

THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
École nationale supérieure d'ingénieurs (Poitiers)
Laboratoire d'informatique et d'automatique pour les systèmes - LIAS (Poitiers)
(Diplôme National - Arrêté du 25 mai 2016)

École doctorale : Sciences et ingénierie pour l'information, mathématiques - S2IM (Poitiers)
Secteur de recherche : Informatique et applications

Présentée par :
Nassima Benammar

Modélisation, évaluation et validation des systèmes temps réel distribués

Directeur(s) de Thèse :
Pascal Richard, Frederic Ridouard

Soutenue le 17 septembre 2018 devant le jury

Jury :

Président	Laurent George	Professeur, LIGM, Université de Paris-Est
Rapporteur	Ahlem Mifdaoui	Professeur, ISAE-Supaéro, Université de Toulouse
Rapporteur	Jean-Luc Scharbarg	Professeur, IRIT/ENSEEIHT, Toulouse
Membre	Pascal Richard	Professeur, LIAS, Université de Poitiers
Membre	Frederic Ridouard	Maître de conférences, LIAS, Université de Poitiers
Membre	Henri Bauer	Maître de Conférences, LIAS, ENSMA, Poitiers

Pour citer cette thèse :

Nassima Benammar. *Modélisation, évaluation et validation des systèmes temps réel distribués* [En ligne]. Thèse Informatique et applications. Poitiers : Université de Poitiers, 2018. Disponible sur Internet
<<http://theses.univ-poitiers.fr>>

THESE

Pour l'obtention du Grade de

DOCTEUR DE L'UNIVERSITE DE POITIERS

(Faculté des Sciences Fondamentales et Appliquées)
(Diplôme National - Arrêté du 25 mai 2016)

Ecole Doctorale Sciences et Ingénierie pour l'Information, Mathématiques
Secteur de Recherche : Informatique et Applications

Présentée par :

Nassima BENAMMAR

Modélisation, évaluation et validation des systèmes temps réel distribués

Directeur de Thèse : Pascal RICHARD

Co-encadrants : Henri BAUER, Frédéric RIDOUARD

Soutenue le 17/09/2018
devant la Commission d'Examen

JURY

Laurent GEORGE	Professeur, LIGM/ Université de Paris-Est	Président
Ahlem MIFDAOUI	Professeur, DISC/ ISAE-Supaéro, Université de Toulouse	Rapporteur
Jean-Luc SCHARBARG	Professeur, IRT/ENSEEIH, Toulouse	Rapporteur
Frédéric RIDOUARD	Maître de Conférences, LIAS/Université de Poitiers	Examineur
Henri BAUER	Maître de Conférences, LIAS/ENSMA, Poitiers	Examineur
Pascal RICHARD	Professeur, LIAS/Université de Poitiers	Examineur

*À mon mari,
mes parents,
ma sœur,
et tout ceux qui me sont chers.*

Remerciements

Je souhaite adresser ma reconnaissance et mes remerciements à toutes les personnes qui m'ont accompagnée tout au long de cette thèse. Je remercie mon directeur de thèse Pascal RICHARD, et mes co-encadrants, Frédéric RIDOUARD et Henri BAUER, qui m'ont guidée et qui ont beaucoup contribué à la réussite de cette thèse. Je les remercie également pour leur intégrité, leur disponibilité et leur générosité à partager leurs connaissances. Merci à chacun d'eux pour leur implication qui m'a permise de terminer ma thèse à temps.

Je souhaite également remercier le directeur du laboratoire Emmanuel GROLLEAU de m'avoir bien accueillie et pour sa bonne humeur.

Je remercie Ahlem MIFDAOUI et Jean-Luc SCHARBARG pour avoir accepté la lourde tâche de rapporter mon manuscrit. Je vous remercie pour tout l'intérêt que vous avez accordé à mes travaux de recherche. Je remercie également Laurent GEORGE qui m'a fait l'honneur de présider le jury de ma thèse.

Je voudrais également remercier tous les membres du LIAS qui m'ont accompagnée de près ou de loin et contribué à l'épanouissement de mon quotidien. Merci à tous ceux que j'ai côtoyés quotidiennement pour les débats interminables mais toujours constructifs.

Je souhaite exprimer toute ma reconnaissance à mes proches qui m'ont soutenue de très loin tout au long de ma thèse, surtout mes parents, ma sœur et son mari, et ma meilleure amie Maria. Et je ne pourrais terminer sans citer celui qui m'a soutenue pour le meilleur et pour le pire, mon cher mari Kamel.

Table des matières

Remerciements	iii
Liste des figures	vii
Liste des tableaux	xi
1 Introduction	1
1.1 Contexte du projet et motivations	3
1.2 Problématiques et objectifs	3
1.3 Principales contributions	4
1.4 Organisation du document	5
I Etat de l’art	7
2 Les systèmes temps réel distribués	9
2.1 Introduction	11
2.2 Système distribué temps réel	11
2.3 Évolution des systèmes temps réel distribués	12
2.4 Le domaine avionique : <i>Integrated Modular Architecture</i>	15
2.5 Le domaine automobile	17
2.6 Conclusion	20
3 Les réseaux temps réel	21
3.1 Introduction	23
3.2 Réseau informatique	23
3.3 Évolution des réseaux temps réel	29
3.4 Le réseau Ethernet dans le temps réel	32
3.5 <i>Avionic Full Duplex Switched Ethernet</i> (AFDX)	33
3.6 <i>Audio/Vidéo Bridging Switched Ethernet</i> (AVB)	39
3.7 Conclusion	42
4 Évaluation des réseaux temps-réel	43
4.1 Introduction	45
4.2 Problématique	45
4.3 Les principales méthodes d’analyse	46
4.4 Conclusion	53

II Contributions	55
5 La méthode FA en FIFO	57
5.1 Introduction	59
5.2 Modèle réseau	60
5.3 Le calcul du délai pire cas d'une trame	61
5.4 Expérimentations	72
5.5 Conclusion	77
6 La méthode FA en FP/FIFO	79
6.1 Introduction	81
6.2 Modèle réseau en FP/FIFO	81
6.3 Le calcul du délai pire cas d'une trame en FP/FIFO	82
6.4 Expérimentations en FP/FIFO	91
6.5 Conclusion	96
7 La méthode FA en CBS	97
7.1 Introduction	99
7.2 Modèle réseau en CBS	99
7.3 Le calcul du délai pire cas d'une trame en CBS	101
7.4 Expérimentations en CBS	107
7.5 Conclusion	115
8 Dimensionnement des Files d'attente	117
8.1 Introduction	119
8.2 Etat de l'art et motivation	120
8.3 Dimensionnement des files d'attente FIFO, sans sérialisation	121
8.4 Dimensionnement des files d'attente FIFO, avec sérialisation	124
8.5 Expérimentations	125
8.6 Conclusion	131
9 Conclusion	133
9.1 Principaux résultats	135
9.2 Perspectives	136
Bibliographie	137
Liste des acronymes	145
Liste des notations	147

Liste des figures

2.1	Schéma d'un système temps réel.	11
2.2	L'évolution des systèmes avioniques d'Airbus.	13
2.3	Schéma d'une architecture répartie dans l'avionique.	14
2.4	Schéma d'un LRM de type CPM.	16
2.5	Les réseaux de l'architecture IMA.	17
2.6	Exemple de systèmes automobiles avec des ECUs [101].	18
2.7	L'évolution du marché chinois des ECUs dans le domaine de l'automobile [88].	19
2.8	Les systèmes de communications automobiles [47].	20
3.1	Schéma d'un réseau informatique.	23
3.2	Les types de réseaux selon la distance.	24
3.3	La topologie en anneau.	25
3.4	La topologie en bus.	25
3.5	La topologie en étoile.	25
3.6	La topologie en arbre.	26
3.7	La topologie maillée.	26
3.8	La topologie linéaire.	27
3.9	Le modèle en couches.	27
3.10	Le dialogue entre les couches.	28
3.11	Schéma d'un réseau ARINC 429.	30
3.12	Schéma d'un réseau CAN.	30
3.13	Les systèmes de communications automobiles [1].	31
3.14	Schéma d'un réseau Ethernet commuté.	32
3.15	La structure d'une trame Ethernet	32
3.16	Schéma de la configuration avionique du réseau illustré dans la Fig. 2.5. . .	34
3.17	Régulation de l'émission de trames d'un flux.	35
3.18	Schéma d'un ES.	35
3.19	Schéma d'un commutateur AFDX Ethernet.	36
3.20	Modélisation d'une file d'attente FPFIFO.	36
3.21	Exemple d'un ordonnancement FP/FIFO.	37
3.22	Configuration d'un réseau AFDX.	38
3.23	Exemple de domaines AVB.	39
3.24	Modélisation d'un port de sortie dans le réseau TSN.	41
4.1	Les résultats d'une analyse par simulation.	45
4.2	La courbe d'arrivée d'un flux.	48
4.3	La courbe de service et la courbe d'arrivée d'un nœud N.	48

4.4	Scénario possible de traversée de la trame 3 du flux v_3 .	49
4.5	Scénario pire cas de traversée de la trame 3 du flux v_3 .	50
4.6	Les parties constantes et variables du délai de bout en bout.	52
5.1	Modélisation d'un nœud h .	60
5.2	Calcul itératif de R_i^h et $Smax_i^h$.	61
5.3	Illustration de la période d'activité pour une valeur de $t \geq 0$ donnée.	62
5.4	Le scénario pire cas d'arrivée des trames générées par $v_i \in I^h$ dans le nœud h , maximisant l'arriéré de travail durant un intervalle de temps $[0, t]$.	63
5.5	La première trame arrivant dans h après un délai $d < Smax_i^h$.	64
5.6	Modèle de nœud avec des ports d'entrée.	67
5.7	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 .	68
5.8	Les instants t à tester.	72
5.9	Une configuration AFDX.	73
5.10	La modélisation de la configuration représentée par la Fig. 5.9.	74
5.11	Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA).	76
6.1	Modèle nœud en FP/FIFO.	82
6.2	Les scénarios d'arrivée pire cas des trames (tel que $v_j \in I^h$) où la rafale de $(1 + k_j^h)$ trames arrive à l'instant 0 (a) ou à l'instant t (b).	85
6.3	Un exemple de configuration AFDX pour illustrer l'impact de l'ordre d'arrivée des trames avec la politique FP/FIFO.	86
6.4	Le scénario d'arrivée en plaçant la rafale de trames à l'instant $t = 0$.	86
6.5	Le scénario d'arrivée de la rafale de trames arrive à l'instant 0 pour les flux de sp_i^h et à l'instant $t = 40 \mu s$ pour ceux de hp_i^h .	87
6.6	Configuration AFDX utilisée pour tester la version FP/FIFO de FA.	91
6.7	Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA) en FP/FIFO.	95
7.1	Un exemple d'ordonnancement suivant le régulateur CBS.	100
7.2	Scénario pire cas défini et prouvé par Bordoloi <i>et al.</i> [31] maximisant le temps de transmission total généré par une classe Y sur un nœud CBS durant un intervalle de temps de durée t .	102
7.3	Configuration d'un réseau AVB pour les applications ADAS et <i>Infotainment</i> .	108
7.4	Port de sortie AVB	109
7.5	Modélisation de la configuration représentée dans la Fig. 7.3.	110
7.6	Scénario pire cas pour le flux v_3 .	112
7.7	Configurations AVB.	113
7.8	Comparaison des résultats obtenus par FA et CPA pour la configuration représentée par la Fig. 7.7.a.	114
7.9	Comparaison des résultats obtenus par FA et CPA pour la configuration représentée par la Fig. 7.7.b.	115
8.1	Contre-exemple sur la différenciation entre les arriérés pire cas en bits et en trames.	120
8.2	Scénarios d'arrivée dans une file d'attente FIFO.	121

8.3	Instants d'arrivée au plus tôt des trames de ν_i dans une file d'attente FIFO h , en considérant $rbf_i^h(t)$	122
8.4	Optimalité de LPT pour minimiser le nombre de trames transmises.	123
8.5	Configuration AFDX pour l'analyse du dimensionnement de file d'attente.	125
8.6	La configuration AFDX représenté par la Fig. 8.5 transposé au modèle réseau de FA.	126
8.7	Évolution du nombre de trames dans la file d'attente du nœud S_{21}	127
8.8	Différence en pourcentage entre les bornes pire cas sur le nombre de trames présentes simultanément dans chaque port de sortie de la configuration de l'A380 entre obtenues par notre méthode de dimensionnement par trame et la méthode naïve.	130

Liste des tableaux

4.1	Synthèse bibliographique des principales méthodes d'analyse.	53
5.1	Les caractéristiques des flux et leurs délais de bout en bout pire cas (par NC et FA)	74
6.1	Les caractéristiques des flux de l'exemple dans la Fig. 6.3	86
6.2	Les caractéristiques des flux de la configuration dans la Fig. 6.6	91
6.3	Les délais de bout en bout pire cas (en μ s) de la configuration dans la Fig. 6.6.	94
7.1	Les caractéristiques des flux et les délais pire cas calculés par FA pour la configuration dans la Fig. 7.3.	108
8.1	Caractéristiques des flux de la configuration dans la Fig. 8.5.	126
8.2	Comparaison des approches de dimensionnement de file d'attente FIFO en nombre de trames.	129

Chapitre 1

Introduction

Sommaire

1.1	Contexte du projet et motivations	3
1.2	Problématiques et objectifs	3
1.3	Principales contributions	4
1.4	Organisation du document	5

1.1 Contexte du projet et motivations

L'informatisation des systèmes dans les domaines de l'avionique et de l'automobile implique une évolution majeure. Ces systèmes sont dits temps-réel car à des contraintes fonctionnelles, s'ajoute des contraintes temporelles. Pour répondre aux besoins croissants, les systèmes temps réel sont basés sur une architecture distribuée. L'objectif de cette architecture est le partage de ressources et de permettre l'échange de données entre les fonctions via un média de communication appelé le réseau. Le contexte de cette thèse se situe dans l'analyse des réseaux utilisés dans les systèmes temps réel distribués, particulièrement dans les domaines de l'avionique et de l'automobile. Ainsi, nous étudions le réseau *Avionic Full Duplex Switched Ethernet* (AFDX) dans le contexte avionique et le réseau *Audio Vidéo Bridging Switched Ethernet* (AVB), populaire dans le milieu automobile. Ces deux protocoles réseaux sont basés sur le standard Ethernet et bénéficient, ainsi, de ses avantages (haut débit, disponibilité, facilité de maintenance et faible coût).

L'AFDX et l'AVB sont des réseaux commutés. Ils sont composés de plusieurs terminaux (points d'entrée/sortie) reliés via un ensemble de commutateurs et de liens physiques. Chaque commutateur possède une file d'attente par port de sortie régie par une politique de service (*p.ex.* la politique *First-In-First-Out*, FIFO). En se plaçant dans le cas nominal (absence de panne), la démonstration du déterminisme de ces réseaux est requis pour la certification. En particulier, il faut prouver que le délai de bout-en-bout de chaque flux respecte son échéance et qu'aucune trame ne soit perdue :

Le délai de bout en bout : le temps de traversée du réseau par chacun des messages doit être analysé. Il faut démontrer que même dans le scénario le plus défavorable, le délai de traversée n'excède pas une échéance fixée.

La taille des files d'attente : les liens utilisés dans les réseaux commutés temps réel tels que l'AFDX ou l'AVB assurent qu'aucune collision n'est possible sur ces liens physiques. Ainsi, seul un dépassement d'une file d'attente peut entraîner la perte d'une trame. Par conséquent, il doit être démontré que la taille des files d'attente est suffisante quel que soit le scénario, pour s'assurer de la non perte de trame.

1.2 Problématiques et objectifs

La problématique principale de cette thèse est la modélisation, l'évaluation et la validation des systèmes de communication temps réel.

Ainsi, la modélisation du réseau doit permettre son évaluation et la démonstration de son déterminisme. La notion de déterminisme s'entend par la garantie d'aucune perte ou dépassement de délai dans le cas nominal (absence de panne), par opposition à une garantie probabiliste qui pourrait tolérer un taux de panne ou de dépassement inférieur à un seuil. Les réseaux étudiés sont : l'AFDX conçu pour les réseaux avioniques et l'AVB développé dans le domaine de l'automobile.

Plusieurs approches d'analyse pire cas du délai de bout en bout telles que *Network Calculus* (NC) [39, 40, 32] et la méthode des Trajectoires (TA) [77, 25] ont été développées pour l'AFDX (avec la politique de service FIFO). Elles ont été étendues à la différenciation de service avec l'introduction de priorités fixes au niveau des flux avec la politique de service *Fixed Priority/First-In-First-Out* (FP/FIFO). Ces approches ont également été appliquées au réseau AVB. La méthode *Forward ETE delay Analysis* (FA) [66] obtient des

résultats très intéressants comparés à ceux obtenus avec NC et TA. Cependant, FA n'a été développée que dans le contexte de l'AFDX avec la politique FIFO. Par conséquent, un des objectifs de ce travail est d'étendre FA à la politique de service FP/FIFO mais également au réseau AVB et sa politique *Credit Based Shaper* (CBS). Au préalable, FA devra être reformulée dans sa version FIFO afin de faciliter son extension.

Le problème du dimensionnement des files d'attente est connexe à celui de l'analyse du délai pire cas de traversée d'un réseau. En effet, dans un nœud, le temps d'attente maximal subi par une trame, correspond au temps de transmission de la plus grande quantité de données qui peut être présente dans la file d'attente à son instant d'arrivée. La taille des files d'attente peut être exprimée :

- en nombre de bits (allocation dynamique de la mémoire) : à chaque demande, il est alloué exactement l'espace mémoire demandé ;
- en nombre de trames (allocation statique) : la file d'attente est composée d'un ensemble d'espaces mémoire de même taille. Quel que soit la requête, à chaque demande, un emplacement mémoire est alloué.

Dans les systèmes critiques tels que l'avionique ou l'automobile, l'allocation mémoire statique est privilégiée. Cependant, au meilleur de nos connaissances, dans la littérature, il existe peu de travaux concernant le dimensionnement des files d'attente en nombre de trames. Ainsi dans cette thèse, nous avons développé une approche qui nécessite, une topologie réseau, des contrats de régulation de trafic à l'entrée du réseau et une méthode d'évaluation des délais de bout-en-bout (NC, TA, ou FA) et qui détermine une borne supérieure garantie de la taille de chaque file d'attente FIFO d'un réseau en nombre de trames.

1.3 Principales contributions

Les principales contributions développées durant cette thèse sont :

Analyse de délai de bout en bout pire cas avec la politique FIFO pour le réseau AFDX :

FA a déjà été définie dans un contexte AFDX avec la politique de service FIFO. Cependant, durant cette thèse, nous avons généralisé le modèle réseau supporté par FA. De plus, plusieurs éléments importants dont le scénario pire cas ont été reformulés afin de faciliter l'extension de FA à la différenciation de service et en particulier aux politiques de service FP/FIFO et CBS.

Analyse de délai de bout en bout pire cas avec la politique FP/FIFO pour le réseau AFDX :

FA est étendue à la politique de service FP/FIFO en incluant le paradigme de l'effet de sérialisation.

Analyse de délai de bout en bout pire cas du avec la politique de service CBS pour le réseau AVB :

FA est appliquée au réseau AVB en incluant une modélisation et une prise en compte de la politique *Credit Based Shaper* (CBS). Cette extension inclut la prise en compte de la sérialisation.

Dimensionnement de file d'attente dans le contexte AFDX : Nous proposons une approche qui détermine des bornes supérieures garanties sur la taille des files d'attente FIFO pire cas en nombre de trames. Cette approche nécessite une méthode d'évaluation des délais de bout-en-bout dans un réseau mais, pas de méthode précise en particulier. Nous prenons en compte l'effet de sérialisation afin d'améliorer nos résultats.

1.4 Organisation du document

Ce manuscrit est constitué de deux parties : état de l'art et contributions.

