : *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, p. 1-8 (cf. p. 70, 72, 78).
- [Kha+15] Nima KHALILZAD et al. « Towards Adaptive Resource Reservations for Component-Based Distributed Real-Time Systems ». In : *SIGBED Rev 12.3* (2015), p. 24-27 (cf. p. 49).
- [Lau12] Michaël LAUER. « Une méthode globale pour la vérification d'exigences temps réel : application à l'avionique modulaire intégrée ». Thèse de doct. INPT, 2012 (cf. p. 49).
- [LCG14] Xiaoting LI, Olivier CROS et Libin GEORGE. « The Trajectory Approach for AFDX FIFO networks revisited and corrected ». In : *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2014 IEEE 20th International Conference on*. IEEE, 2014, p. 1-10 (cf. p. 53, 75).
- [Le 98] J. . LE BOUDEC. « Application of network calculus to guaranteed service networks ». In : *IEEE Transactions on Information Theory* 44.3 (1998), p. 1087-1096 (cf. p. 50).
- [Leh90] J.P. LEHOCZKY. « Fixed priority scheduling of periodic task sets with arbitrary deadlines ». In : *[1990] Proceedings 11th Real-Time Systems Symposium*. 1990, p. 201-209 (cf. p. 71, 72).
- [LG17] Xiaoting LI et Laurent GEORGE. « Deterministic delay analysis of AVB switched Ethernet networks using an extended Trajectory Approach ». In : *Real-Time Systems* 53.1 (2017), p. 121-186 (cf. p. 53, 105).
- [Lim+12] Hyung-Taek LIM et al. « Performance Analysis of the IEEE 802.1 Ethernet Audio/Video Bridging Standard ». In : *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques. SIMUTOOLS '12*. ICST (Institute for Computer Sciences, Social-Informatics et Telecommunications Engineering), 2012, p. 27-36 (cf. p. 49).
- [LO11] Phillip A. LAPLANTE et Seppo J. OVASKA. *Real-Time Systems Design and Analysis : Tools for the Practitioner, 4th Edition*. Wiley-IEEE Press, 2011 (cf. p. 35).
- [LSF11] Xiaoting LI, Jean-Luc SCHARBARG et Christian FRABOUL. « Analysis of the pessimism of the trajectory approach for upper bounding end-to-end delay of sporadic flows sharing a switched Ethernet network ». In : *International Conference on Real-Time and Network Systems* (sept. 2011), p. 149-158 (cf. p. 53).
- [Lux12] Syntavision (now : LUXOFT). *pyCPA*. 2012. URL : <https://pycpa.readthedocs.io/en/latest/> (visité le 20/04/2020) (cf. p. 54).
- [LVH11] Hyung-Taek LIM, Lars VÖLKER et Daniel HERRSCHER. « Challenges in a future IP/Ethernet-based in-car network for real-time applications ». In : *Proceedings of the 48th Design Automation Conference*. ACM, 2011, p. 7-12 (cf. p. 49).
- [Mey+13] Philipp MEYER et al. « Extending IEEE 802.1 AVB with time-triggered scheduling : A simulation study of the coexistence of synchronous and asynchronous traffic ». In : *2013 IEEE Vehicular Networking Conference, Boston, MA, USA, December 16-18, 2013*. IEEE, 2013, p. 47-54 (cf. p. 49).