L'état de l'art est constitué de trois chapitres. Dans le chapitre 2, nous présentons les systèmes temps réel utilisés dans les domaines de l'avionique et de l'automobile ainsi que l'évolution de leurs architectures. Ensuite, chapitre 3, nous présentons une synthèse bibliographique sur les réseaux informatiques temps réel. Nous nous focalisons, en particulier, sur deux standards : l'AFDX et l'AVB. Finalement, le chapitre 4 présente un état de l'art des principales méthodes d'analyse du délai pire cas de traversée du réseau par une trame, appliquées à ces protocoles.

La partie détaillant les contributions est composée de quatre chapitres. Chaque chapitre de contribution est composé d'une partie théorique incluant des démonstrations formelles des résultats et d'une partie expérimentale avec application de chaque contribution sur une configuration industrielle réelle.

La méthode FA a déjà été définie pour le réseau AFDX avec la stratégie FIFO. Cependant, dans le chapitre 5, nous généralisons le modèle réseau de FA et reformulons plusieurs points. Dans le chapitre 6, FA est étendue à la politique de service FP/FIFO puis au réseau AVB dans le chapitre 7. Finalement, chapitre 8, nous présentons une méthode de dimensionnement pire cas des files d'attente FIFO.

PREMIÈRE PARTIE :

ETAT DE L'ART

Chapitre 2

Les systèmes temps réel distribués

Sommaire

2.1	Introduction	11
2.2	Système distribué temps réel	11
2.2.1	Système informatique	11
2.2.2	Système temps réel distribué	11
2.3	Évolution des systèmes temps réel distribués	12
2.3.1	Le cas de l'avionique	12
2.3.2	Historique des architectures des systèmes temps réel	13
2.4	Le domaine avionique : <i>Integrated Modular Architecture</i>	15
2.4.1	La partie système de l'architecture IMA	15
2.4.2	La partie réseau de l'architecture IMA	16
2.5	Le domaine automobile	17
2.5.1	Les systèmes temps réel de l'automobile	17
2.5.2	Les sous-systèmes d'un véhicule	18
2.5.3	Les systèmes automobiles modernes	19
2.5.4	L'architecture de communications des systèmes automobiles	19
2.6	Conclusion	20

2.1 Introduction

Dans cette thèse, nous nous focalisons sur l'analyse de performances dans les systèmes temps réel distribués des domaines de l'avionique et de l'automobile. Dans ce premier chapitre de l'état de l'art, nous définissons d'abord les systèmes temps réel distribués. Ensuite nous prenons l'exemple de l'avionique pour illustrer l'évolution des architectures temps réel. Ensuite, nous présentons les principaux systèmes temps réel utilisés dans le domaine de l'automobile avec leurs architectures.

2.2 Système distribué temps réel

2.2.1 Système informatique

Un système informatique est défini par un ensemble organisé d'équipements logiciels et matériels dont l'objectif premier est de répondre à des besoins fonctionnels.

Les systèmes informatiques sont de trois types différents :

- les systèmes réactifs [56] : ils fonctionnent en interaction avec l'environnement (comme les systèmes automobiles).
- les systèmes interactifs : ils requièrent l'intervention d'un utilisateur (par exemple les logiciels bureautiques).
- les systèmes transformationnels : activités de calculs, base de données, etc.

Les systèmes temps réel distribués sont des systèmes réactifs.

2.2.2 Système temps réel distribué

Un système est dit temps réel si son bon fonctionnement ne dépend pas uniquement de l'exactitude de ses résultats mais également des temps auxquels ils surviennent [96]. Il contrôle un procédé (cf. Fig. 2.1) et a pour objectif de le conserver dans un état stable.

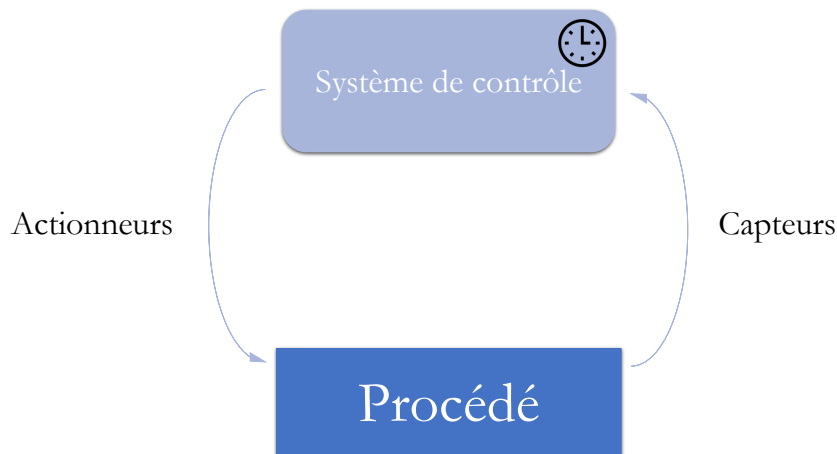


FIGURE 2.1 – Schéma d'un système temps réel.

Le système temps réel récupère des données à partir de capteurs sur le procédé. Il effectue des calculs et contrôle le procédé par l'intermédiaire des actionneurs. Ces opérations doivent être effectuées en respectant les contraintes temporelles afin d'éviter une défaillance.

Les systèmes temps réel sont classifiés en deux catégories selon les conséquences du non-respect d'une contrainte temporelle :

- système temps réel critique ou *dur* : le non-respect d'une contrainte temporelle engendre la perte du procédé contrôlé (par exemple les systèmes avioniques) avec des conséquences potentiellement catastrophiques comme des morts ou des blessés graves, des dégâts matériels importants, ou des conséquences graves pour l'environnement.
- système temps réel non critique ou *mou* : le non-respect d'une contrainte temporelle entraîne une dégradation du procédé contrôlé mais pas une perte (par exemple le système embarqué d'un téléphone), sans conséquence dramatique.

Dans la suite, nous nous focalisons sur les systèmes temps réel critiques appliqués aux domaines de l'avionique et de l'automobile.

2.3 Évolution des systèmes temps réel distribués

Les systèmes temps réel ont connu une évolution majeure avec l'informatisation des systèmes. Les fonctions implémentées sont ainsi de plus en plus nombreuses et complexes. La conséquence principale est une augmentation importante des données échangées entre les sous-systèmes via le réseau. Cette évolution des systèmes se retrouve dans leurs architectures. Dans la suite, nous prenons l'exemple de l'avionique.

2.3.1 Le cas de l'avionique

2.3.1.1 De l'analogique au numérique

Avant les années 60, les systèmes avioniques étaient principalement composés d'équipements analogiques embarqués dans l'avion. Ces équipements sont devenus de plus en plus complexes avec l'augmentation des besoins : augmentation du nombre de passagers, plus de confort, plus de sécurité, plus de rapidité et de robustesse.

L'informatique a révolutionné les systèmes avioniques. Des micro-ordinateurs ont été installés à bord des avions dès les années 1970/80. Depuis, on retrouve de plus en plus de systèmes numériques dans un avion. Dans la Fig. 2.2, l'évolution exponentielle du volume logiciel implémenté dans les aéronefs d'Airbus [65] est illustrée ainsi que celle des communications. Le nombre de bus est ainsi passé de quelques dizaines dans les années 1980 à plusieurs centaines au début des années 2000.

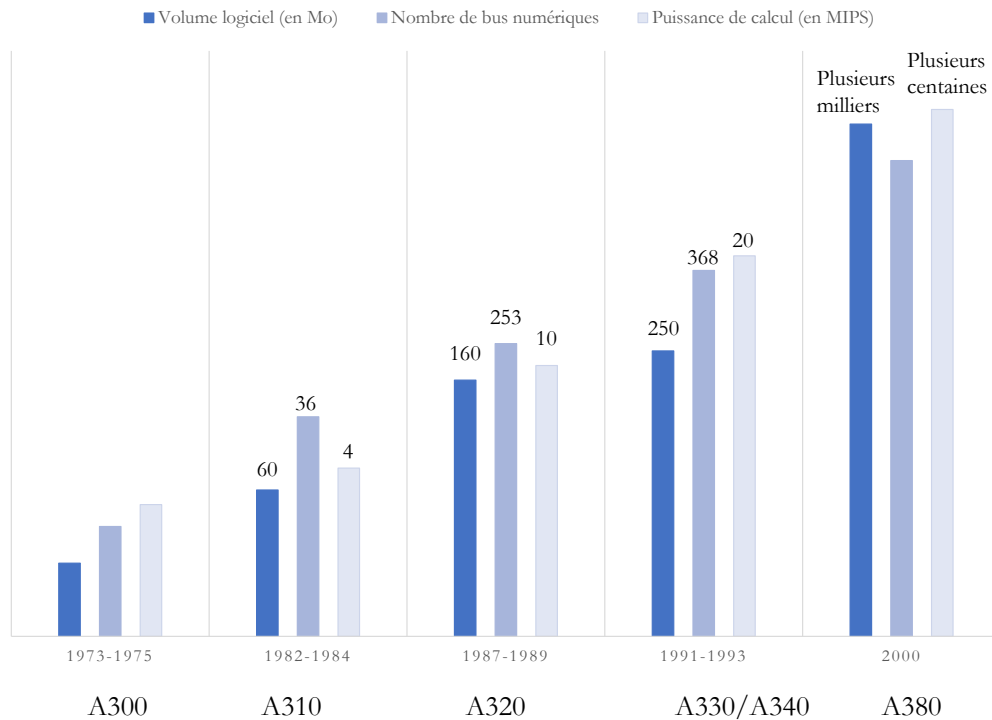


FIGURE 2.2 – L'évolution des systèmes avioniques d'Airbus.

2.3.1.2 Quelques fonctions avioniques

Les systèmes avioniques sont temps réel et distribués. Ils doivent s'assurer de l'exécution de l'ensemble des fonctions de l'avion telles que :

- la génération et la distribution électrique,
- la navigation et le pilotage automatique,
- la surveillance des portes,
- la protection incendie et fumées,
- l'éclairage,
- la génération et la distribution hydraulique,
- le train d'atterrissage,
- les services aux passagers,
- la gestion de carburant.

2.3.2 Historique des architectures des systèmes temps réel

L'évolution des besoins dans les systèmes avioniques (nouvelles fonctionnalités) et l'informatisation des systèmes ont eu des conséquences sur l'architecture des systèmes avioniques. Nous présentons les différentes architectures utilisées dans les systèmes avioniques par ordre chronologique.

2.3.2.1 Architecture centralisée

Le système est composé d'un unique calculateur pour effectuer l'ensemble des fonctionnalités du système. Le nombre de fonctions implémentées est faible. De plus, la perte

du calculateur entraîne la perte du système avionique mais n'a pas de conséquence majeure sur l'avion.

2.3.2.2 Architecture fédérée

L'architecture fédérée a pour objectif de réduire la vulnérabilité du système. Ainsi, un système organisé sous l'architecture fédérée peut être assimilé à un ensemble de sous-systèmes centralisés [50]. Ainsi, chaque sous-système est dédié à une fonction avionique. Il s'appuie sur un calculateur unique dédié et un ensemble d'équipements. Ces équipements sont placés à proximité des capteurs/actionneurs pour réduire la longueur des câbles composant le système. Par conséquent, le temps de communication et le poids sont réduits par rapport à l'architecture centralisée. De plus, un calculateur supplémentaire est chargé de contrôler les communications du système avionique dans son ensemble.

Cette architecture a été principalement utilisée dans les aéronefs militaires tels que les aéronefs de l'*US Air Force*. Dans le domaine civil, elle reste problématique car la vulnérabilité liée aux communications centralisées impacte des contraintes fortes dues à la certification requise pour obtenir l'autorisation de vol.

2.3.2.3 Architecture répartie

C'est une extension à l'architecture fédérée. Le rôle principal de l'architecture répartie est de minimiser la vulnérabilité des communications par sous-système. Les calculateurs sont des modules LRU (*Line Replaceable Unit*). Chaque équipement est doté d'un LRU qui exécute uniquement une fonction avionique.

La communication entre les équipements est basée sur des bus mono-émetteur, multi-récepteurs. Chaque LRU est responsable des données qu'il transmet. On retrouve ce type d'architecture dans les Airbus A320 et A340.

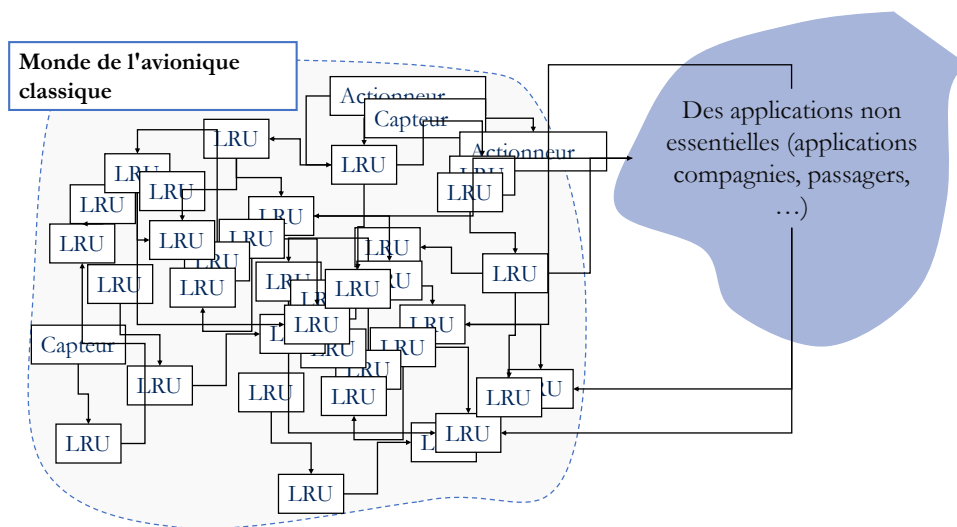


FIGURE 2.3 – Schéma d'une architecture répartie dans l'avionique.

La croissance du nombre d'applications implique une augmentation du nombre de LRUs nécessaires ainsi qu'une multiplication des liens point-à-point. La complexité du maillage des liens (cf. Fig. 2.3) entraîne des problèmes de maintenance, de flexibilité et de poids.

2.3.2.4 Conclusion

Les limites des architectures précédentes en termes de flexibilité, de complexité de maillage et d'adaptation aux futurs besoins ont poussé les avionneurs à introduire le partage de ressources entre les applications via l'architecture *Integrated Modular Avionics* ou IMA.

2.4 Le domaine avionique : *Integrated Modular Architecture*

L'architecture Modulaire Intégrée ou IMA (*Integrated Modular Architecture*) est issue des projets européens de recherche PAMELA [61] (*Process for Advanced Management of End of life Aircraft*) et VICTORIA [46] (*Validation platform for Integration of standard Components, Technologies and tools in an Open, modular and Improved Aircraft electronic system*). Elle est définie par la norme ARINC 651 [16]. L'objectif principal de l'IMA est de partager l'ensemble des ressources (processeur, réseau, etc.) entre les fonctions, tout en garantissant leurs besoins de criticité. Les systèmes avioniques sous l'architecture IMA sont découpés en plusieurs sous-systèmes distribués (*flight control system, cockpit system, etc.*). Ils sont inter-connectés via un réseau commun.

2.4.1 La partie système de l'architecture IMA

Les systèmes avioniques sont découpés en modules *Line Replaceable Module* (LRM). Un LRM est constitué d'un calculateur. Contrairement au LRU, un LRM est conçu pour ordonnancer plusieurs applications simultanément. Les LRMs sont regroupés par étagère. Chaque étagère est constituée de trois LRMs de types distincts selon la norme ARINC 651 :

- **Core Processing Module (CPM)** : ces modules sont chargés d'exécuter les fonctions avioniques. Un CPM est composé de plusieurs partitions distinctes. Selon la norme DO178B, le partitionnement fournit une isolation temporelle et spatiale contre les fautes entre des fonctions indépendantes afin de diminuer les coûts de certification. Chaque partition exécute un ensemble de fonctions avioniques ou applications. Un système d'exploitation respectant la norme ARINC 653 [17] permet la gestion des partitions et notamment l'accès aux ressources (cf. Fig. 2.4). Cette séparation entre les parties physiques et logicielles permet une meilleure intégration.
- **Input/Output Module (IOM)** : ce module est chargé de gérer l'hétérogénéité des messages échangés entre les équipements standardisés IMA et les équipements non standardisés. Le module IOM reçoit des messages depuis l'ARINC 429 et les transfère sur le réseau AFDX (ou ARINC 659), et inversement, après les avoir traités dans l'Unité de Traitement des Communications (UTC).
- **GateWay Module (GWM)** : semblable au module IOM, le module GWM est composé d'une UTC et d'une unité d'émission et de réception. Cependant, le module

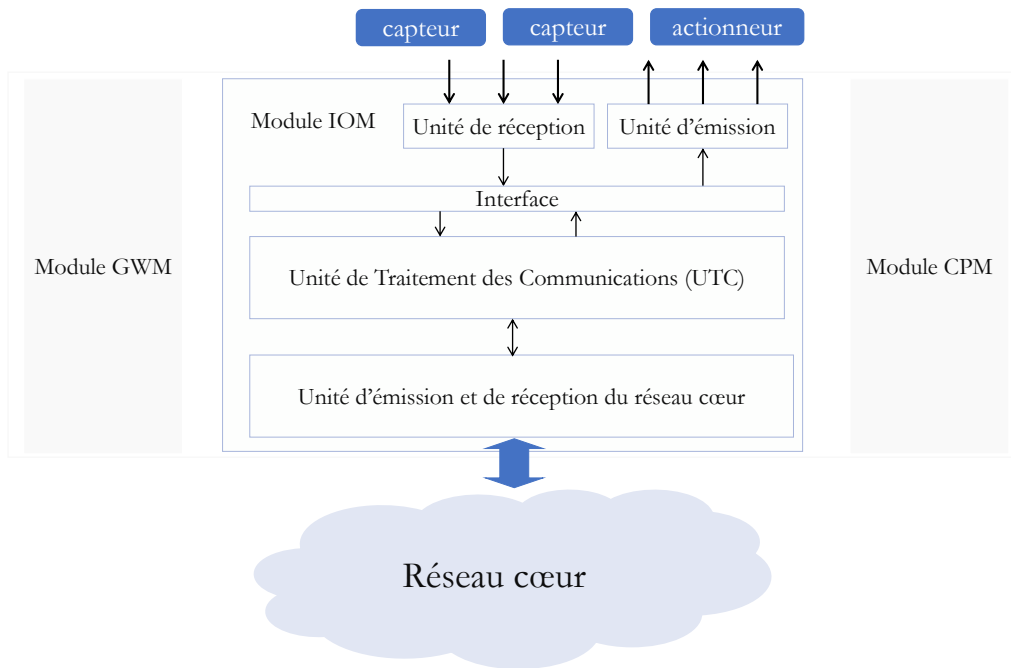


FIGURE 2.4 – Schéma d'un LRM de type CPM.

GWM établit un filtre sur les messages en provenance de l'extérieur de l'étagère selon les adresses et ne garde que celles qui doivent être traitées. Ensuite, il envoie les messages traités.

2.4.2 La partie réseau de l'architecture IMA

Le réseau qui relie l'ensemble des LRMs est hétérogène. Il comprend un réseau cœur de haute performance (*p.ex.* le réseau AFDX) inter-connecté à d'autres protocoles (*p.ex.* CAN, ARINC 429, etc.) comme illustré par la Fig. 2.5.

Le réseau embarqué dans les avions d'Airbus (A380, A350 et A400M) est nommé *Aircraft Data Communication Network* (ADCN). Par exemple, cinq types de réseaux sont utilisés dans l'ADCN de l'A380 : ARINC 429, le bus CAN, le réseau AFDX, les signaux analogiques et les signaux discrets.

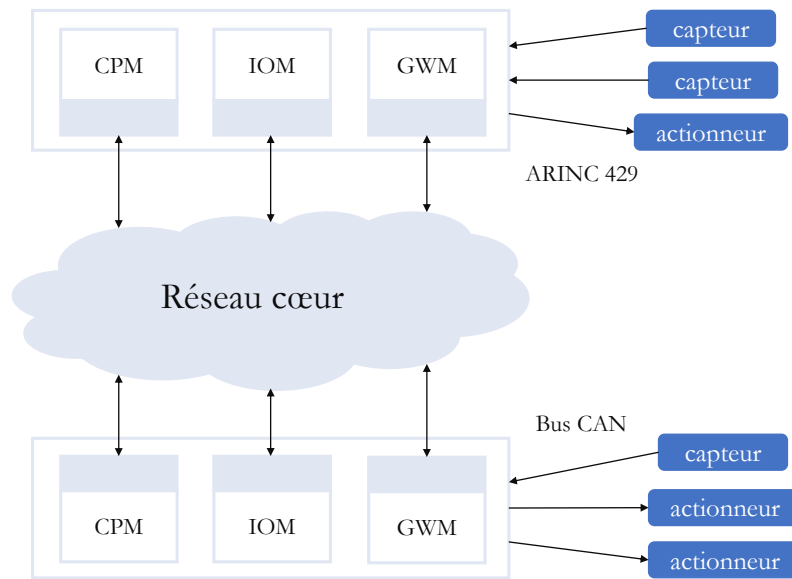


FIGURE 2.5 – Les réseaux de l'architecture IMA.

2.5 Le domaine automobile

Les systèmes temps réel de l'automobile ont connu une évolution similaire à celle des systèmes avioniques. Le nombre d'applications automobiles a augmenté permettant plus de confort et de sécurité au conducteur et surtout plus d'assistances.

Les systèmes temps réel dans l'automobile sont composés de plusieurs sous-systèmes via une architecture distribuée. Cependant, beaucoup de travaux ont été menés dans le but d'utiliser un seul réseau cœur basé sur le standard Ethernet pour réduire la complexité des systèmes automobiles [55].

2.5.1 Les systèmes temps réel de l'automobile

Les systèmes temps réel de l'automobile sont composés d'un ensemble de dispositifs électroniques, principalement des unités de contrôle électronique (ECU ou *Electronic Control Unit*), et logiques en relation avec les parties mécaniques et hydrauliques de la voiture [97].

Une ECU est un système embarqué composé d'un micro-contrôleur avec des capteurs et des actionneurs. Une ECU récupère les données des capteurs qui sont placés dans différentes parties du procédé et contrôle les différents équipements via les actionneurs.

La Fig. 2.6 représente plusieurs ECUs installées à proximité des éléments concernés. Par exemple, l'ECU chargée du réglage électronique du volant est placée à proximité du volant.

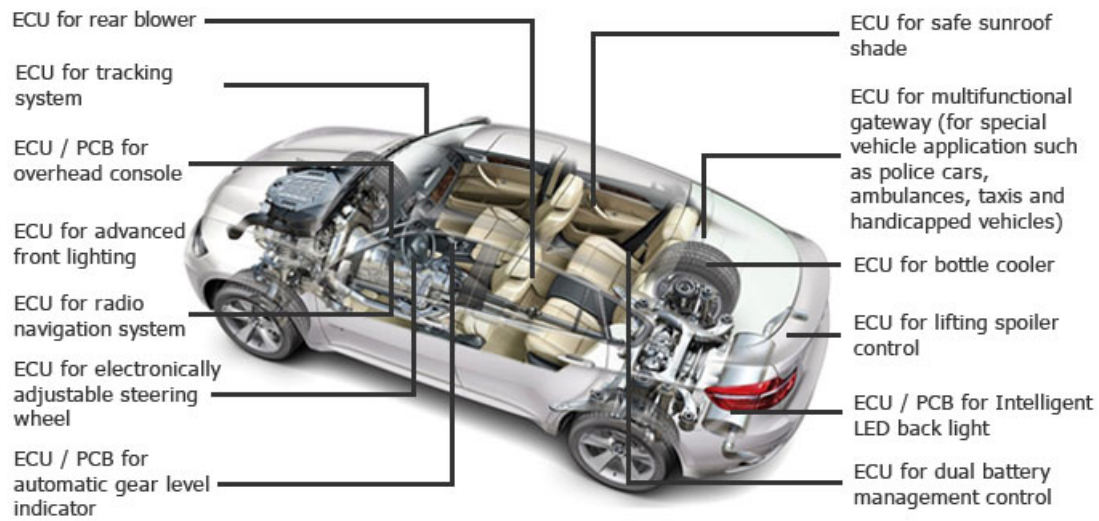


FIGURE 2.6 – Exemple de systèmes automobiles avec des ECUs [101].

2.5.2 Les sous-systèmes d'un véhicule

Les fonctions automobiles sont regroupées selon leur domaine : divertissement, *powertrain*, etc. Ces domaines sont composés de plusieurs ECUs reliées par un réseau dédié. Les fonctions de l'automobile sont regroupées en sous-systèmes. Nous en citons quelques-uns :

- **Chassis** : il assure la stabilité et la dynamique du véhicule. Il inclut plusieurs programmes tels que le *Anti-lock Brake System* (ABS).
- **Powertrain** : il contrôle le moteur et la boîte de vitesses (transmission de la puissance du moteur lors du changement de vitesse). Il est chargé également de la coordination de l'injection du carburant.
- **Les systèmes X-by-wire** : sont chargés de remplacer les fonctions mécaniques et hydrauliques à l'aide de systèmes de contrôle dans le but de les automatiser. Parmi les fonctions X-by-wire, on retrouve le freinage (en anglais *brake-by-wire*), le changement de vitesse (*shift-by-wire*), etc.
- **L'assistance au conducteur (Driver Assistance)** : regroupe les fonctions d'aide à la conduite. Il doit assurer la sécurité et diminuer les tâches à effectuer pour le conducteur. A titre d'exemple : régulateur de vitesse, détecteur de pluie pour déclencher les essuie-glaces automatiquement, détecteur de dépassement de voie, etc.
- **Multimédia et système d'infotainment**¹ : ce système n'assure pas de fonctionnalité primaire du véhicule. Il inclut les programmes audio/vidéo, le GPS, le réglage de sièges arrières, la connexion Internet, Wi-Fi, etc. Il permet également d'informer constamment le conducteur sur le trafic, l'état de la voiture, etc.
- **Diagnostic** : établit des vérifications/diagnostic sur le véhicule au niveau des composants matériels et logiciels à des fins de maintenance et de mises à jour.

1. le terme *infotainment* est une fusion entre *information* et *entertainment* en anglais qui veut dire divertissement.

2.5.3 Les systèmes automobiles modernes

Les véhicules actuels disposent de plus de 50 ECUs (jusqu'à 100 ECUs pour les voitures de luxe [97]). L'évolution du nombre d'ECUs utilisées dans les sous-systèmes automobiles est croissante. L'informatisation des fonctions automobiles et, surtout, l'augmentation du nombre de consommateurs ayant des préférences pour les voitures sophistiquées ont renforcé le marché des ECUs. Ainsi, la taille du marché des ECUs dans le domaine de l'automobile a atteint environ 33 Milliards de Dollars US en 2025 [88]. Par exemple, la Fig. 2.7 illustre l'évolution de l'investissement dans les ECUS des sous-systèmes automobiles en Chine tels que l'*Advanced Driver Assistance System* (ADAS) et l'*infotainment*.

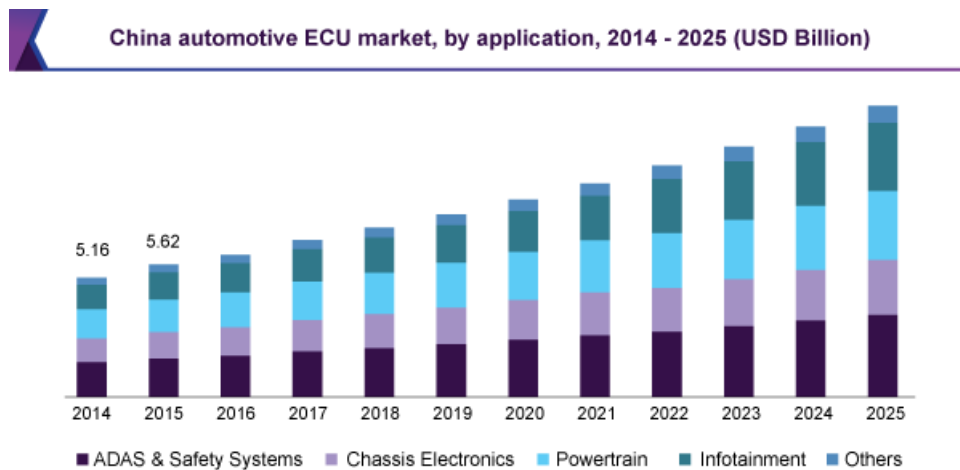


FIGURE 2.7 – L'évolution du marché chinois des ECUs dans le domaine de l'automobile [88].

2.5.4 L'architecture de communications des systèmes automobiles

Les sous-systèmes automobiles ont des besoins différents en communication. Dans [26], l'auteur propose une table de correspondance entre les sous-systèmes et leurs besoins. Par exemple, les besoins des sous-système châssis et *powertrain* sont exprimés en termes de tolérance aux pannes et de déterminisme contrairement au sous-système de diagnostic qui a un besoin plus important en terme de flexibilité à cause de la quantité de données échangées.

Plusieurs technologies de communication sont utilisées pour les différents sous-systèmes dans le but de répondre aux différents besoins (cf. Fig. 2.8). Des réseaux moins performants peuvent, ainsi, être utilisés dans des sous-systèmes ne nécessitant pas des besoins importants en performance ou en tolérance aux pannes. Par exemple, le réseau LIN est utilisé dans le sous-système qui contrôle les fenêtres et le verrouillage des portes.

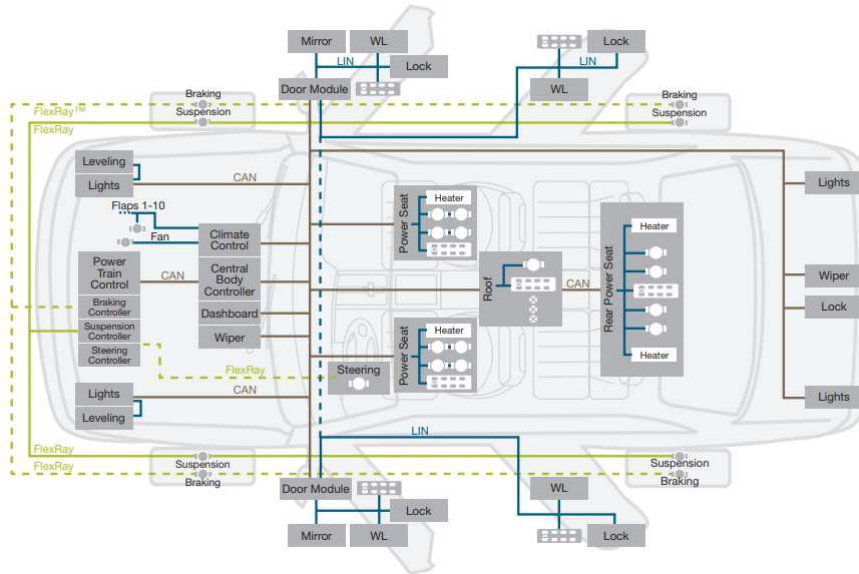


FIGURE 2.8 – Les systèmes de communications automobiles [47].

2.6 Conclusion

L'avènement des systèmes numériques a révolutionné les domaines de l'avionique et de l'automobile. La numérisation des fonctions a créé une nouvelle génération de systèmes informatiques temps réel avec de plus en plus de besoins (connectivité, sécurité, robustesse, confort, etc.). Par conséquent, l'architecture des systèmes a connu une évolution importante, de l'architecture centralisée à l'architecture distribuée. L'architecture distribuée est désormais la plus utilisée dans les systèmes avioniques et automobiles. Elle permet l'échange de données et le partage de ressources entre plusieurs sous-systèmes composant le système temps réel distribué. Le réseau informatique entre ces sous-systèmes est une composante importante de l'architecture distribuée. Par conséquent, l'évolution des besoins des systèmes temps réel distribués a entraîné une évolution des besoins en réseau (bande passante élevée, sûreté, flexibilité).

Dans le chapitre suivant, nous présentons les réseaux temps réel ainsi que leur évolution. Nous nous focalisons sur les réseaux temps réel embarqués dans les systèmes avioniques et automobiles.

Chapitre 3

Les réseaux temps réel

Sommaire

3.1 Introduction	23
3.2 Réseau informatique	23
3.2.1 Définition	23
3.2.2 Les types de réseaux selon la distance	23
3.2.3 Les infrastructures	24
3.2.4 Les topologies	24
3.2.5 Le modèle OSI	27
3.2.6 Les politiques de service	29
3.3 Évolution des réseaux temps réel	29
3.3.1 Le réseau temps réel	29
3.3.2 Des exemples de réseaux temps réel	29
3.4 Le réseau Ethernet dans le temps réel	32
3.4.1 Le réseau Ethernet	32
3.4.2 Les principaux protocoles de l'Ethernet temps réel	33
3.5 Avionic Full Duplex Switched Ethernet (AFDX)	33
3.5.1 Définition générale	33
3.5.2 Virtual Link (VL)	34
3.5.3 Le End System (ES)	35
3.5.4 Le commutateur AFDX	35
3.5.5 Exemple industriel	37
3.6 Audio/Vidéo Bridging Switched Ethernet (AVB)	39
3.6.1 Définition du réseau AVB	39
3.6.2 Les composants du réseau AVB	39
3.6.3 Les propriétés du réseau AVB	39
3.6.4 Le protocole Time Sensitive Networking	41
3.7 Conclusion	42

3.1 Introduction

Dans un système temps réel distribué, le support de communication ou réseau est fondamental. Dans ce chapitre, nous définissons le réseau informatique. Ensuite, nous présentons les propriétés des réseaux temps réel embarqués et citons quelques réseaux. Pour finir, nous détaillerons les réseaux étudiés durant cette thèse : l'*Avionic Full Duplex Switched Ethernet* (AFDX) et l'*Audio Video Bridging Ethernet* (AVB).

3.2 Réseau informatique

3.2.1 Définition

Un réseau informatique est constitué de l'ensemble des éléments électroniques (ordinateur, imprimante, système embarqué, etc.) reliés entre eux à travers des infrastructures physiques (filaire ou non filaire). Le but principal d'un réseau est d'assurer l'échange d'informations entre ces équipements (ordinateurs, imprimantes et fax dans la Fig. 3.1).

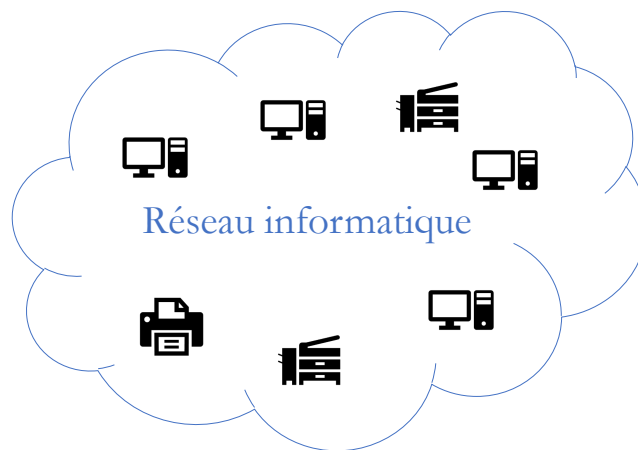


FIGURE 3.1 – Schéma d'un réseau informatique.

3.2.2 Les types de réseaux selon la distance

Plusieurs critères permettent de différencier les réseaux (distance, débit, topologies, infrastructures, etc.). Ainsi, les réseaux sont classifiés selon la distance entre les terminaux interconnectés (cf. Fig. 3.2) :

- **Les réseaux locaux personnels (Personnal Area Network ou PAN)** : ils relient des appareils personnels à l'échelle de quelques mètres.
- **Les réseaux locaux (Local Area Network ou LAN)** : réseaux à l'échelle d'une pièce ou d'un bâtiment.
- **Les réseaux métropolitains (Metropolitan Area Network ou MAN)** : réseaux à l'échelle d'une ville.
- **Les réseaux étendu (Wide Area Network ou WAN)** : réseaux reliant des terminaux distants à l'échelle de plusieurs centaines de kilomètres.

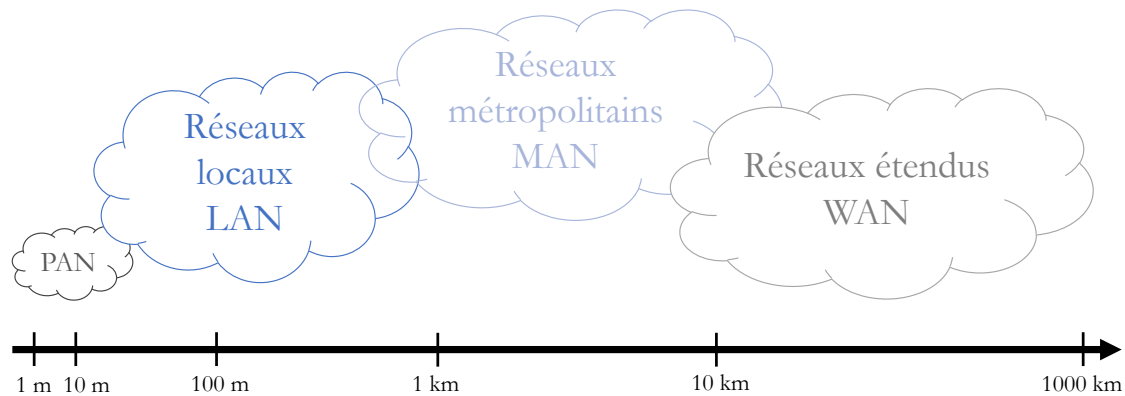


FIGURE 3.2 – Les types de réseaux selon la distance.

3.2.3 Les infrastructures

L'infrastructure définit le support physique reliant les terminaux appartenant à un réseau. C'est le support de transmission des données échangées entre les terminaux. Ces infrastructures peuvent être classées comme suit :

- **Les infrastructures filaires** : les données sont transmises à travers des câbles tels que le câble coaxial ou les paires torsadées qui envoient des signaux électriques à travers un fil de cuivre. Par exemple, le réseau Ethernet est un réseau filaire.
- **Les infrastructures sans fil** : les données sont transmises à travers des ondes radio. Le réseau GSM est un réseau sans fil.
- **La fibre optique** : les données sont transmises à travers des ondes lumineuses comme le réseau ATM.

3.2.4 Les topologies

La topologie d'un réseau définit la configuration selon laquelle les terminaux (ou équipements informatiques) sont reliés via le réseau.

1. **La topologie en anneau** : tous les terminaux sont reliés en chaîne les uns aux autres. Deux terminaux qui s'échangent des informations doivent les envoyer à travers les terminaux intermédiaires (cf. Fig. 3.3) dans un sens unique. L'avantage de cette topologie est qu'elle ne nécessite que peu de câbles (pour N terminaux, N liaisons sont suffisantes). Les collisions sont évitées à l'aide de techniques telles que la détection de jeton. En revanche, la panne d'un seul équipement rompt tout le réseau. Un exemple du réseau en anneau est le *Token ring*.
2. **La topologie en bus** : les terminaux sont reliés par un câble unique appelé *bus* (cf. Fig. 3.4). La défaillance d'un équipement n'entraîne pas de dysfonctionnement global du réseau. Un bus peut émettre des informations dans les deux sens. L'inconvénient majeur de cette topologie est que tout le réseau repose sur un seul segment. Il ne gère pas les collisions. Le réseau CAN est un réseau en bus [2].
3. **La topologie en étoile (ou *Hub and spoke*)** : tous les terminaux du réseau sont reliés à un élément central (un concentrateur ou un commutateur) (cf. Fig. 3.5). Si cet élément tombe en panne, le réseau ne fonctionne plus. Par contre la défaillance d'une machine n'affecte pas le réseau. Le réseau Ethernet commuté est un réseau en étoile [37].