- [Mig+18] Jörn MIGGE et al. « Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks ». In : *Proc. Embedded Real-Time Software and Systems (ERTS 2018)* (2018) (cf. p. 53).
- [MM05] Steven MARTIN et Pascale MINET. « Holistic and Trajectory approaches for Distributed Non-preemptive FP/DP* Scheduling ». In : *ICN*. Berlin Heidelberg : Springer-verlag, 2005, p. 296-305 (cf. p. 50, 57).
- [MM06a] S. MARTIN et P. MINET. « Worst case end-to-end response times of flows scheduled with FP/FIFO ». In : *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*. 2006, p. 54-54 (cf. p. 53).
- [MM06b] Steven MARTIN et Pascale MINET. « Schedulability analysis of flows scheduled with FIFO : application to the expedited forwarding class ». In : *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*. IEEE. 2006, 8-pp (cf. p. 53).
- [MMG06] Steven MARTIN, Pascale MINET et Laurent GEORGE. « The Trajectory Approach for the End-to-End Response Times with Non-preemptive FP/EDF* ». In : *Software Engineering Research and Applications*. Sous la dir. de Walter DOSCH, Roger Y. LEE et Chisu WU. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 229-247 (cf. p. 53).
- [MOG18] Hendrik MEYER, Tatiana ODINETS et Volker GOLLNICK. « Sizing of essential SHM components of IVHM architecture and computing the relevant costs via scalable simulation ». In : *Deutscher Luft- und Raumfahrtkongress, At Friedrichshafen*. Sept. 2018 (cf. p. 38).
- [Moo89] J.F. MOORE. « ARINC 629, the civil aircraft databus for the 1990s ». In : *IEE Colloquium on Time Critical Communications for Instrumentation and Control* (oct. 1989) (cf. p. 37).
- [MZ95] Nicholas MALCOLM et Wei ZHAO. « Hard real-time communication in multiple-access networks ». In : *Real-Time System*, (8). 1995, p. 35-77 (cf. p. 44).
- [NS320] Consortium NS3. *NS-3 : Network Simulator*. 2020. URL : <https://www.nsnam.org/> (visité le 19/03/2020) (cf. p. 49).
- [OMN20] OMNET++. *OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators*. 2020. URL : <https://omnetpp.org/> (visité le 19/03/2020) (cf. p. 49).
- [PJM12] Katalin POPOVICI et Pieter J. MOSTERMAN. *Real-Time Simulation Technologies : Principles, Methodologies, and Applications*. CRC Press, 2012 (cf. p. 48).
- [Qbv] « IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25 : Enhancements for Scheduled Traffic ». In : *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)* (2016), p. 1-57 (cf. p. 131).

- [Que12] Rene QUECK. « Analysis of ethernet avb for automotive networks using network calculus ». In : *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*. IEEE. 2012, p. 61-67 (cf. p. 53).
- [RCM96] I. RIPOLL, A. CRESPO et A.K. MOK. « Improvement in Feasibility Testing for Real-Time Tasks ». In : *J. Real-Time Systems* 11.1 (1996), p. 19-39 (cf. p. 101).
- [Rei+13] F. REIMANN et al. « Timing analysis of Ethernet AVB-based automotive E/E architectures ». In : *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*. 2013, p. 1-8 (cf. p. 54).
- [Rev+11] Kasper REVSBECH et al. « Worst-Case Traversal Time Modelling of Ethernet Based In-Car Networks Using Real Time Calculus ». In : *Smart Spaces and Next Generation Wired/Wireless Networking*. Sous la dir. de Sergey BALANDIN, Yevgeni KOUCHERYAVY et Honglin HU. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011, p. 219-230 (cf. p. 54).
- [Riv20] RIVERBED. *Technologies OPNET - Simulateur de réseau*. 2020. URL : <https://www.riverbed.com/fr/products/steelcentral/opnet.html> (visité le 19/03/2020) (cf. p. 49).
- [RSF08a] Frédéric RIDOUARD, Jean-Luc SCHARBARG et Christian FRABOUL. « Probabilistic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network ». In : *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. IEEE. 2008, p. 1220-1227 (cf. p. 52).
- [RSF08b] Frédéric RIDOUARD, Jean-Luc SCHARBARG et Christian FRABOUL. « Stochastic upper bounds for heterogeneous flows using a Static Priority Queueing on an AFDX network ». In : *Journées FAC (2008)* (cf. p. 52).
- [RSS17] Nejla REJEB, Ahmed Karim Ben SALEM et Slim Ben SAOUD. « AFDX simulation based on TTEthernet model under OMNeT++ ». In : *Advanced Systems and Electric Technologies (IC_ASET), 2017 International Conference on*. IEEE. 2017, p. 423-429 (cf. p. 49).
- [RTC11] Special C. of RTCA. *DO-178C, Software Considerations in Airborne Systems and Equipment Certification*. 2011 (cf. p. 36).
- [SB11] W. M. SOFACK et M. BOYER. « Non preemptive static priority with network calculus ». In : *Emerging Technologies and Factory Automation (ETFA)*. 2011, p. 18 (cf. p. 52).
- [SCG11] Luca SANTINELLI et Liliana CUCU-GROSJEAN. « Toward Probabilistic Real-Time Calculus ». In : *SIGBED Rev.* 8.1 (2011), p. 54-61 (cf. p. 54).
- [SLT05] Françoise SIMONOT-LION et Yvon TRINQUET. « Exemples de systèmes temps réel et choix d'implémentation ». In : *Encyclopédie de l'informatique et des systèmes d'information*. Sous la dir. de Jacky Akoka Isabelle COMYN-WATTIAU. Vuibert, 2005 (cf. p. 35).
- [Spu96] Marco SPURI. *Analysis of Deadline Scheduled Real-Time Systems*. Rapp. tech. 2772. INRIA, 1996 (cf. p. 101).