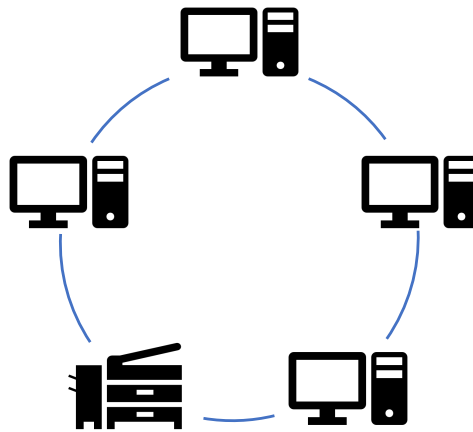


FIGURE 3.3 – La topologie en anneau.

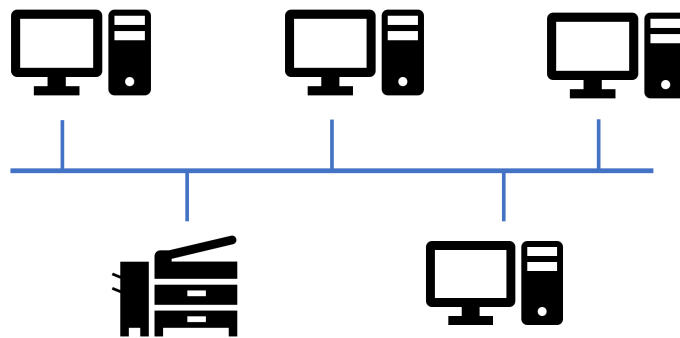


FIGURE 3.4 – La topologie en bus.

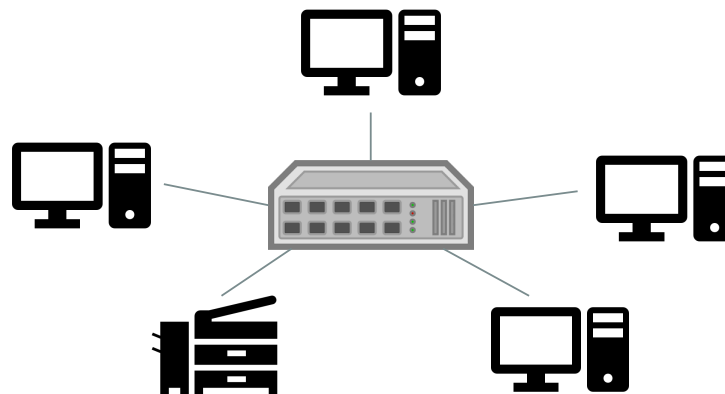


FIGURE 3.5 – La topologie en étoile.

4. **La topologie en arbre :** dans cette topologie, le réseau est hiérarchisé (la machine au sommet est reliée directement aux machines du niveau inférieur et ainsi de suite) (cf. Fig. 3.6). Si une machine à un noeud de l'arborescence est en panne, les deux branches du réseau deviennent indépendantes et ne peuvent plus communiquer entre elles.

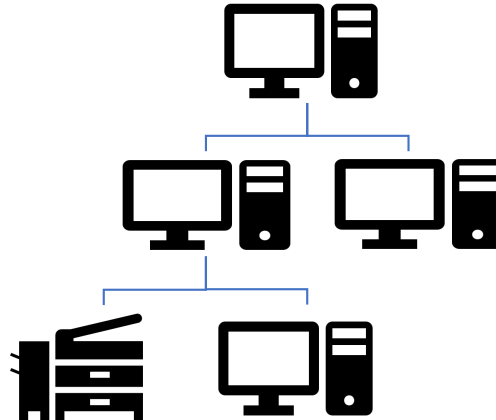


FIGURE 3.6 – La topologie en arbre.

5. **Le réseau maillé :** chaque machine peut être reliée à une ou plusieurs machines (cf. Fig. 3.7). S'il y a N machines dans le réseau, on peut atteindre jusqu'à $\frac{N \cdot (N-1)}{2}$ liaisons. Si une ligne est coupée, les communications ne sont pas perturbées. Par contre, le maillage peut être problématique lorsque le nombre de nœuds devient grand. Un exemple de réseau maillé est le réseau Internet [58].

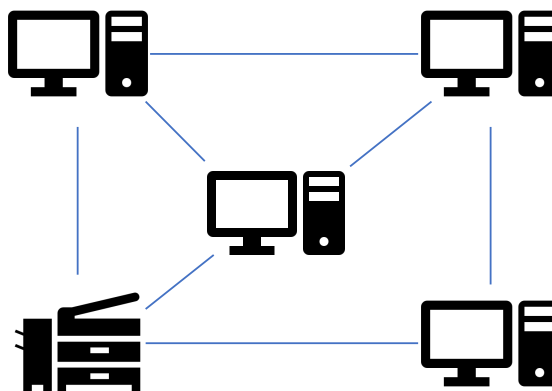


FIGURE 3.7 – La topologie maillée.

6. **La topologie linéaire :** les machines sont reliées en chaîne mais le réseau n'est pas bouclé comme en topologie en anneau (cf. Fig. 3.8). La défaillance d'une machine divise le réseau en deux sous-réseaux indépendants.

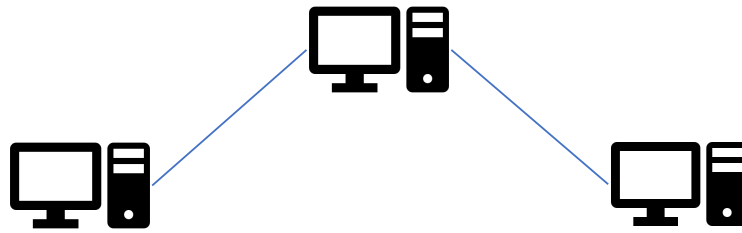


FIGURE 3.8 – La topologie linéaire.

3.2.5 Le modèle OSI

La communication entre deux équipements reliés par un réseau a été standardisée par le modèle **OSI** (*Open Systems Interconnection*) [107] proposé par l'ISO (Organisation Internationale de Normalisation). Il est défini par une succession de couches allant de la couche physique à la couche application (cf. Fig. 3.9). Chaque couche correspond à un niveau d'abstraction. La couche physique est au plus bas du niveau d'abstraction car elle est en lien direct avec le support de transmission.

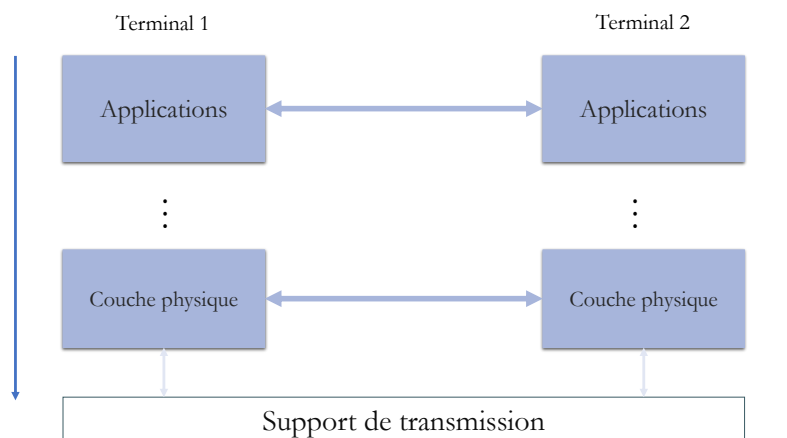


FIGURE 3.9 – Le modèle en couches.

3.2.5.1 Services et protocoles

Les couches communiquent entre elles à travers des services et des protocoles. Un service est l'ensemble de primitives fournies par une couche à sa couche voisine supérieure. Une primitive est une commande qui peut être une demande de service, une réponse à une demande ou une notification à un événement.

Le protocole, quant à lui, est le langage utilisé pour réaliser un service. Il est l'ensemble des règles (*p.ex.* format des messages) qui permet la réalisation des opérations fournies par une couche. Deux couches du même niveau doivent avoir des protocoles compatibles afin de pouvoir communiquer. Par exemple, dans la Fig. 3.10, la couche 2 du terminal 1 communique avec la couche 2 du terminal 2 à travers les protocoles prédéfinis de la couche 2.

Le passage d'une couche à une autre permet d'en-capsuler, à l'aide de protocoles, la don-

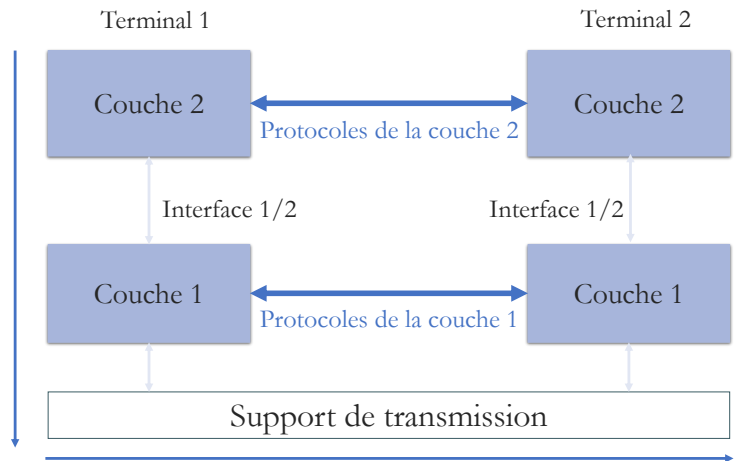


FIGURE 3.10 – Le dialogue entre les couches.

née destinée à être transmise sur le réseau avec les informations nécessaires pour assurer sa transmission : adresse source, adresse destination, priorité, etc. Le **paquet** est la structure résultante de cette encapsulation. Chaque couche rajoute un entête au message à l'aide des protocoles et des services qui lui sont fournis à son niveau et l'envoie à la couche sous-jacente à travers une interface (Interface 1/2 entre la couche 2 et la couche 1 dans la Fig. 3.10).

3.2.5.2 Les couches du modèle OSI

Le modèle *Open Systems Interconnection* (OSI) a été conçu par l'Organisation Internationale de Normalisation (ISO). Il comporte sept couches différentes :

1. **La couche physique** : établit les règles de conversion des données en signaux numériques.
2. **La couche liaison de données** : également appelée la couche **MAC** (*Medium Access Control*) détermine l'interface avec la carte réseau.
3. **La couche réseau** : assure le routage des informations (destinataire, chemin).
4. **La couche transport** : contrôle les informations et corrige les erreurs. Elle assure un transfert fiable des informations.
5. **La couche session** : établit et coordonne les communications.
6. **La couche présentation** : gère la mise en forme des données (cryptage, compression, etc.).
7. **La couche application** : gère le transfert de données entre les programmes.

De la septième couche à la première, l'information destinée à être échangée avec un autre terminal est en-capsulée à chaque niveau (couche) sous forme de **paquet**. Elle est composée d'un entête et d'une fin de message. Elle comporte les informations établies par les protocoles de chaque couche. La dernière trame (ou paquet au niveau de la couche physique) est envoyée sur le réseau à travers le support de transmission.

Dans nos travaux, nous nous intéressons à la couche 2 ou MAC du modèle OSI.

3.2.6 Les politiques de service

Les terminaux échangent des données en-capsulées sous forme de trames au niveau de la couche 2 du modèle OSI. Ces trames sont transférées sur le réseau. Lorsque plusieurs trames arrivent simultanément (par différents liens) au même point appelé *point de multiplexage*, une politique de service doit être mise en place pour décider de la trame à transmettre en premier.

Il existe plusieurs politiques d'ordonnancement qui ont été définies dans la littérature. Dans cette thèse, nous abordons les deux politiques suivantes :

- la politique *First-In-First-Out* ou FIFO : la première trame arrivée est la première à être transmise ;
- la politique *Fixed Priority/First-In-First-Out* (FP/FIFO) : chaque trame possède un niveau de priorité. La trame la plus prioritaire est transmise en premier. Si plusieurs trames sont présentes avec un même niveau de priorité, elles sont départagées par la politique FIFO.

Une caractéristique importante d'une politique de service est l'oisiveté :

Définition 1 *Une politique d'ordonnancement qui n'insère pas de temps creux lorsqu'elle a des trames prêtes à être transmises est dite conservative ou non-oisive. Dans le cas contraire, la politique est qualifiée d'oisive.*

3.3 Évolution des réseaux temps réel

Les réseaux utilisés dans le contexte temps réel sont multiples. Ils ont évolué afin de répondre aux besoins toujours plus importants. Nous définissons les propriétés d'un réseau temps réel. Ensuite, nous traçons l'historique des réseaux temps réel, des connexions point-à-point au réseau Ethernet commuté.

3.3.1 Le réseau temps réel

Un réseau est dit temps réel si les propriétés suivantes sont respectées [76] :

Déterministe : un réseau est dit déterministe s'il garantit la transmission de l'ensemble des données (sans perte) dans un temps fixé, sous l'hypothèse d'un fonctionnement nominal (hors panne).

Statique : la configuration du réseau est statique (caractéristiques des trames, chemins empruntés). Tout est stocké en statique dans des tables de configurations.

Sûre : les messages peuvent être erronés. La gestion des erreurs doit donc être garantie afin d'éviter qu'elles se propagent dans le réseau.

Tolérant aux pannes : le réseau doit garantir une disponibilité continue et trouver des alternatives à toute défaillance potentielle.

3.3.2 Des exemples de réseaux temps réel

3.3.2.1 Le ARINC 429

Le réseau ARINC 429 est un réseau point-à-point [18]. Il est utilisé dans le domaine avionique car il garantit la sûreté. Le ARINC 429 est un lien mono-émetteur et multides-

tinataires avec un débit faible variant de 12,5 à 100kbps. Chaque donnée est affectée à un lien ARINC 429. Par exemple, dans la Fig. 3.11, le nœud 1 est connecté à trois autres nœuds (2, 3 et 4) par un seul lien à trois embranchements. Il est simple et facile à mettre en œuvre. En revanche, il y a autant de flux que de liaisons. Sa flexibilité est ainsi réduite.

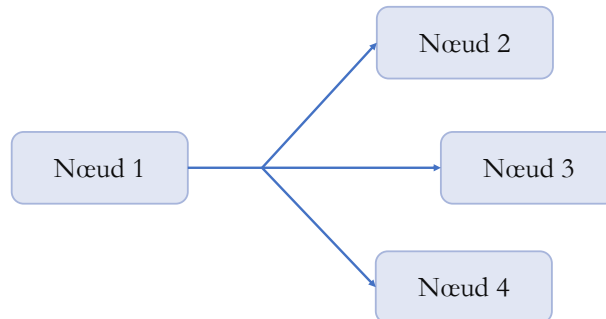


FIGURE 3.11 – Schéma d'un réseau ARINC 429.

3.3.2.2 Le bus CAN

Le bus *Controler Area Network* **CAN** a été développé par Bosch pour les systèmes temps réel automobiles. Il a été standardisé avec la norme ISO 118981 (Organisation Internationale de Normalisation) [2]. C'est un bus multiplexé de 1 Mbps. Il permet de relier plusieurs nœuds comme illustré dans la Fig. 3.12. Chaque nœud peut envoyer et recevoir des messages à travers le bus CAN. Il y a risque de collision entre les messages. Par exemple, si les nœuds 1 et 3 envoient un message simultanément au nœud 2, leurs messages entreront alors en collision.

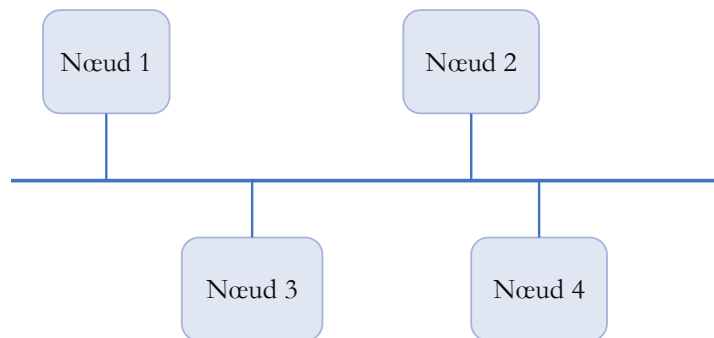


FIGURE 3.12 – Schéma d'un réseau CAN.

Pour gérer les collisions, le bus CAN utilise le protocole *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) et le protocole *Time Division Multiple Access* (TDMA).

- **Le protocole CSMA/CA** : il permet d'affecter une priorité à chaque message. A chaque instant, lorsque plusieurs équipements tentent d'envoyer un message, le médium est réservé au message le plus prioritaire.
- **Le protocole TDMA** : il affecte un intervalle de temps statique à chaque équipement pour la transmission de messages émis depuis cet équipement.

Aujourd'hui c'est le réseau le plus utilisé par un grand nombre d'industriels notamment dans l'avionique et l'automobile car c'est un réseau bon marché. En effet, en utilisant des bus multiplexés dans les portes, BMW a réussi à réduire le poids de 15 kg. En revanche, son principal inconvénient est sa bande passante limitée.

3.3.2.3 Autres exemples de protocoles

D'autres réseaux sont utilisés dans les systèmes temps réel. Par exemple, dans les systèmes automobiles [64] :

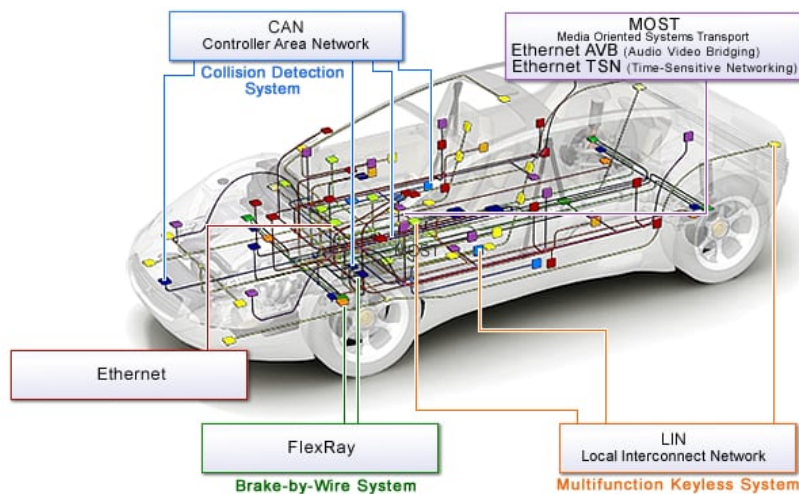


FIGURE 3.13 – Les systèmes de communications automobiles [1].

- **Local Interconnect Network (LIN)** [106] : C'est un réseau maître/esclave de type LAN, développé par un consortium d'industriels automobiles (notamment Audi et BMW). Il est moins cher que CAN et suffisamment rapide (20kbps) pour être utilisé dans certaines parties du véhicule. Il est utilisé, par exemple, dans les portes du véhicule pour gérer le verrouillage et l'ouverture des fenêtres. Les réseaux LIN sont interconnectés par des liens CAN à travers des passerelles CAN/LIN.
- **FlexRay** [105] : C'est un réseau défini par le standard ISO 17458. Il étend le réseau *Byteflight* en introduisant des fenêtres statiques pour l'ordonnancement de messages dits *Time-triggered*. Ces messages sont synchronisés et doivent être transmis dans des fenêtres prédéfinies par un ordonnancement hors-ligne. Le réseau Flexray offre une bande passante élevée de 10Mbps. Il est généralement employé pour des fonctions à très haute criticité telles que *X-by-Wire*. Mais il est très cher ce qui restreint son utilisation.
- **Media Oriented Systems Transport (MOST)** [53] : C'est un réseau conçu pour les applications multimédia (audio, vidéo). Sa bande passante varie de 25 à 150Mbps.

Chaque protocole est dédié à un ensemble de sous-systèmes selon leurs besoins. Cette hétérogénéité complexifie l'intégration de nouvelles applications et définit une solution coûteuse.

3.4 Le réseau Ethernet dans le temps réel

Depuis les années 90, les systèmes temps réel ne cessent d'évoluer. Les principaux besoins sont : bande passante élevée, flexibilité, optimisation de maillage et gain de poids. Les protocoles cités paragraphe 3.3.2 sont toujours utilisés mais ne peuvent pas représenter le réseau principal. Le réseau Ethernet est une alternative car il bénéficie de plusieurs avantages et répond aux besoins des systèmes temps réel distribués actuels [13, 82, 94].