- [SRF09] Jean-Luc SCHARBARG, Frédéric RIDOUARD et Christian FRABOUL. « A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network ». In : *IEEE Transactions on Industrial Informatics* 5.1 (2009), p. 38-49 (cf. p. 52).
- [SS+15] Chakkaphong SUTHAPUTCHAKUN, Zhili SUN et al. « Impact of end system scheduling policies on afdx performance in avionic on-board data network ». In : *Advanced Informatics : Concepts, Theory and Applications (ICAICTA), 2015 2nd International Conference on*. IEEE. 2015, p. 1-6 (cf. p. 49).
- [Sta88] John A. STANKOVIC. « Misconceptions about real-time computing : a serious problem for next-generation systems ». In : *Computer* 21.10 (1988), p. 10-19 (cf. p. 35).
- [Ste+11] Till STEINBACH et al. « An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy ». In : *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. Barcelona, Spain : ICST (Institute for Computer Sciences, Social-Informatics et Telecommunications Engineering), 2011, p. 375-382 (cf. p. 49).
- [Ste+12] Till STEINBACH et al. « Tomorrow's In-Car Interconnect? A Competitive Evaluation of IEEE 802.1 AVB and Time-Triggered Ethernet (AS6802) ». In : *Proceedings of the 76th IEEE Vehicular Technology Conference, VTC Fall 2012, Quebec City, QC, Canada, September 3-6, 2012*. 2012, p. 1-5 (cf. p. 43, 44, 105).
- [SZ06] Jens SCHMITT et Frank ZDARSKY. « The DISCO network calculator : a toolbox for worst case analysis ». In : *1st International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2006, Pisa, Italy*. Oct. 2006, p. 11-13 (cf. p. 51).
- [TAE15] D. THIELE, P. AXER et R. ERNST. « Improving formal timing analysis of switched ethernet by exploiting FIFO scheduling ». In : *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 2015, p. 1-6 (cf. p. 54).
- [Tan+19] Y. TANG et al. « Pay-Burst-Only-Once in Real-Time Calculus ». In : *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 2019, p. 1-6 (cf. p. 54).
- [TC94] Ken TINDELL et John CLARK. « Holistic Schedulability Analysis for Distributed Hard Real-Time Systems ». In : *Microprocess. Microprogram.* 40.2-3 (avr. 1994), p. 117-134 (cf. p. 50).
- [TCN00] Lothar THIELE, Samarjit CHAKRABORTY et Martin NAEDELE. « Real-time calculus for scheduling hard real-time systems ». In : *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*. T. 4. IEEE. 2000, p. 101-104 (cf. p. 54).
- [Thi+14] D. THIELE et al. « Improving formal timing analysis of switched ethernet by exploiting traffic stream correlations ». In : *2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. 2014, p. 1-10 (cf. p. 54).
- [Tin94] Ken W. TINDELL. « Fixed Priority scheduling of hard real-time systems ». Thèse de doct. University of York, 1994 (cf. p. 50, 57).

- [vic04] Projet européen de recherche VICTORIA. Validation platform for Integration of standardised Components, Technologies and tools in an Open, modular and Improved Aircraft electronic System. 2004 (cf. p. 37).
- [Wan+06] Ernesto WANDELER et al. « System Architecture Evaluation Using Modular Performance Analysis : A Case Study ». In : *Int. J. Softw. Tools Technol. Transf.* 8.6 (nov. 2006), p. 649-667 (cf. p. 54).
- [Zha+18] L. ZHAO et al. « Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus ». In : *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 2018, p. 25-36 (cf. p. 53).
- [ZPC18] L. ZHAO, P. POP et S. S. CRACIUNAS. « Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus ». In : *IEEE Access* 6 (2018), p. 41803-41815 (cf. p. 53).
- [ZS14] Joann ZIMMERMAN et Charles E. SPURGEON. *Ethernet : The Definitive Guide, 2nd Edition (chap. 4)*. O'Reilly. O'Reilly, 2014 (cf. p. 39).

Liste des acronymes

- ADCN** Aircraft Data Communication Network. 35, 36
- AFDX** Avionics Full Duplex switched ethernet. vii, ix, 3, 16–18, 20, 21, 27–29, 35–41, 43, 45, 47–52, 55, 56, 61, 72–74, 79, 87, 90, 93, 104, 119, 125, 126, 149
- ARINC** Aeronautical Radio, Incorporated. 35, 36, 129
- AVB** Audio Video Bridging. vii, ix, 16, 17, 27, 36, 38, 41–43, 45, 47, 48, 51–53, 55, 56, 61, 91, 93, 103–106, 108, 126, 129, 149
- BAG** Bandwidth Allocation Gap. 39, 73
- BDF** Buffer Dimensioning per Frame. iv, v, viii, 17, 109, 111–114, 117–121, 126, 127
- BE** Best-Effort. 41, 103–106
- CAN** Controller Area Network. 35, 36
- CBS** Credit Based Shaper. iv, viii, 17, 41, 51–53, 91, 93–97, 99–106, 108, 126, 129
- CMI** Class Measurement Interval. 41, 103–105
- CPA** compositional Performance Analysis. viii, 52, 53, 55, 56, 103, 105–108, 126
- CPIOM** Core Processing Module. 35
- CPM** Core Processing Module. 35, 36
- CSMA/CD** Carrier Sense Multiple Access with Collision Detection. 37
- DAL** Design Assurance Level. 34
- DIMA** Distributed Integrated Modular Architecture. 36
- DO-178C** Software considerations in airborne systems and equipment certification. 34
- ECU** Electronic Control Unit. 36, 41, 103
- ES** End System. 38, 39, 45, 73, 74, 103, 104, 119
- FA** Forward end-to-end delay Analysis. iv, vii, viii, 3, 17, 18, 21, 22, 27, 28, 56, 59, 61–64, 70–77, 79, 81, 83, 87–90, 93, 95, 97, 103–108, 117, 119, 120, 125, 126, 129, 130
- FIFO** First-In-First-Out. iv, viii, 51, 55, 56, 59, 61, 62, 64–67, 70–72, 76, 79–84, 87, 93, 103, 113, 114, 125, 126
- Fmax** Maximum frame size. 39, 73
- FP/EDF*** Fixed Priority/Earliest Deadline First. 51
- FP/FIFO** Fixed Priority/First-In-First-Out. iv, viii, 17, 39, 41, 51, 55, 56, 76, 77, 79–83, 85, 87–90, 93, 95, 101, 103, 104, 108, 125, 126, 129
- GWM** GateWay Module. 36