3.4.1 Le réseau Ethernet

Le réseau Ethernet est standardisé par la norme IEEE 802.3 [9]. C'est un standard connu, reconnu et à faible coût. De plus, il offre un débit élevé (10Mbps, 100Mbps voir 1 Gbps) et tous les outils ont déjà été développés et sont accessibles.

Ethernet commuté a une topologie en étoile. Il permet de réduire le maillage et offre un gain de poids. Dans un réseau Ethernet commuté, les terminaux sont reliés par des commutateurs et des liens physiques (cf. Fig. 3.14). Un commutateur, ou *Switch* en anglais, gère le multiplexage des données (réception, traitement et retransmission).

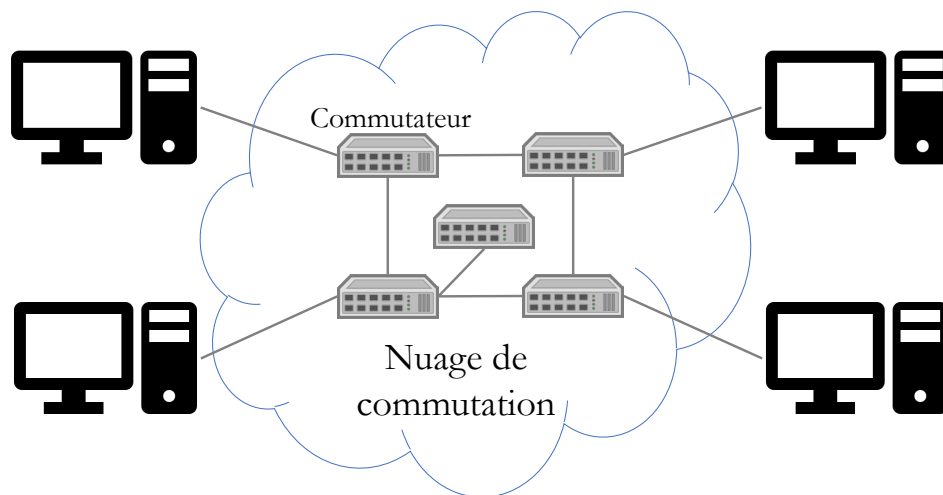


FIGURE 3.14 – Schéma d'un réseau Ethernet commuté.

La taille d'une trame Ethernet varie entre 64 et 1518 octets. Elle est divisée en plusieurs champs (cf. Fig. 3.15).

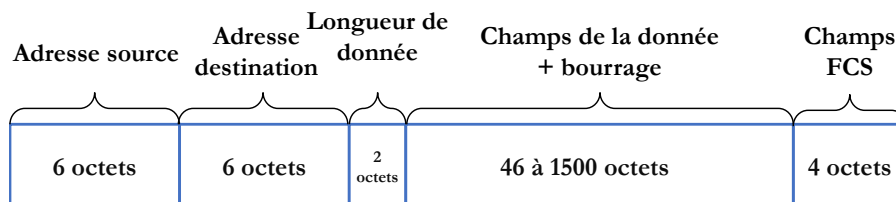


FIGURE 3.15 – La structure d'une trame Ethernet

- L'adresse source (respectivement l'adresse destination) : c'est l'adresse MAC du terminal source (respectivement terminal destination).
- La taille des données : donne la longueur du champ des données.

- Le champ des données : contient la donnée à transmettre. Le bourrage est un espace qui permet de compléter la donnée si le champ de la donnée n'est pas suffisant (46 octets).
- Le champ *Frame Check Sequence* (FCS) : le terminal source effectue un calcul appelé *Cyclic Redundancy Checking* (CRC) sur les champs adresse source, adresse destination, longueur de donnée et données. Elle enregistre le résultat dans le champ FCS. Le terminal destinataire effectue le même calcul et compare ses résultats avec le champ FCS. Une incohérence des résultats indique l'existence d'une erreur dans la trame. Par exemple : longueur de trames incorrecte ou destination non identifiée.

Cependant, le standard Ethernet doit être adapté aux systèmes temps réel distribués car il ne répond pas aux besoins de ce genre de systèmes en communication. Le réseau utilisé dans les systèmes temps réel de nouvelle génération doit garantir le respect des contraintes temps réel. Plusieurs protocoles de type Ethernet ont été développés dans le contexte temps réel.

3.4.2 Les principaux protocoles de l'Ethernet temps réel

Les trois principaux protocoles réseau temps réel basés sur Ethernet sont :

- **Le réseau *Avionics Full-duplex Switched Ethernet* (AFDX) [3]** : a été développé pour les systèmes avioniques de nouvelle génération par Airbus. Il est utilisé comme le réseau coeur de l'architecture IMA.
- **Le réseau *Time Triggered Ethernet* (TTE) [19]** : est un standard temps réel qui permet d'utiliser trois types de flux : *Time triggered* (TT), *Rate-constrained* (RC) et *Best-effort* (BE). Le premier type de flux suit un ordonnancement hors-ligne sous la politique *Time Division Multiple Access* (TDMA). Les flux RC sont asynchrones et contraints en bande passante. La classe BE est dédiée aux messages ne nécessitant pas une bande passante élevée et sans contraintes.
- **Le réseau *Audio/Vidéo Bridging Ethernet* (AVB) [6]** : est un ensemble de standards visant à transmettre des messages multimédia avec des priorités multiples. Sa politique de service permet de limiter l'impact des flux plus prioritaires. Ses performances ont motivé son utilisation pour des messages temps réel. Il est amené à être étendu avec le standard **TSN** (*Time Sensitive Network*) [83]. TSN est un protocole basé sur le réseau AVB avec l'utilisation du protocole TTE pour les flux critiques [81].

Dans cette thèse, nous nous sommes focalisés sur le réseau *Avionic Full Duplex Switched Ethernet* (AFDX) et le réseau *Audio/Vidéo Bridging Switched Ethernet* (AVB) que nous détaillons dans les prochains paragraphes.

3.5 *Avionic Full Duplex Switched Ethernet* (AFDX)

3.5.1 Définition générale

L'*Avionic Full Duplex Switched Ethernet* a été standardisé par l'ARINC 664 (partie 7) [3] et est destiné aux systèmes avioniques. Les entrées/sorties de l'AFDX sont modélisés par

des *End Systems* ou **ESs** qui s'échangent des données par l'intermédiaire d'un ensemble de commutateurs et de liens physiques. Les liens physiques sont bidirectionnels et de type *full duplex* garantissant ainsi qu'il n'y a aucune collision possible. Le débit des liens est de 10Mbps ou 100Mbps. Les flux de données du réseau AFDX sont modélisés par des *liens virtuels*, ou *Virtual Link (VL)* en anglais.

La Fig. 3.16 représente un exemple de configuration AFDX composée de 5 flux, 6 ESs et 4 commutateurs (ou *switchs*).

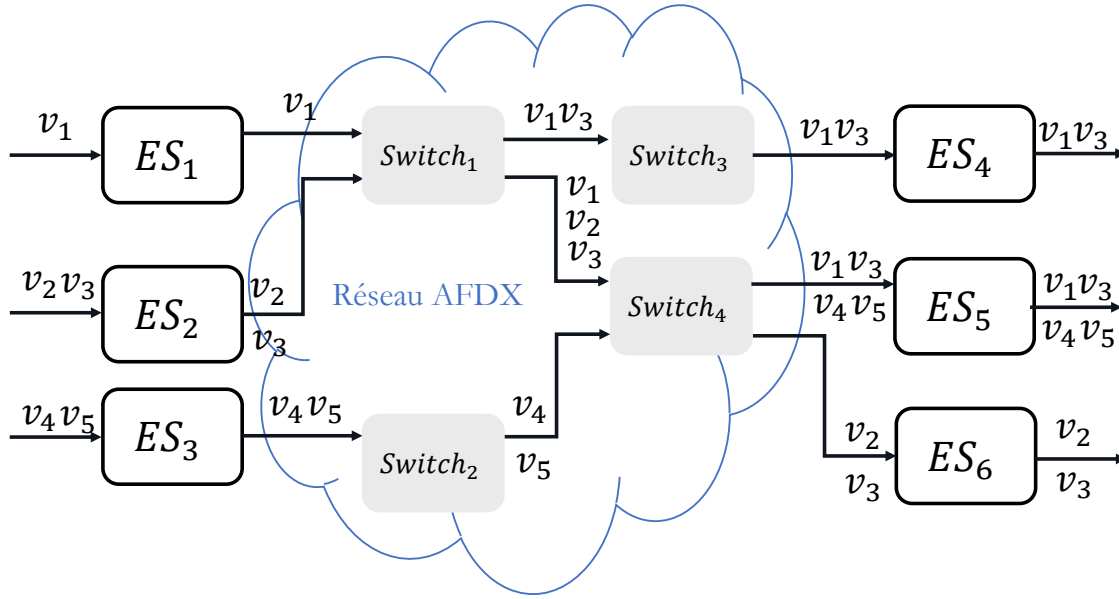


FIGURE 3.16 – Schéma de la configuration avionique du réseau illustré dans la Fig. 2.5.

Il n'y a pas de synchronisation entre les composants du réseau AFDX (ESs et commutateurs). Plusieurs trames peuvent ainsi provenir de plusieurs liens d'entrée et interférer dans un point de multiplexage simultanément.

3.5.2 Virtual Link (VL)

Chaque flux est affecté à un **VL**. Un VL est un lien unidirectionnel et *multicast* (un ou plusieurs destinataires). Il est défini par un chemin statique comprenant la liste ordonnée des nœuds traversés du ES source aux ESs destinataires. Par exemple, dans la Fig. 3.16, le ES2 émet un flux à destination de trois ESs ES4, ES5 et ES6 via les commutateurs *Switch1*, *Switch3* et *Switch4*. Le flux v_3 est multicast.

Un flux VL est caractérisé par :

- un identifiant unique ID ;
- la taille maximale d'une trame générée par le VL, **Fmax** ;
- la taille minimale d'une trame générée par le VL, **Fmin** ;
- et un **Bandwidth Allocation Gap** ou **BAG** : le délai minimum entre l'émission de deux trames consécutives par le flux.

La transmission d'une trame de taille maximale tous les BAGs (cf. Fig. 3.17) définit le contrat de trafic garanti au VL à son entrée sur le réseau. Le débit d'un VL est donc contraint par $\frac{F_{max}}{BAG}$.

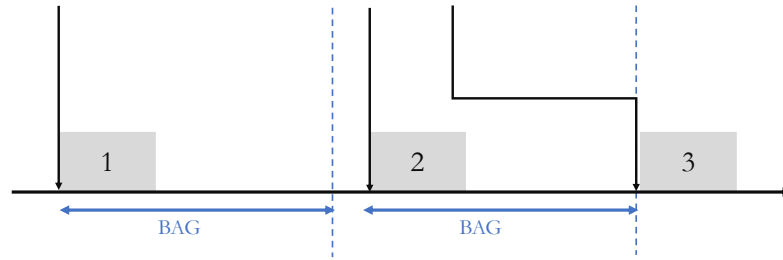


FIGURE 3.17 – Régulation de l'émission de trames d'un flux.

Par exemple, dans Fig. 3.17, la trame 3 est générée un BAG après la trame 2.

3.5.3 Le End System (ES)

Les ESs représentent l'interface d'accès au réseau pour les calculateurs. Il définit les points d'entrée/sortie du réseau AFDX. Il encapsule les données produites par les applications sous formes de trames pour les émettre dans le réseau AFDX à travers son port de sortie (cf. Fig. 3.18). Il reçoit les trames à partir du réseau AFDX à travers son port d'entrée pour les dés-encapsuler et les envoyer aux applications.

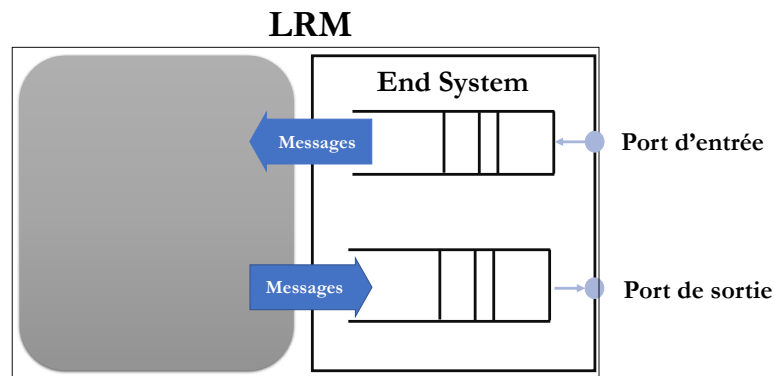


FIGURE 3.18 – Schéma d'un ES.

3.5.4 Le commutateur AFDX

Le commutateur est l'élément central de l'AFDX. Il est composé d'un ensemble de ports d'entrée et d'un ensemble de ports de sortie (cf. Fig. 3.19). Le commutateur récupère les trames en mode *store & forward* sur les ports d'entrée (il attend d'avoir complètement reçu une trame pour la traiter). Il les analyse avant de les orienter vers le(s) port(s) de sortie correspondant à leurs chemins. L'analyse des trames comprend plusieurs traitements (vérification de l'intégrité des trames et du contrat de trafic). Le délai de commutation d'une trame (récupération sur le port d'entrée, contrôle et routage) est borné par $16\mu\text{s}$ [52].

Chaque port de sortie correspond à un point de multiplexage doté d'une file d'attente avec une politique de service. Le réseau AFDX supporte les politiques d'ordonnancement *First-In-First-Out* (FIFO) et *Fixed Priority/FIFO* (FP/FIFO).

Un port de sortie avec la politique de service FP/FIFO est composé de n files d'attente FIFO : un niveau de priorité différent par file (cf. Fig. 3.20). Chaque flux v possède un niveau de priorité P .

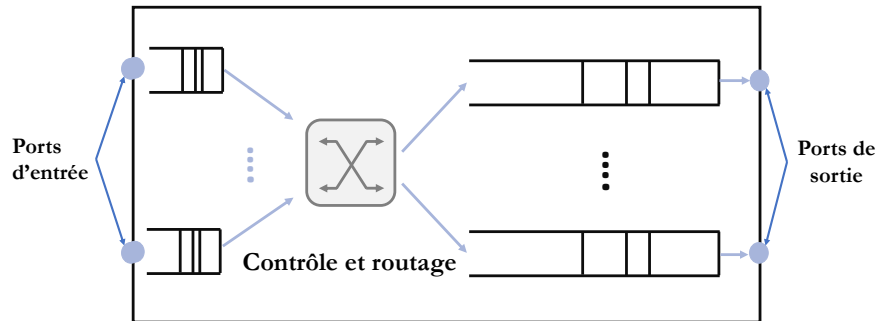


FIGURE 3.19 – Schéma d'un commutateur AFDX Ethernet.

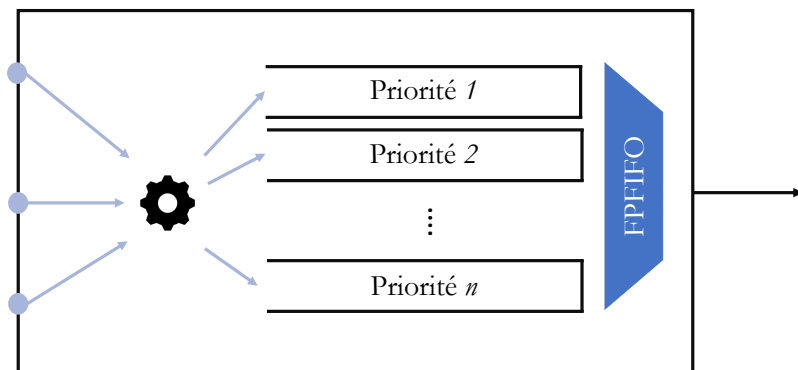


FIGURE 3.20 – Modélisation d'une file d'attente FPFIFO.

Ainsi, considérons l'exemple illustré dans la Fig. 3.21 avec 3 niveaux de priorité. La trame 4 est servie après la trame 3 même si elle est arrivée avant. En effet, à la fin de transmission de la trame 1, une trame est en attente dans chacune des files de priorités 2 et 3 (respectivement les trames 3 et 4). Puisque la trame 3 est plus prioritaire alors elle passe avant la trame 4.

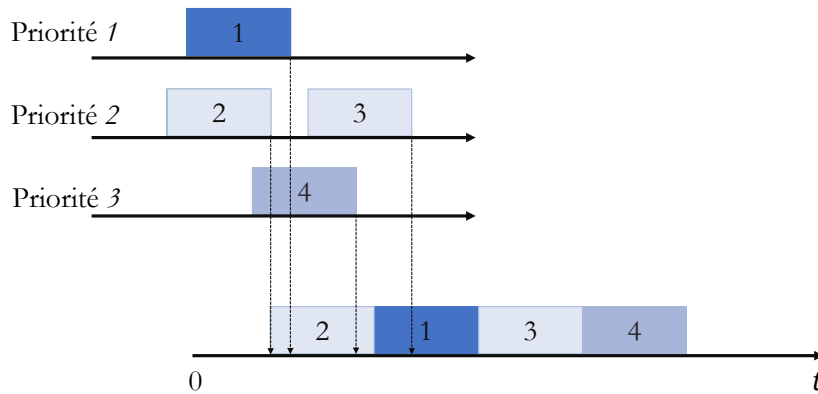


FIGURE 3.21 – Exemple d'un ordonnancement FP/FIFO.

Le commutateur est le point faible du réseau AFDX. Si un commutateur tombe en panne, tous les flux qui le traversent seront perdus. Pour des raisons de certification, le réseau AFDX est dupliqué en un réseau miroir. Les trames sont émises sur les deux réseaux en parallèle et les ESs destinataires traitent le premier message arrivé.

3.5.5 Exemple industriel

Le réseau AFDX défini pour l'A380 [63] est composé d'une centaine d'ESs reliée par huit commutateurs AFDX et environ mille flux multicast. Cette configuration est illustrée par la Fig. 3.22 (commutateurs en noir et ESs en gris). Les BAG des flux sont exprimés en puissances de 2 ne dépassent pas les 128 μ s. De plus, la taille des trames varie entre 65 et 1518 octets. Quant aux chemins des flux, le nombre de sauts n'excède pas les 4 sauts (commutateurs). La majorité des chemins ne dépassent pas 3 sauts.

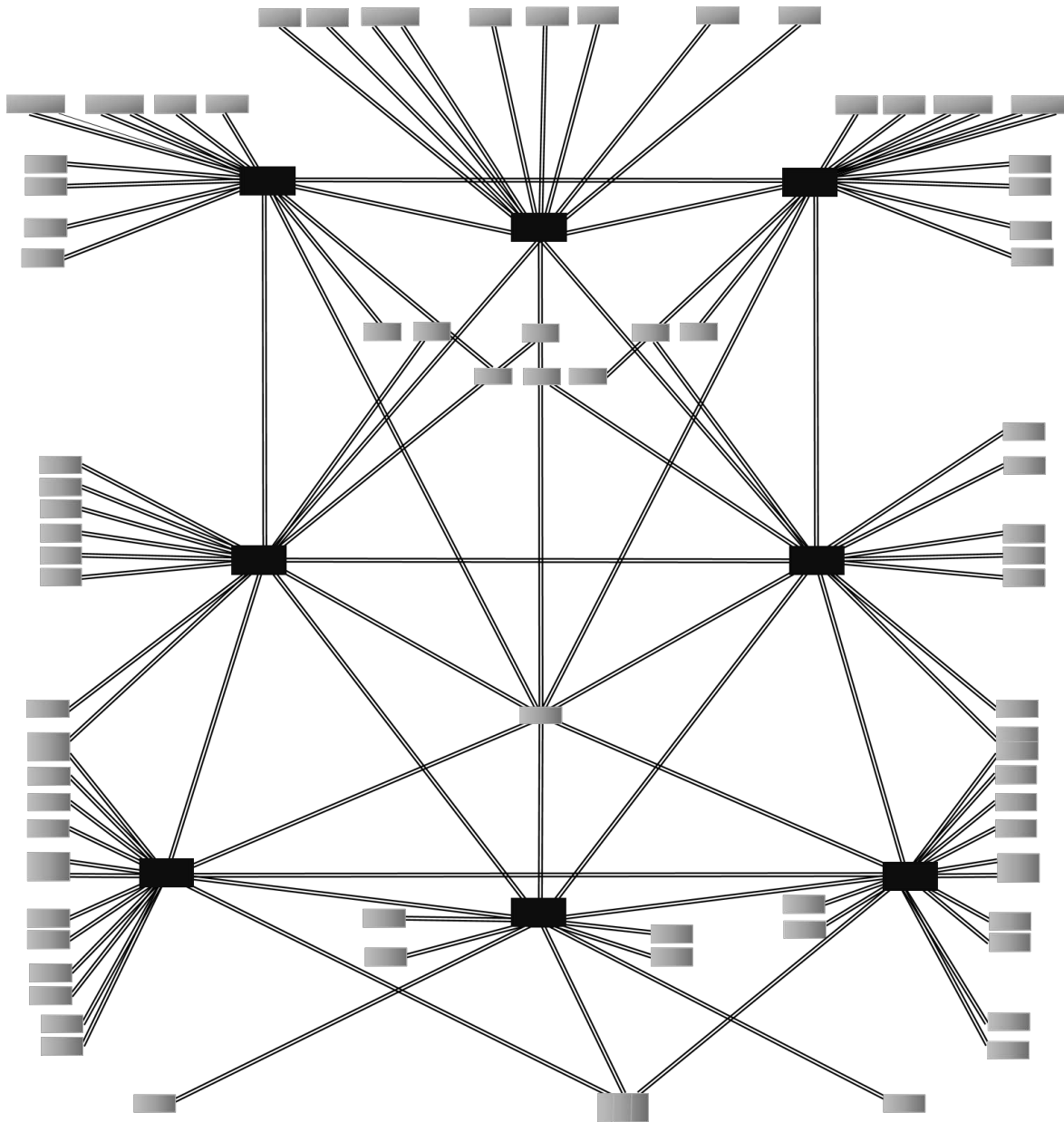


FIGURE 3.22 – Configuration d'un réseau AFDX.

3.6 Audio/Vidéo Bridging Switched Ethernet (AVB)

3.6.1 Définition du réseau AVB

Le réseau AVB a été développé par le groupe *AVB Work Group* de IEEE 802.1 [6]. Il regroupe un ensemble de standards pour garantir la qualité de service. C'est un réseau dédié aux flux audio et vidéo, basé sur trois protocoles : *Precision Time Protocol*, *Queueing and Forwarding rules* et *Stream Reservation Protocol* [62]. Chaque protocole fournit une ou plusieurs propriétés telles que la qualité de service et la synchronisation des flux.

3.6.2 Les composants du réseau AVB

Tout comme le réseau AFDX, le réseau AVB est composé d'un ensemble de commutateurs (*AV Bridge*) servant d'intermédiaires entre les terminaux émetteurs (*talkers*) et récepteurs (*listener*) de données (cf. Fig. 3.23).

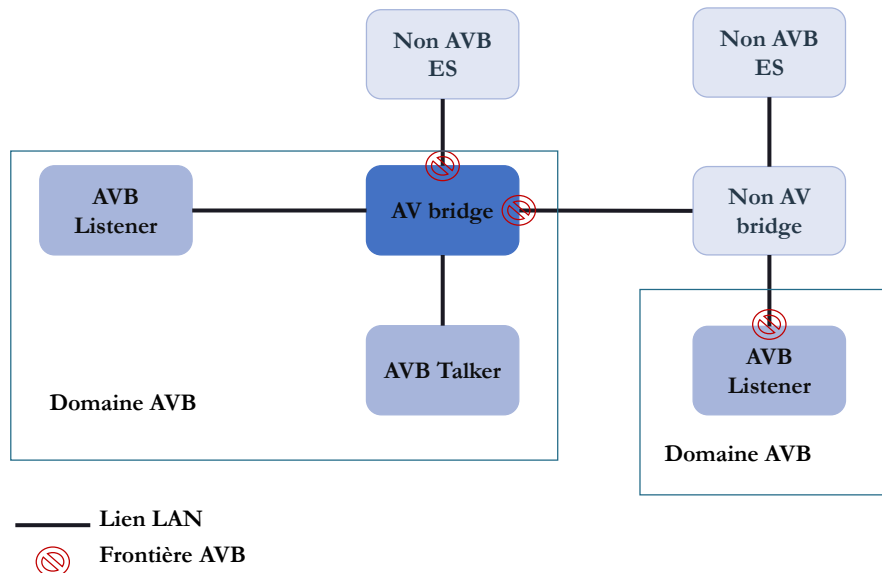


FIGURE 3.23 – Exemple de domaines AVB.

Les flux *streams* vont des *talkers* aux *listeners*. Cependant il peut y avoir des flux dans le sens inverse. Ces flux sont généralement de type *Best Effort* BE.

Les composants d'un réseau AVB forme l'*AVB domain* (cf. Fig. 3.23). Ils sont tous dotés des protocoles AVB nécessaires. Les nœuds qui n'implémentent pas les protocoles de l'AVB sont considérés comme des nœuds étrangers. Le domaine AVB est déterminé par la présence ou non de ces nœuds étrangers. Par exemple dans la Fig. 3.23, les trois nœuds non AVB ne font pas partie des domaines AVB encadrés dans le schéma.

3.6.3 Les propriétés du réseau AVB

Les protocoles du standard AVB fournissent plusieurs propriétés.

3.6.3.1 Synchronisation des flux

Dans un réseau destiné au multimédia, il est important d'assurer la cohérence d'une horloge commune entre la source et la destination de chaque flux. Cette synchronisation est assurée par le standard *Timing and Synchronization for Time-Sensitive Applications* IEEE 802.1AS [4] basé sur le protocole *Precision Time Protocol* (PTP) [5]. La synchronisation s'établit en deux étapes : sélection du nœud "*grand master*" par l'algorithme BCMA, ensuite, synchroniser les nœuds distribués avec le nœud *grand master* en construisant un arbre recouvrant tout le reste du réseau.

3.6.3.2 Différentiation des flux

Dans le réseau AVB, il existe des classes de flux *streaming*. L'AVB peut fournir jusqu'à 8 classes à niveaux de priorité distincte. En général, seulement deux classes *Stream Reservation* SR : *SR Class A* (SRA) et *SR Class B* (SRB) sont utilisées. La classe SRA est plus prioritaire que la classe SRB. Leurs délais de traversée de bout en bout respectifs sont en général respectivement *2ms* et *50ms*. Le réseau AVB fournit également la possibilité d'introduire des flux *Best-Effort* (BE). Ces flux sont les moins prioritaires.

3.6.3.3 Qualité de service

La qualité de service (QoS) permet la priorisation de certaines applications sur le réseau. Les applications sont classifiées selon leur niveau d'importance. La QoS assigne suffisamment de bande passante à une classe d'applications afin de diminuer son délai de traversée de bout en bout et d'optimiser le réseau.

Le rôle principal du réseau AVB est d'assurer la qualité de service en réservant de la bande passante aux applications *streaming* via le protocole *Stream Reservation Protocol* ou SRP IEEE 802.1Qat [7]. Ce dernier réserve 75% de la bande passante aux flux *streaming*. La réservation de la bande passante pour chaque flux se fait par un processus appelé *Signaling Process*. La première étape de ce processus est d'envoyer un flux d'annonce en *multicast* dans lequel on envoie les informations sur le flux *streaming* en question. Puis chaque commutateur intermédiaire calcule la bande passante nécessaire à ce flux. S'il y a assez de bande passante, le nœud destinataire envoie un flux d'enregistrement pour allouer la bande passante.

3.6.3.4 Politique de service à base de crédit

La politique de service employée dans le réseau AVB est la politique FP/FIFO couplée avec un algorithme de régulation de flux à base de crédit dénommé *Credit Based Shaper* ou CBS. L'objectif de CBS est de limiter la bande passante pour les flux plus prioritaires et éviter la famine pour les flux moins prioritaires comme les flux *Best Effort*. Cette politique est définie par le protocole *Forwarding & Queuing* IEEE 802.1Qav [8]. L'algorithme CBS est appliqué uniquement sur les classes SR. Un crédit est défini pour chaque classe SR. La transmission des flux dépend de la valeur de leur crédit. Une trame ne peut être transmise que lorsque son crédit est positif ou nul.

Le nouveau protocole *Time Sensitive Networking* (TSN) définit une évolution du protocole AVB. Nous le présentons dans le paragraphe suivant.

3.6.4 Le protocole Time Sensitive Networking

Le protocole *Time Sensitive Networking* (TSN) [83] est un standard Ethernet. L'objectif du projet TSN est de standardiser un réseau commuté temps réel optimisé et à faible coût qui minimise les latences des flux critiques. En effet, en plus des classes SR et du trafic BE, un troisième type de flux est introduit dans le protocole TSN pour les flux critiques (plus haute priorité). Sa politique d'ordonnancement *Time Aware Shaper* (TAS) est une extension de la politique de service définie par *Forwarding & Queuing* IEEE 802.1Qav [8] (FP/FIFO avec CBS).

3.6.4.1 Les types de flux

TSN est une version évoluée du réseau AVB. Par conséquent, les classes SR et les flux BE font partie des types de flux traités dans TSN. La différence avec le réseau AVB est l'introduction de flux de type *Time Triggered* (TT) du standard TTE (*Time Triggered Ethernet*). La classe de flux TT est plus prioritaire que les classes SR. Dans le réseau TTE, un flux TT est ordonné en hors-ligne avec le protocole *Time Division Multiple Access* (TDMA). Ce dernier définit des fenêtres de transmissions. Les trames générées par un flux TT sont transmises dans ces fenêtres de transmissions.

3.6.4.2 Politique de service Time Aware Shaper

La politique de service employée est définie par le protocole *Enhancements for Scheduled Traffic* IEEE 802.1 Qbv [85]. Il s'agit de la politique *Time Aware Shaper* (TAS). Elle est basée sur la stratégie FP/FIFO avec l'introduction d'un régulateur CBS pour les classes de flux SR et du protocole TDMA pour la classe de flux TT. Le port de sortie est modélisé par les files d'attente suivantes : les trois files d'attente du réseau AVB (SRA, SRB et BE), ainsi que la file de la classe TT.

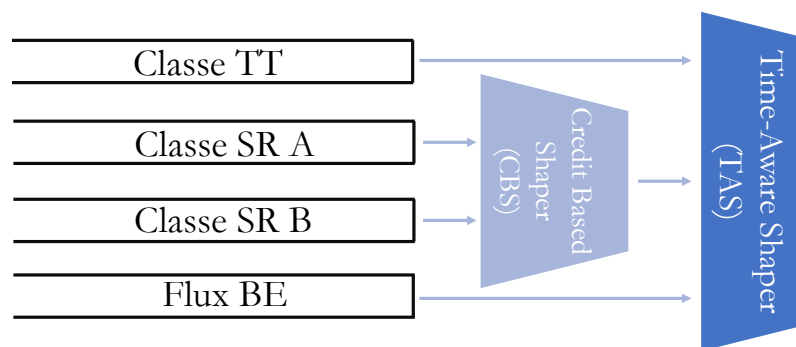


FIGURE 3.24 – Modélisation d'un port de sortie dans le réseau TSN.

La politique de service est non-préemptive exceptée pour les trames TT. En effet, l'ordonnanceur préempte un flux SR ou BE si dans une fenêtre d'ordonnancement TT, une

trame du flux TT est prête à être transmise. La préemption des trames est gérée par le protocole *Frame Preemption* IEEE 802.1Qbu [84].

3.7 Conclusion

L'évolution des besoins des systèmes temps réel distribués en communication a motivé l'utilisation du réseau Ethernet commuté. Des protocoles basés sur le réseau Ethernet ont été développés tout en respectant les propriétés temps réel (déterminisme, sécurité, flexibilité). Par exemple, le réseau AFDX est un protocole Ethernet commuté destiné aux systèmes avioniques.

En revanche, le déploiement d'un réseau temps réel ne peut pas être établi sans la validation de sa configuration. Une configuration est valide lorsque son évaluation garantit le respect de toutes les contraintes temps réel. En particulier, il est nécessaire de garantir une borne supérieure sur le délai de traversée du réseau pour toutes les trames. De plus, il faut garantir la non perte de trame notamment au niveau des files d'attente.

Dans le chapitre suivant, nous présentons les principales méthodes d'évaluation des réseaux temps réel embarqués.

Chapitre 4

Évaluation des réseaux temps-réel

Sommaire

4.1 Introduction	45
4.2 Problématique	45
4.3 Les principales méthodes d'analyse	46
4.3.1 La simulation	46
4.3.2 Méthodes par vérification de modèles	47
4.3.3 Network Calculus	47
4.3.4 La méthode holistique	49
4.3.5 Méthode des trajectoires	49
4.3.6 <i>Compositional Performance Analysis</i>	51
4.3.7 La méthode des intervalles éligibles	51
4.3.8 La méthode FA	52
4.4 Conclusion	53

4.1 Introduction

Le déterminisme des réseaux temps-réel doit être garanti pour les applications critiques. En particulier, il faut s'assurer qu'aucune trame ne soit perdue sur le réseau. Les liens physiques AFDX et AVB assurent qu'aucune collision n'est possible. Le problème est déplacé dans les files d'attente des ports de sortie. Elles doivent être dimensionnées suffisamment grandes pour garantir le non débordement d'une d'entre elles.

Un autre besoin est de déterminer une borne supérieure garantie du délai de bout en bout de chaque trame. Les deux problèmes sont connexes et nécessitent une méthode d'analyse du délai de bout en bout de traversée du réseau par une trame.

Dans le paragraphe 4.2, nous présentons les difficultés quant à l'évaluation des réseaux AFDX et AVB. Nous présentons ensuite, dans le paragraphe 4.3, les principales méthodes d'analyse du délai de bout en bout ainsi que le dimensionnement de files d'attente dans de tels réseaux.

4.2 Problématique

Déterminer le délai de bout en bout de traversée d'un réseau temps réel comme l'AFDX ou l'AVB est un problème complexe. En effet, en fonction des différents scénarios possibles, les trames générées par un même flux ne croisent pas le même nombre de trames dans les files d'attente qu'elles traversent. Ainsi, elles peuvent souffrir de grandes variations dans leurs délais. De plus, le nombre de scénarios possibles sur les instants d'arrivée de chaque trame dans chaque file d'attente est très important même pour de petites configurations composées de quelques flux traversant deux ou trois commutateurs. Cette grande variabilité des délais est accentuée par la nature même des flux qui sont sporadiques. Ainsi, le délai entre la génération de deux trames consécutives n'est pas fixe mais seulement défini par une durée minimum.

Une autre conséquence est qu'une trame peut, au cours de la traversée du réseau, rejoindre une trame générée avant elle par le même flux. Ainsi, plusieurs trames d'un même flux peuvent retarder une trame sous analyse, y compris des trames du flux étudié.

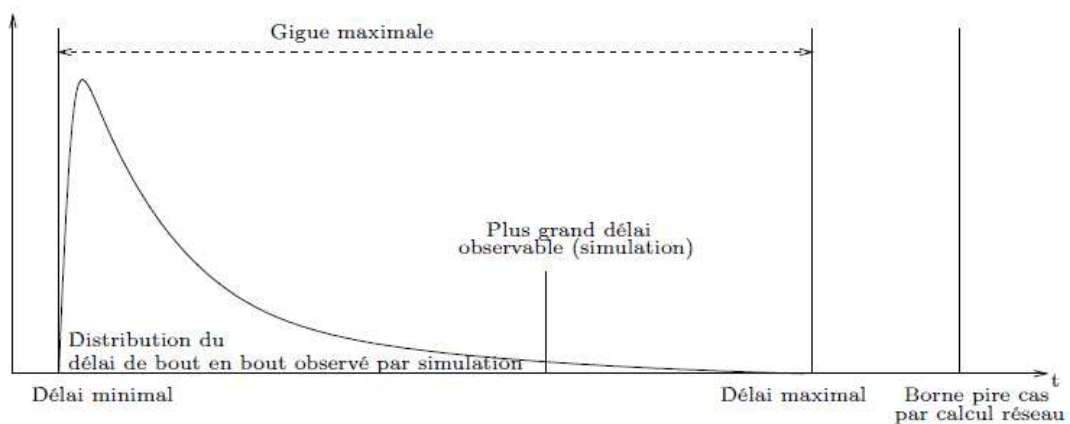


FIGURE 4.1 – Les résultats d'une analyse par simulation.

Dans le réseau AFDX, il n'y a pas de protocole de synchronisation. Ainsi, chaque élément (commutateur ou *End System*) dispose d'une horloge locale qui peut dériver avec le temps. Cependant, dans les réseaux AVB le protocole IEEE 802.1 AS [4] permet la synchronisation des nœuds.

L'analyse du délai de bout en bout de traversée du réseau par une trame comporte différents éléments (cf. Fig. 4.1) :

- le délai minimal de traversée du réseau. Ce délai est une constante obtenue par le scénario le plus favorable : la trame analysée est transmise, sans attente, à son arrivée dans chacun des éléments réseaux traversés;
- la distribution observée du délai en fonction d'un ensemble de scénarios joués;
- le délai maximum exact de traversée du réseau. Il correspond au scénario pire cas réalisable pendant lequel la trame subit les pires temps d'attente dans chaque élément réseau traversé. Jusqu'à présent, ce délai est impossible à calculer pour des configurations industrielles car le nombre de scénarios à considérer est exponentiel;
- des bornes supérieures sur le délai pire cas. Ces bornes correspondent à des calculs pessimistes mais fournissent des bornes supérieures garanties sur le délai de bout en bout. Une borne supérieure garantie est suffisante pour la validation des applications critiques. En pratique, c'est l'approche privilégiée par les organismes de certification

Plusieurs approches ont été définies dans la littérature au cours des années afin d'analyser le délai de bout en bout de chaque trame générée par les flux avioniques des réseaux AFDX et/ou AVB. Nous présentons dans la suite les principales approches avec pour chacune d'elle, les domaines d'applications et les résultats fournis.

4.3 Les principales méthodes d'analyse

4.3.1 La simulation

La simulation consiste à générer des scénarios aléatoires pour simuler la traversée des trames sur le réseau en les paramétrant [35] (instant d'arrivée, taille de trame, etc.). Pour chaque scénario, le délai de bout en bout de chaque trame est enregistré. La distribution en est déduite. Le pire délai pour une trame par simulation est le plus grand délai observé.

Cependant, elle ne peut pas couvrir tous les scénarios possibles. Un scénario très rare peut potentiellement être oublié par la simulation et donc ne pas figurer dans les résultats. Ainsi, l'analyse par simulation ne peut ainsi pas être utilisée pour valider les réseaux temps réel.

La méthode par simulation est utilisée pour analyser le comportement moyen du réseau. Elle a été appliquée au réseau AFDX pour déterminer les délais de bout en bout des flux avioniques [87]. Les résultats de simulation sont comparés à d'autres méthodes d'analyse comme le *Model Checking* dans [36]. Les performances sur la gigue (la gigue est la différence de délai de transmission de bout en bout entre des trames générées par un flux) dans les ESs d'un réseau AFDX sont analysées par simulation dans [100] avec différentes politiques d'ordonnancement notamment FIFO et *Round Robin*. Une

deuxième étude consiste à comparer l'impact de trois différentes politiques d'ordonnement (FIFO et deux autres politiques basées sur la différenciation de flux) sur les flux avioniques d'un réseau AFDX [60].