IEEE Institute of Electrical and Electronics Engineers. 41, 129

IMA Integrated Modular Architecture. 35, 36

IOM Input/Output Module. 36

IRIT Institut de Recherche en Informatique de Toulouse. 56

LIAS Laboratoire d'Informatique et d'Automatique pour les Systèmes. 56

LIN Local Interconnect Network. 36

LPT Longest Processing Time. 115–118

MFS Maximum Frame Size. 41, 104, 105

MIF Maximum Interval Frames. 41, 103–105

MOST Media Oriented Systems Transport. 36

NC Network Calculus. vii, 48–52, 55, 56, 61, 73–76, 87, 89, 90, 117, 120, 125, 126

NS3 Network Simulator. 47

OMNeT++ Objective Modular Network Testbed in C++. 47

OSI Open Systems Interconnection. vii, 42, 43

PAMELA Process for Advanced Management of End of life Aircraft. 35

PBOO Pay-Burst-Only-Once. 52, 73

RAM Random Access Memory. 35

RBF Request Bound Function. 66, 68, 83, 113–117

RTC Real-Time Calculus. 52

SPT Shortest Processing Time. 116–118

SR Stream Reservation. 41, 53, 103–105

TA Trajectory Approach. vii, 51, 55, 73–76, 87, 89, 90, 103, 125, 126

TAS Timing and Synchronization. 41, 129

TDMA Time Division Multiple Access. 41

TSN Time-Sensitive Networking. v, 17, 18, 41, 51–53, 56, 61, 129

TT Time Triggered. 41

VICTORIA Validation platform for Integration of standard Components, Technologies and tools in an Open, modular and Improved Aircraft electronic system. 35

VL Virtual Link. 38, 39, 73, 74, 104

W4L Workload For Labs. 3, 21

Glossaire

- $BHP_i^h(t)$ Charge de travail minimum générée par hp_i^h durant un intervalle de durée t dans le nœud h en décalant la rafale de trames hp_i^h à l'instant t .. 85
- $Bklg^h$ Arriéré temporel pire-cas présent simultanément dans le nœud h .. 63–65, 68, 69, 71, 80, 88
- $Bklg_i^h$ Arriéré temporel pire-cas présent simultanément dans le nœud h à l'arrivée d'une frame générée par le flux v_i .. 88
- $Bord_{X_i}^h(t)$ Charge de travail maximum générée par la classe X_i dans le nœud h selon le scénario de Bordoloi.. 99, 100
- $Bord_Y^h(t)$ Charge de travail maximum générée par la classe Y dans le nœud h selon le scénario de Bordoloi.. 99
- BP^{X_i} La plus longue période d'activité des classes plus prioritaires que la classe X_i dans le nœud h .. 98
- C_i^h Temps de transmission maximum sur le nœud h des trames générées par le flux v_i et $C_i^h = \frac{Fmax_i}{r^h}$. 62, 64, 67, 68, 70, 71, 81, 82, 84–86, 88, 104, 114, 117, 118
- $Crd_{X_i,max}^h$ Crédit maximum de la classe X_i dans le nœud h .. 98, 99
- $Crd_{Y,max}^h$ Crédit maximum de la classe Y dans le nœud h .. 98
- $dep_i^h(t)$ Instant pire-cas de début de transmission d'une trame du flux v_i arrivée au le nœud h à l'instant t .. 81–86, 95, 96, 100, 102
- f_i Trame générée par le flux v_i , sous analyse.. 62–66, 80–85, 95–102, 118
- $first_i$ Nœud source du flux v_i .. 62–65, 67, 70, 71, 88, 114
- $Fmax_i$ Taille maximale des trames générées par le flux v_i .. 62, 71, 73, 88, 112
- h_i nœud du chemin \mathcal{P}_i de v_i .. 72
- $hp_{0,i}^h$ Ensemble des flux de priorité strictement plus grande que celle du flux v_i et générés par le nœud h .. 80, 84, 85, 100, 102
- hp_i^h Ensemble des flux plus prioritaires que le flux v_i au niveau du nœud h .. 80–82, 84, 86, 93, 96, 98, 99, 101, 102, 145–147
- $hp_{x,i}^h$ Ensemble des flux strictement plus prioritaires que le flux v_i au niveau du nœud h en provenance de IP_x^h .. 80, 84–86, 100, 102, 146
- HPX_i^h Ensemble des classes plus prioritaires que la classe X_i au nœud h .. 93, 98
- h_x nœud précédent du nœud h , via le lien entrant IP_x^h ($1 \leq x \leq n$.. 68, 69, 115
- IP_x^h Lien d'entrée x du nœud h .. vii, 61, 62, 67–69, 71–73, 80, 84–86, 97, 99, 100, 115–118
- J_i^h Gigue maximum du flux v_i sur le nœud h .. 66, 67, 71, 88, 113, 114, 117, 118