D'autres travaux sur l'analyse par simulation ont été menés pour le réseau AVB. Dans [75], les auteurs présentent une étude par simulation pour analyser plusieurs configurations du réseau AVB dans un réseau automobile. Le but est de démontrer le potentiel du réseau AVB dans le domaine temps-réel. Une autre analyse par simulation compare les bornes sur une configuration AVB avec les bornes obtenues pour un réseau Ethernet standard dans [62] avec les politiques de services CBS et FP/FIFO. Steinbach *et al.* ont étendu l'outil OMNeT++ INET Framework pour pouvoir simuler un réseau AVB [98]. Une autre analyse intéressante vise à évaluer un réseau avionique commuté avec la politique de service CBS du réseau AVB [49] définie dans le paragraphe 3.6.3.4.

4.3.2 Méthodes par vérification de modèles

La méthode par vérification de modèle [14, 15], ou *Model Checking* (MC) en anglais, est une méthode formelle basée sur l'analyse d'automates temporisés. Elle consiste à analyser tous les scénarios possibles d'arrivée de chaque trame dans chaque nœud traversé. Cette analyse correspond à l'exploration de tous les états du réseau. Elle permet donc d'obtenir le délai pire cas exact. Elle a été appliquée au réseau AFDX dans [36]. À notre connaissance, il n'y a pas de travaux dans la littérature sur l'application du *model checking* à l'AVB ni sur le dimensionnement de file d'attente.

Cependant, le nombre de scénarios à considérer entraîne une explosion combinatoire. Il est impossible de passer à l'échelle pour une configuration industrielle. Néanmoins, certains travaux visant à limiter le nombre de scénarios sont menés [10, 11, 70]. D'autres travaux menés pour optimiser la méthode MC consistent à considérer la corrélation du temps entre les flux [102].

4.3.3 Network Calculus

La méthode utilisée par Airbus pour certifier le réseau AFDX est le **Network Calculus** (NC) [32, 39, 40]. Elle est basée sur l'algèbre (min,+). Pour chaque flux, NC analyse séquentiellement chaque nœud (ou port de sortie) traversé. Il fait une analyse locale du pire délai et le propage dans la suite du chemin.

Chaque composant (flux, lien physique, port de sortie) est modélisé par des fonctions affines par morceaux, dépendantes du temps t .

Chaque nœud (*p.ex.* port de sortie d'un commutateur ou ES de l'AFDX) est modélisé par une courbe de service. Elle définit le nombre maximum de bits que peut transmettre le port de sortie durant un intervalle de temps t . Si le lien de sortie est de débit R alors l'élément réseau est modélisé par la courbe de service $\beta(t) = Rt$.

Chaque flux est modélisé par une courbe d'arrivée dans un nœud. Il existe plusieurs types de courbes d'arrivée mais la plus classique est celle de type "seau percé" (*Leaky Bucket*). Elle représente le nombre maximal de bits pouvant arriver sur un nœud durant un intervalle de temps t . À l'instant 0, le seau est vide et au maximum une rafale de b bits (b étant la taille maximum de la trame, F_{max}) peut arriver. Ensuite, le seau est plein et le trafic arrive au débit de la fuite du seau r . Par exemple, dans l'AFDX, une trame de taille maximale peut arriver tous les BAG unités de temps. Nous obtenons $r = \frac{F_{max}}{BAG}$. Le flux v

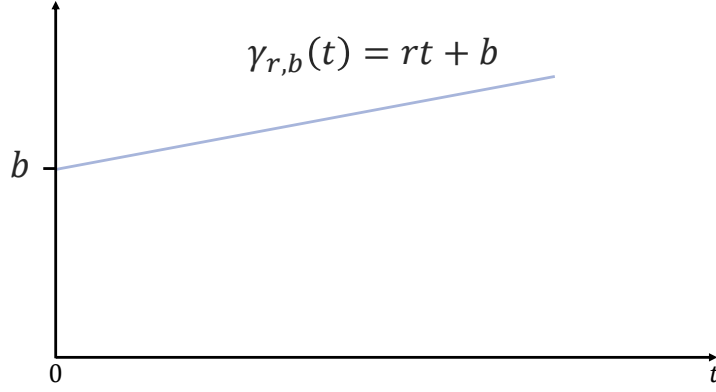


FIGURE 4.2 – La courbe d’arrivée d’un flux.

est ainsi modélisé par la courbe d’arrivée $\gamma_{r,b}(t) = rt + b$ (cf. Fig. 4.2).

Quand on analyse un nœud N, les courbes d’arrivée de tous les flux sont sommées pour obtenir une unique courbe d’arrivée α . A partir de ces deux courbes (α et β), le délai maximum encouru dans le nœud N peut être déduit ainsi que la quantité de données maximum en nombres de bits présents dans le nœud N à n’importe quel instant. En effet, le temps d’attente maximal correspond à la distance horizontale maximale entre les deux courbes (cf. Fig. 4.3), notée $h(\alpha, \beta)$.

A noter que le délai pire cas subi dans le nœud N pour une trame générée par n’importe quel flux parmi les flux agrégés est identique car la trame étudiée est toujours considérée comme la plus retardée.

La distance verticale maximale correspond au nombre pire cas de bits présents dans le nœud simultanément à n’importe quel instant, dénoté $v(\alpha, \beta)$ (cf. Fig. 4.3). Cependant, à notre connaissance, il n’existe pas dans la littérature de travaux permettant de dimensionner les files d’attente à partir de NC. Néanmoins, Airbus et ses fournisseurs utilisent NC pour les dimensionner dans la pratique.

Finalement, la courbe de sortie de chacun des flux est déduite pour être propagée dans le nœud suivant.

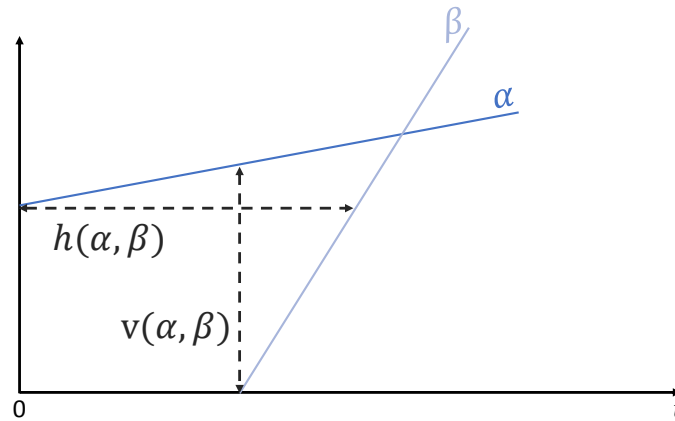


FIGURE 4.3 – La courbe de service et la courbe d’arrivée d’un nœud N.

La méthode NC a été développée pour les politiques de service FIFO [32], FP/FIFO [33] pour le réseau AFDX. Une version probabiliste de NC est proposée par Ridouard *et al.* dans [91, 90]. NC a été étendue à l’AVB et la politique CBS [86, 41] ainsi qu’au protocole TSN et l’algorithme TAS [57] présentés dans le paragraphe 3.6.4.

4.3.4 La méthode holistique

La méthode holistique [104] calcule, également, le pire délai local pour chaque nœud traversé. Itérativement, les giges sont mises à jour jusqu'à l'obtention d'un point fixe qui correspond au pire délai final. Initialement, la méthode holistique a été conçue pour les systèmes temps réel avec ordonnancement préemptif. George *et al.* [48] ont étendu cette approche à l'ordonnancement non-préemptif. L'analyse holistique a été appliquée aux réseaux de paquets mais sans preuve formelle [78]. Dans [54], la méthode holistique a été appliquée pour analyser les End Systems du réseau AFDX.

4.3.5 Méthode des trajectoires

La méthode des trajectoires TA a été développée dans un contexte distribué pour la politique d'ordonnancement FIFO par Martin *et al.* [79]. Elle a été appliquée à l'AFDX par Bauer *et al.* [23] puis étendue aux politiques FP/FIFO et WFQ [24]. Finalement, la méthode des trajectoires a été appliquée à l'AVB [73].

Pour illustrer le principe de la méthode des trajectoires nous allons analyser le délai du flux v_3 suivant le chemin $ES_2 - S_1 - S_4$ de la configuration présentée dans la Fig. 3.16. Il croise respectivement v_2 sur ES_2 , v_1 sur S_1 et finalement v_4 et v_5 sur S_4 . Pour simplifier la présentation, les trames ont toutes une durée de transmission de $10\mu s$ dans tous les nœuds. De plus le délai entre la génération de deux trames d'un même flux est suffisamment grand pour que seulement, au plus, une trame de chaque flux ne puisse retarder v_3 . La Fig. 4.4 représente un scénario possible de traversée du réseau pour une trame du flux v_3 . À noter que la trame générée par le flux v_i est notée i et que sa date d'arrivée sur un nœud est symbolisée par une flèche verticale accompagnée de son numéro. L dénote le délai de commutation.

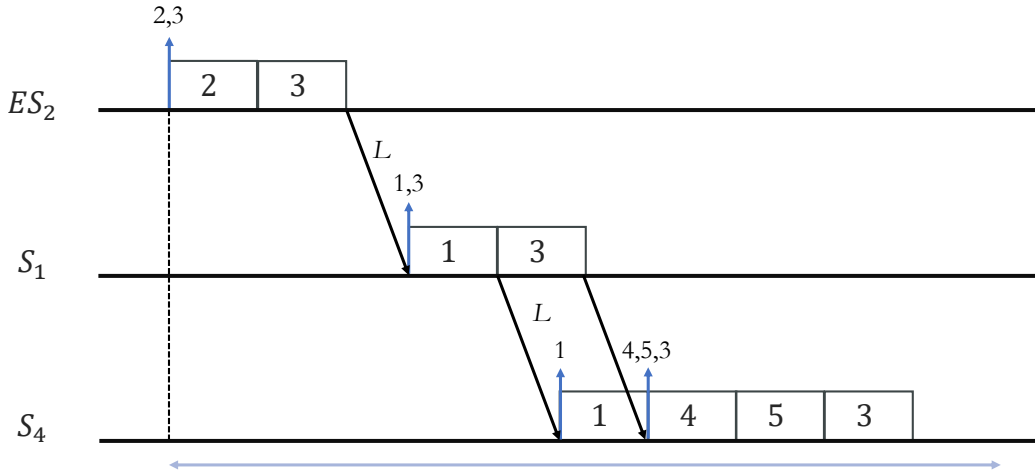


FIGURE 4.4 – Scénario possible de traversée de la trame 3 du flux v_3 .

La méthode TA fonctionne en *backtracking* (du dernier nœud en remontant vers le premier). Elle essaie de maximiser la période d'activité (définie par la définition 2 et la définition 3) générée par chacun des flux, dans chaque nœud.

Définition 2 Une *période d'activité* est définie par un intervalle $[t_1, t_2)$ ($t_1 < t_2$) tel que t_1 et t_2 sont tous deux des instants oisifs et il n'y a aucun instant oisif dans l'intervalle $[t_1, t_2)$.

Définition 3 Un *instant oisif* est un instant t auquel toutes les trames générées avant t ont été traitées à l'instant t .

Le pire scénario construit par la méthode TA pour le flux v_3 est dépeint Fig. 4.5. Les périodes d'activité entre deux nœuds consécutifs sont reliées entre elles par des trames dites pivot (*p.ex.* la trame 1 entre S_1 et S_4). La méthode TA construit un scénario qui maximise cette période d'activité inter-nœuds en agrégeant tous les nœuds en un seul.

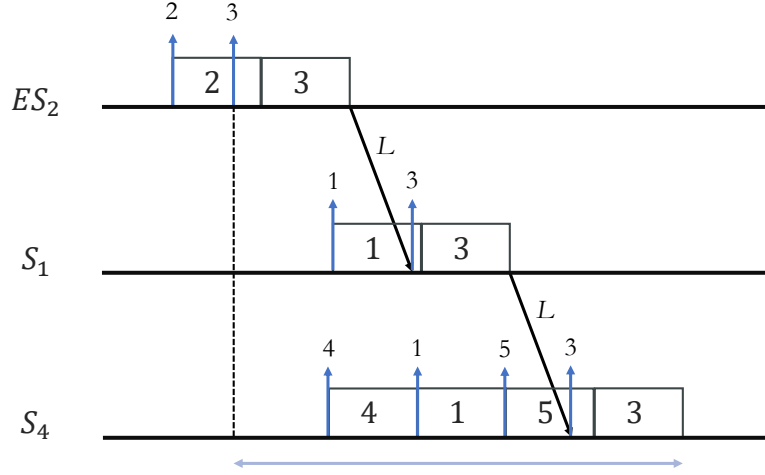


FIGURE 4.5 – Scénario pire cas de traversée de la trame 3 du flux v_3 .

Cependant, cette agrégation est aussi une limitation importante de la méthode TA. Ainsi, même si un flux traverse des nœuds avec des charges locales inférieures à 1 (*cf.* définition 4), si la charge globale de tous les flux croisés sur le chemin complet est supérieure à 1, TA ne peut pas s'appliquer.

Définition 4 La charge locale d'un nœud N traversé par un ensemble de flux Γ est donnée par :

$$\sum_{v_i \in \Gamma} \frac{C_i}{T_i}$$

Où C_i (resp. T_i) dénote le temps de transmission (resp. période) d'une trame du flux v_i .

De plus, même si un problème d'optimisme dans la borne déterminée par TA a été démontrée dans [67] et corrigée par [72], il n'y a, à ce jour, aucune démonstration formelle de la correction de cette approche.

Pour déterminer la quantité de données maximum présentes dans chaque file d'attente, Coelho *et al.* [38] déduisent le nombre de trames maximum présentes dans une file d'attente à partir du calcul du temps d'attente pire cas dans la file si toutes les trames sont de même taille.

4.3.6 *Compositional Performance Analysis*

La méthode *Compositional Performance Analysis* (CPA), implémentée dans l'outil SymTA/S [59], est basée sur la modélisation des événements comme la méthode *Real Time Calculus* RTC. La méthode RTC [103] est semblable à la méthode NC développée pour les réseaux temps réel. Cependant, RTC est une méthode d'analyse des systèmes temps réel qui traitent les événements de flux de données (une tâche est générée par un événement). Elle consiste à modéliser chaque flux de données par deux courbes d'arrivée dans chaque système. Ces deux courbes représentent les bornes respectivement inférieure et supérieure du nombre d'événements susceptibles d'être déclenchés à n'importe quel instant. Un graphe est défini pour modéliser les dépendances entre les tâches. Chaque ressource du système est également modélisée par des courbes (*Request Curve* et *Delivery Curve*). Les bornes sont calculées de la même manière que le Network Calculus. Il s'agit de maximiser et de minimiser la courbe d'arrivée en jouant sur les chemins possibles dans le graphe des tâches. Cette méthode a été utilisée dans le contexte des réseaux. Elle est appliquée au réseau Ethernet commuté dans [89].

La différence réside dans l'analyse des commutateurs et la modélisation des liens. Elle est appliquée au réseau Ethernet commuté Full Duplex [93] et au réseau AVB [44, 45]. Le modèle réseau de CPA est défini par un ensemble de tâches (trames de flux) qui sont traitées par un ensemble de ressources (ou ports de sortie). Les tâches sont générées par des événements.

La borne supérieure du délai de bout en bout d'un flux est calculée de manière itérative en parcourant toutes les ressources (ou ports de sortie) traversées par ce flux. Dans chaque ressource, un modèle d'événement pire cas est calculé pour chaque tâche. Ce modèle représente le scénario d'activation d'événements pire cas des tâches qui retardent l'exécution de la tâche en question. Une fois le modèle en sortie généré, il est transféré à la ressource suivante comme modèle en entrée. Chaque ressource effectue ainsi une analyse en utilisant les modèles d'événements en entrée de toutes les tâches et envoie les modèles mis à jour au port de sortie successeur. Le délai de bout en bout pire cas est déterminé au niveau de la dernière ressource. L'algorithme CPA est détaillé dans [43]. Elle a été améliorée dans [102] en exploitant la corrélation entre les flux.

4.3.7 La méthode des intervalles éligibles

Cette méthode consiste à définir des intervalles éligibles au lieu d'étudier des périodes d'activité [34]. Un intervalle éligible est constitué de la charge de travail maximum en tenant compte de la disponibilité du crédit. Elle a été développée pour un seul nœud. Les auteurs estiment que c'est une méthode prometteuse.

4.3.8 La méthode FA

La méthode *Forward end-to-end Delay Analysis* FA est basée sur les résultats d'ordonnement temps réel. En effet, la transmission des trames par un nœud est assimilable à l'exécution de tâches sur un système mono-processeur. Comme pour le Network Calculus, FA analyse les nœuds un par un dans l'ordre du chemin parcouru par un flux. Pour chaque nœud, elle construit un pire scénario local. Il est, bien souvent, impossible de construire un scénario global réaliste pendant lequel une trame subirait son pire scénario local sur chaque nœud traversé. En effet, les scénarios locaux pire cas sont construits indépendamment et ils tiennent compte uniquement de la gigue produite dans les nœuds précédents. Ainsi, FA est pessimiste et détermine une borne supérieure garantie sur le délai de bout en bout.

Cette méthode a été développée pour l'AFDX avec une politique de service FIFO [68, 66]. Le délai de bout en bout d'un flux traversant un réseau AFDX est composé, pour chaque élément traversé, de deux parties (cf. Fig. 4.6) :

- une partie constante, notée L qui correspond à la latence technologique. L est composé du délai de commutation et de la propagation de la trame sur les liens ;
- une partie variable qui représente le temps d'attente maximum d'une trame du flux dans chaque nœud traversé avant d'être transmise. Ce temps correspond à la transmission des autres trames interférant avec la trame analysée.

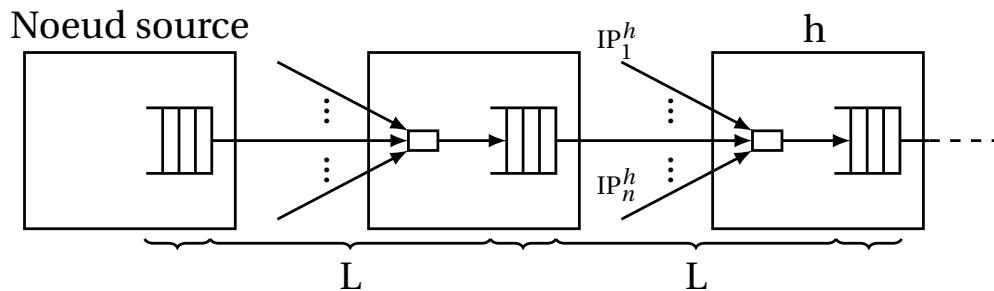


FIGURE 4.6 – Les parties constantes et variables du délai de bout en bout.

Le temps d'attente dans chaque nœud dépend de la quantité de trames interférentes avec la trame en question selon le contrat de trafic imposé. Ce délai peut subir de grandes variations en fonction des scénarios.

La méthode FA détermine une borne supérieure sur ce temps d'attente dans chaque nœud du réseau AFDX en construisant pour chaque flux entrant le pire scénario maximisant le nombre de trames générées qui interfèrent avec la trame sous analyse dans le nœud concerné.

4.4 Conclusion

Déterminer le pire délai des flux de bout en bout dans les réseaux temps réel tels que l'AFDX ou l'AVB peut s'avérer rapidement complexe même pour des configurations de taille modérée. C'est dû, en particulier, au nombre important de scénarios à considérer qui ne peuvent pas être exhaustivement étudiés.

Méthodes d'analyses	AFDX		AVB
	FIFO	FP/FIFO	CBS
Simulation	[36, 87, 100, 60]	[100, 60]	[62, 98, 49, 75]
Vérification de modèle	[36]	–	–
NC	[32]	[33]	[86, 41]
Méthode holistique	[104, 48]	[78, 54]	–
TA	[22, 23, 25]	[24]	[73]
CPA	–	–	[93, 44, 45]
Méthode des intervalles éligibles	–	–	[34]
FA	[68, 66]	–	–

TABEAU 4.1 – Synthèse bibliographique des principales méthodes d'analyse.