- r^h Débit du lien de sortie du nœud h .. 61, 62, 65, 68, 69, 71, 88, 94, 104, 105, 111
- k_i^h Nombre de trames maximum générée par le flux v_i arrivant simultanément dans le nœud h .. 66, 68, 114
- $last_i$ Nœud destination du flux v_i .. 62, 64, 71, 88
- $LinkRate_x^h(t)$ Temps de transmission total arrivant sur le nœud h via le lien IP_x^h durant l'intervalle $[0, t]$, en considérant uniquement le débit de ce lien physique.. 68, 69, 71, 72, 86, 99, 102
- lp_i^h Ensemble des flux moins prioritaires que le flux v_i au niveau du nœud h .. 80–83, 86, 98, 101, 102, 147
- $lp_{x,i}^h$ Ensemble des flux moins prioritaires que le flux v_i au niveau du nœud h via le lien IP_x^h .. 80
- NBF^h Nombre de trames maximum présentes simultanément dans le nœud h .. 117, 118
- \mathcal{P}_i chemin du flux v_i : Liste ordonnée des nœuds depuis son nœud source $first_i$, jusqu'à son nœud destination $last_i$.. 62–65, 70–72, 79, 88, 114, 145
- P_i Niveau de priorité du flux v_i . 79–83, 86, 93, 104, 105
- $rbf_i^h(t)$ fonction RBF du flux v_i dans le nœud h pour un intervalle de temps de durée t .. 66–69, 71, 83, 114, 117, 118
- $rbf_x^h(t)$ Temps de transmission total arrivant sur le nœud h via le lien IP_x^h durant l'intervalle $[0, t]$, comme somme des fonctions RBFs des flux arrivant sur h , via le lien IP_x^h , durant l'intervalle $[0, t]$.. 68, 69, 71, 72, 99
- R_i Délai de traversée pire cas de bout-en-bout du réseau pour toute trame générée par le flux v_i .. 70–72, 88
- R_i^h Délai de traversée pire cas du réseau pour toute trame générée par le flux v_i , entre sa génération sur $first_i$ jusqu'à la fin de sa transmission par le nœud h .. vii, 63, 64, 70
- $ShaperRate_{x,i}^h(t)$ Temps de transmission total arrivant sur le nœud h via le lien IP_x^h durant l'intervalle $[0, t]$, en considérant uniquement le crédit de la classe du flux.. 99, 102
- shp_i^h Ensemble des flux sp_i^h et hp_i^h .. 80, 81, 95
- $shp_{x,i}^h$ Ensemble des flux $sp_{x,i}^h$ et $hp_{x,i}^h$.. 80, 86, 99
- $\sigma_{X_i}^{h,-}$ SendSlope de la classe X_i dans le nœud h .. 93, 94, 96, 98, 99, 101, 102, 104
- $\sigma_{X_i}^{h,+}$ IdleSlope de la classe X_i dans le nœud h .. 93, 94, 96, 98, 99, 101, 102, 104
- $Smax_i^h$ Délai de traversée pire cas du réseau pour toute trame générée par le flux v_i , entre sa génération sur $first_i$ jusqu'à son arrivée sur le nœud h .. vii, 63, 64, 66, 67, 71, 88, 112, 113
- $Smin_i^h$ Délai de traversée meilleur cas du réseau pour toute trame générée par le flux v_i , entre sa génération sur $first_i$ jusqu'à son arrivée sur le nœud h .. 64, 66, 67, 71, 88, 112, 113
- $sp_{0,i}^h$ Ensemble des flux de même priorité que le flux v_i et générés par le nœud h .. 80, 85
- sp_i^h Ensemble des flux de même priorité que le flux v_i au niveau du nœud h .. 80–84, 146, 147
- $sp_{x,i}^h$ Ensemble des flux de même priorité que le flux v_i au niveau du nœud h en provenance de IP_x^h .. 80, 84–86, 99, 146

T_i Période du flux v_i .. 62, 65–67, 70, 71, 73, 88, 104, 105, 112, 114, 117, 118

U^h Charge locale du nœud h .. 70

v_i un flux du réseau ($v_i \in \Gamma$.. vii, viii, 62–73, 80–83, 88, 93, 95–97, 99, 100, 104, 105, 112, 114, 115, 117, 118, 145, 147

$W^h(t)$ Charge de travail maximale arrivée dans la file d'attente du nœud h , durant un intervalle de temps de durée t .. 65, 67–72, 80

$W_i^h(t)$ Charge de travail maximale arrivée dans la file d'attente du nœud h , durant un intervalle de temps de durée t pour le flux v_i in Γ^h .. 80–82, 84, 86, 88, 96, 97

$WHP_i^h(t)$ Charge de travail maximum générée par les flux de hp_i^h durant un intervalle de durée t dans le nœud h .. 81, 85, 97

$W_{x,i}^h(t)$ Charge de travail maximale arrivée dans la file d'attente du nœud h , via le lien entrant IP_x^h , durant un intervalle de temps de durée t et retardant une trame f_i du flux v_i .. 97, 99

$W_x^h(t)$ Charge de travail maximale arrivée dans la file d'attente du nœud h , via le lien entrant IP_x^h , durant un intervalle de temps de durée t .. 67–69, 71, 72

$WLP_i^h(t)$ Charge de travail maximum générée par les flux de lp_i^h durant un intervalle de durée t dans le nœud h .. 81, 82, 97

$WNIL_i^h(t)$ Temps de réapprovisionnement du crédit des $WSP_i^h(t)$ en comptant la charge minimum des hpX .. 97, 100, 101

$WSP_i^h(t)$ Charge de travail maximum générée par les flux de sp_i^h durant un intervalle de durée t dans le nœud h .. 81, 82, 85, 97

X_i^h Ensemble des flux traversant le nœud h de la classe X_i .. 99

X_i Classe de trafic de priorité i (avec $1 \leq i | eqm$). 91, 93–101, 145, 147

Résumé — J'ai obtenu mon doctorat d'informatique en 2006 à l'Université de Poitiers. Mon doctorat s'est déroulé au LISI (ex-LIAS) sur des contributions à des problèmes d'ordonnancement temps réel. Après mon doctorat, j'ai intégré l'IRIT de Toulouse où j'ai développé ma thématique de recherche sur la validation temporelle des réseaux embarqués temps réel que je continue de développer aujourd'hui au LIAS. En particulier, je travaille à la démonstration de deux points du déterminisme des réseaux embarqués critiques : calcul d'une borne supérieure garantie sur le délai de bout en bout de traversée du réseau pour chaque trame et un dimensionnement pire cas garanti en nombre de trames de chaque file d'attente du réseau. Ces travaux ont pour contexte le domaine de l'avionique (AFDX) et de l'automobile (AVB).

Mots clés : Réseaux embarqués, déterminisme, délai de bout en bout, dimensionnement des files d'attente

Laboratoire d'Informatique et d'Automatique pour les Systèmes (LIAS)

ISAE-ENSMA / Université de Poitiers