Dans ce chapitre, nous avons présenté les principales méthodes d'analyse de délai. Le tableau 4.1 résume les principaux travaux menés pour chaque approche en fonction du réseau (AFDX ou AVB) et de la politique d'ordonnancement. Plusieurs travaux ont été menés sur la simulation de ces protocoles via OMNET++ notamment. Le *Model Checking* n'a, à notre connaissance été appliquée que pour l'AFDX avec la politique d'ordonnancement FIFO. De plus, les configurations étudiées sont de petites tailles (quelques flux et deux ou trois commutateurs).

Ainsi, les principales méthodes développées pour valider de tels réseaux se focalisent sur le calcul d'une borne supérieure garantie sur ce délai. Ces méthodes sont pessimistes par définition et il est généralement impossible de construire un scénario réaliste correspondant à la borne fournie. Le Network Calculus (NC) est aujourd'hui utilisé par Airbus pour la validation de ses réseaux. NC a été appliquée à l'AFDX (FIFO et FP/FIFO) ainsi qu'à l'AVB. De plus, même si aucun des travaux dans la littérature ne porte sur l'utilisation de NC pour dimensionner des files d'attente, en pratique, Airbus l'applique.

La méthode des Trajectoires (TA) a également été appliquée à l'AFDX (FIFO et FP/FIFO) ainsi qu'à l'AVB. Quelques travaux traitent également de l'utilisation de TA pour dimensionner la taille maximum des files d'attentes mais dans le cas où toutes les trames ont la même taille. Cependant, il n'y a aucune démonstration formelle concernant la méthode TA en elle-même.

D'autres méthodes d'analyse existent comme la méthode holistique pour le réseau AFDX ou la méthode CPA pour le réseau AVB mais leurs résultats sont moins pertinents que ceux obtenus avec les méthodes NC et TA.

Quant à la méthode des intervalles éligibles, elle se limite à l'analyse d'un seul nœud jusqu'à présent et n'a pas été expérimentée sur une configuration industrielle ni comparée à une autre méthode. Néanmoins, les auteurs estiment qu'elle produit des bornes précises.

Nous nous focalisons, dans la suite, sur la méthode *Forward end-to-end Analysis* (FA). Elle n'avait été développée que pour l'AFDX avec la politique FIFO. Dans cette thèse, nous avons étendu FA à la politique FP/FIFO dans le chapitre 6 ainsi qu'à l'AVB dans le chapitre 7. De plus, nous avons également contribué sur le dimensionnement des files d'attente dans le chapitre 8, en développant une approche qui a besoin d'une topologie réseau, d'un contrat de trafic à l'entrée des flux ainsi qu'une méthode (quelconque) d'évaluation des délais. Mais tout d'abord, dans le chapitre 5, nous présentons la méthode FA avec FIFO comme politique de service.

DEUXIÈME PARTIE :

CONTRIBUTIONS

Chapitre 5

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

Sommaire

5.1	Introduction	59
5.2	Modèle réseau	60
5.3	Le calcul du délai pire cas d'une trame	61
5.3.1	Principe général	61
5.3.2	Le scénario pire cas pour maximiser l'arriéré de travail	62
5.3.3	Fonction cumulative et Request Bound Function	65
5.3.4	L'effet de sérialisation	66
5.3.5	L'arriéré de travail maximum	68
5.3.6	Condition suffisante pour l'arrêt du calcul	69
5.3.7	Formules de calcul et algorithmes	70
5.4	Expérimentations	72
5.4.1	Transposition de l'AFDX au modèle réseau de FA	73
5.4.2	Exemple de calcul en FIFO	74
5.4.3	Cas d'étude en FIFO	75
5.4.4	Configuration industrielle en FIFO	76
5.5	Conclusion	77

5.1 Introduction

La méthode *Forward end-to-end delay Analysis* (FA) a été initialement développée pour le réseau AFDX avec la politique de service FIFO dans le cadre de la thèse de G. KEMAYO au sein de notre laboratoire [68]. Dans la suite de ce chapitre, nous présentons la méthode FA issue de ces travaux. Cependant, même si la méthode FA a déjà été formellement démontrée, la première contribution de cette thèse consiste en la reformulation de certains points ainsi que l'extension à un modèle réseau plus général. Précisément, les modifications apportées dans cette thèse, à la version originelle de FA sont :

- **la reformulation du Théorème 1** de [68] : il caractérise la pire interférence générée par un flux sur un nœud durant un intervalle $[0, t]$. Il a été démontré formellement. Cependant, son principal inconvénient est de construire le scénario d'arrivée des trames sur un nœud en *backtracking* (de l'instant t en remontant jusqu'à 0). Nous avons reformulé ce théorème (cf. le théorème 1 dans le paragraphe 5.3) afin d'avoir un scénario équivalent mais construit depuis l'instant 0 jusqu'à t . Ce nouveau scénario facilite l'extension de FA à d'autres politiques de gestion de files d'attente. Le corollaire 1 a aussi été mis à jour pour correspondre à ce scénario.
- **modèle nœud** : dans [68], les nœuds du réseaux sont considérés comme soit un point d'entrée du réseau, soit un port de sortie d'un commutateur qui achemine la trame. Cependant, dans un réseau AFDX, les commutateurs peuvent aussi émettre des messages (*p.ex.* des messages de maintenance). Ainsi, le modèle nœud est généralisé pour lui permettre de faire les deux simultanément;
- Dans la première version de FA [68], **les liens physiques** ont tous le même débit. Le modèle réseau est généralisé pour lever cette restriction (cf. le lemme 1 dans le paragraphe 5.3.4);
- Pour conclure **la condition suffisante d'arrêt du calcul**, garantissant l'obtention d'une borne supérieure du délai de bout en bout de la trame sous analyse, est généralisée pour la rendre applicable à toute politique d'ordonnancement non-oisive.

Dans le paragraphe 5.2, nous détaillons le modèle réseau utilisé par FA. Nous définissons le principe général de FA dans le paragraphe 5.3. La composante principale dans le calcul du délai avec la méthode FA est l'arriéré de travail pire cas. Nous montrons comment maximiser l'arriéré de travail d'un nœud quelconque dans le paragraphe 5.3.5. Nous l'optimisons en tenant compte de l'effet de sérialisation (deux trames consécutives ne peuvent jamais traverser simultanément un lien physique) dans le paragraphe 5.3.4. Ensuite, nous récapitulons le processus du calcul des bornes sur le délai et donnons un exemple de calcul détaillé dans le paragraphe 5.3.7. Nous concluons ce chapitre par des expérimentations dans le paragraphe 5.4.

5.2 Modèle réseau

La méthode FA s'applique à des réseaux de paquets avec la politique d'ordonnement FIFO. Nous nous plaçons sous les hypothèses suivantes :

- le réseau est dans un mode de fonctionnement nominal. Il n'y a pas d'élément réseau en panne;
- Comme le réseau ne subit pas de panne, nous ne prenons pas en compte les dispositifs de redondance;
- le réseau n'est pas soumis à des problèmes de sécurité. Il est protégé des attaques (modification de la taille de trame, du routage...);
- les nœuds sont indépendants les uns des autres. Il n'y a pas de protocole de synchronisation.

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

Les nœuds modélisent les points de multiplexage du réseau. Ainsi, un nœud est composé d'un ensemble de ports d'entrée et d'une file d'attente FIFO, servie à un débit de r^h , reliée potentiellement à un ou plusieurs autres nœuds. Le modèle de nœud est illustré dans la Fig. 5.1.

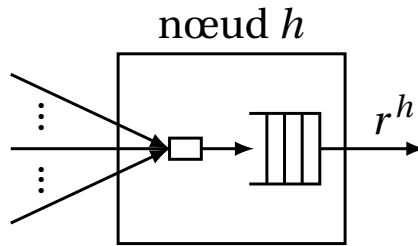


FIGURE 5.1 – Modélisation d'un nœud h .

Chaque flux sporadique $v_i \in \Gamma$ est un flux mono-émetteur, unidirectionnel et unicast (un seul destinataire). Il est généré par un nœud source, noté $first_i$, il suit un chemin statique dénoté $\mathcal{P}_i$ (composé de la liste ordonnée des nœuds traversés) et son dernier nœud est noté $last_i$. L'ensemble des flux qui ont pour nœud source h est dénoté Γ_0^h (avec $\Gamma_0^h \subset \Gamma^h$).

En plus, de son chemin, un flux v_i est caractérisé par :

- $Fmax_i$, la taille maximum (en bits) d'une trame générée par v_i . La durée de transmission maximum dénotée C_i^h , d'une trame générée par v_i sur le nœud 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, le délai induit par une trame générée par le flux v_i , illustré Fig. 4.6, est composé de :

- la **Latence technologique** dénotée L entre deux nœuds consécutifs. Elle inclut, notamment, la propagation électrique (négligeable par rapport à la longueur des liens utilisés);

- **un temps d'attente** (qui peut varier fortement) dans chaque nœud h de son chemin $\mathcal{P}_i$. Il correspond à la durée de transmission présente dans la file d'attente à l'instant d'arrivée de la trame analysée. Ce temps d'attente inclut sa propre durée de transmission C_i^h .

L'objectif de la méthode FA est de déterminer une borne supérieure garantie de ce délai d'attente dans les files.

5.3 Le calcul du délai pire cas d'une trame

5.3.1 Principe général

Pour déterminer le délai de bout-en-bout d'un flux v_i , tous les nœuds de son chemin sont analysés, dans l'ordre, séquentiellement.

Un délai pire cas, dénoté R_i^h (cf. Fig. 5.2), est calculé entre la génération d'une trame f_i de v_i sur son nœud source et sa sortie de chaque nœud h traversé, le calcul est alors propagé au nœud suivant jusqu'à son arrivée dans son nœud destination. Le calcul de R_i^h est donné par :

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

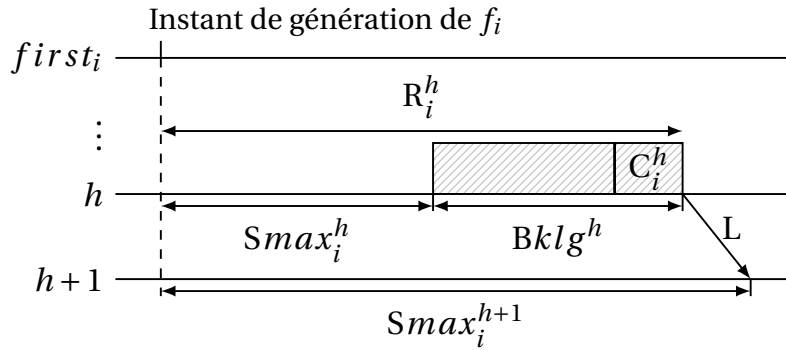


FIGURE 5.2 – Calcul itératif de R_i^h et $Smax_i^h$

Nous détaillons ci-après ces deux termes.

Définition 5 Une définition temporelle de l'arriéré de travail dans une file d'attente est le temps de transmission total des trames en attente dans la file à un instant donné t . $Bklg^h$ définit une borne supérieure de cet arriéré pire cas de travail dans la file d'attente de sortie du nœud h à n'importe quel instant.

Définition 6 Pour un nœud $h \in \mathcal{P}_i$, le terme $Smax_i^h$ dénote le temps maximum mis par une trame du flux v_i entre sa génération sur son nœud source $first_i$ et son arrivée sur le nœud h .

Par définition, nous avons $Smax_i^{first_i} = 0$. $Smax_i^h$ est déterminé de manière itérative. Soient 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 délai maximum pris par une trame de v_i pour atteindre h ($Smax_i^h$), de l'arriéré de travail maximum encouru sur h ($Bklg^h$) et de la latence technologique L :

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

Il est important de souligner que le terme $Bklg^h$ englobe également la durée de transmission de la trame sous analyse C_i^h .

En parallèle, le terme $Smin_i^h$, qui représente le plus petit délai mis par une trame du flux v_i entre sa génération sur $first_i$ et son arrivée sur le nœud $h \in \mathcal{P}_i$ est obtenu itérativement :

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

Finalement, le délai pire cas de traversée du réseau pour une trame du flux v_i est noté R_i et correspond, par définition, à $R_i^{last_i}$.

La principale difficulté du calcul est de déterminer $Bklg^h$ pour chaque nœud h . Ainsi, nous allons pour un nœud h , établir le scénario pire cas maximisant la contribution de chaque flux v_i ($v_i \in I^h$ à $Bklg^h$). Ce problème est étudié dans le paragraphe 5.3.2. Ce scénario est exprimé en utilisant le concept de *Request Bound Functions* (*rbf*) dans le paragraphe 5.3.3. Les formules sont optimisées en prenant en compte l'effet de sérialisation dans le paragraphe 5.3.4. L'expression finale de l'arriéré pire cas est donnée dans le paragraphe 5.3.7 et une condition d'arrêt du calcul dans le paragraphe 5.3.6. Finalement, la méthode complète est résumée, en incluant l'algorithme correspondant.

5.3.2 Le scénario pire cas pour maximiser l'arriéré de travail

Afin de déterminer $Bklg^h$, 0 est considéré comme l'instant de démarrage du nœud h (aucune trame ne peut arriver dans h avant 0). $Bklg^h$ est une borne supérieure sur l'arriéré de travail présent dans la file d'attente à n'importe quel instant $t \geq 0$. Pour une valeur de $t \geq 0$ donnée, il correspond au temps de transmission total des trames arrivées sur h dans l'intervalle $[0, t]$, dénoté $W^h(t)$ et qui n'ont pas été servies à l'instant t . L'intervalle $[0, W^h(t)]$ représente la plus longue période d'activité générée par les flux traversant h durant $[0, t]$.

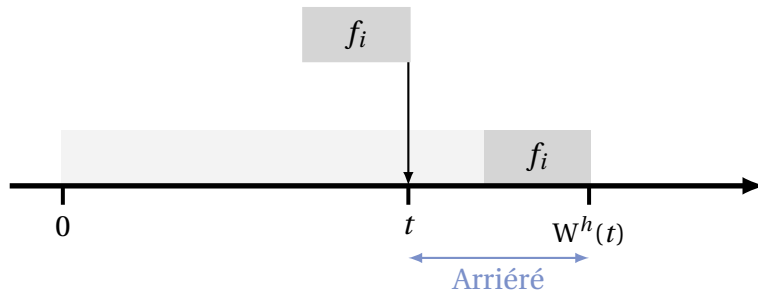


FIGURE 5.3 – Illustration de la période d'activité pour une valeur de $t \geq 0$ donnée.

Ainsi comme illustré dans la Fig. 5.3, pour une valeur de $t \geq 0$, l'arriéré présent dans la file d'attente à l'instant t s'obtient comme la longueur de la plus longue période d'activité [71] (voir la définition 2 dans le paragraphe 4.3.5) générée par l'ensemble des flux traversant h durant l'intervalle $[0, t]$ ($W^h(t)$) moins la longueur de l'intervalle (t) qui correspond à l'arriéré transmis par h . Quelle que soit la trame f_i analysée, elle arrive sur h à l'instant t et est la dernière à être transmise (juste avant $W^h(t)$).

Dans la suite, nous analysons l'intervalle de temps $[0, t]$ en déterminant l'arriéré de travail maximum généré par chaque flux $v_i \in I^h$ afin de maximiser la période d'activité et aussi l'arriéré de travail global $Bklg^h$ pour n'importe quelle valeur de $t \geq 0$.

Théorème 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;
- (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. 5.4).

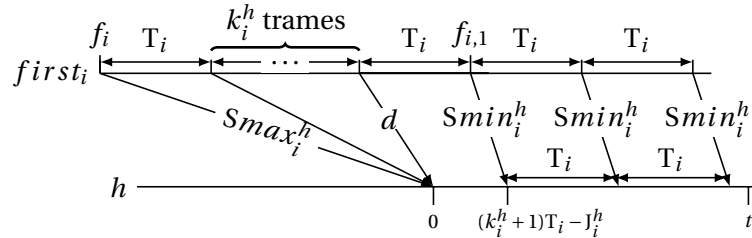
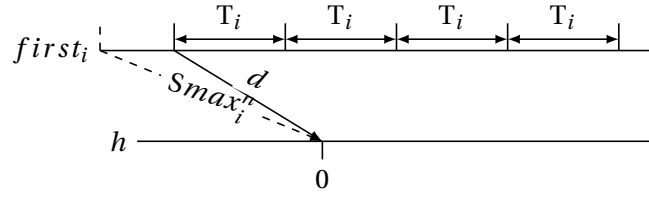


FIGURE 5.4 – Le scénario pire cas d'arrivée des trames générées par $v_i \in I^h$ dans le nœud h , maximisant l'arriéré de travail durant un intervalle de temps $[0, t]$.

Démonstration : Nous démontrons les points (i), (ii) et (iii) successivement.

- (i) L'arriéré de travail maximum est obtenu quand les flux génèrent le maximum d'interférence. Pour des flux sporadiques, l'interférence est maximum lorsque les trames sont générées avec un minimum de temps entre les générations sur leur nœud source. Par conséquent, pour un flux v_i , ceci correspond à la génération d'une trame chaque T_i unités de temps dans le nœud $first_i$. L'élément (i) est donc prouvé.
- (ii) Pour maximiser sa contribution dans un intervalle de temps $[0, t]$, les trames de v_i arrivent dans h aussitôt que possible. (ii) peut être prouvé par contradiction. Supposons que la première trame f_i arrive dans h strictement après 0, il est toujours possible de trouver un scénario pire cas en rapprochant l'arrivée de f_i à l'instant 0. Par conséquent, pour déterminer l'arriéré de travail maximum, la première trame f_i doit arriver dans le nœud h exactement à l'instant 0.


 FIGURE 5.5 – La première trame arrivant dans h après un délai $d < Smax_i^h$.

- (iii) On peut également prouver le point (iii) par contradiction : considérons que f_i arrive à l'instant 0 mais avec un délai $d < Smax_i^h$ pour atteindre h , comme il est représenté dans la Fig. 5.5. Puisque la trame f_i subit un délai plus petit que $Smax_i^h$, son instant de génération dans le nœud source $first_i$ est retardé de $Smax_i^h - d$. En plus, la génération de trames étant strictement périodique, comme il a été prouvé dans (i), les instants de génération de toutes les trames suivantes sont également décalés. Considérant qu'elles traversent toutes le réseau avec un minimum de temps pour rejoindre h ($Smin_i^h$), leurs arrivées sur h sont aussi retardées. Il est ainsi toujours possible de définir un pire scénario en avançant l'instant de génération de f_i .

La trame f_i arrive au nœud h à l'instant 0 avec un délai égal à $Smax_i^h$, mais pour les trames suivantes, ce délai peut varier entre $Smin_i^h$ et $Smax_i^h$. En effet, une ou plusieurs trames peuvent rattraper une trame générée avant elle(s), mais sans jamais la dépasser, à cause de la politique FIFO.

Le scénario, où k_i^h trames (avec $k_i^h \geq 0$) ont rattrapé f_i , est illustré dans la Fig. 5.4. 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 $Smin_i^h$.

Dans la Fig. 5.4, $f_{i,1}$ 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.

□

Le nombre k_i^h de trames arrivant simultanément avec la trame f_i à l'instant 0 sur h , dépend des caractéristiques du flux v_i (T_i , $Smin_i^h$ and $Smax_i^h$). Il est obtenu par le théorème 2 défini dans [68]. Nous redétaillons certains résultats à des fins de complétude du document.

Théorème 2 [68] Dans une file d'attente FIFO, en considérant le scénario pire cas construit par le théorème 1, la première trame f_i générée par un flux v_i ($v_i \in \Gamma^h$) atteint le nœud h simultanément avec k_i^h autres trames. $k_i^h \in \mathbb{N}$ est défini tel que :

$$k_i^h T_i \leq J_i^h < (k_i^h + 1)T_i \quad \text{et donc :} \quad k_i^h = \left\lfloor \frac{J_i^h}{T_i} \right\rfloor \quad (5.4)$$

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

Démonstration : Les deux inégalités sont démontrées séparément $k_i^h T_i \leq J_i^h$ et $J_i^h < (k_i^h + 1)T_i$:

- Considérons la dernière trame de v_i arrivant sur h à l'instant 0. Selon le scénario périodique défini dans le théorème 1 et représenté dans la Fig. 5.4, son instant de génération sur $first_i$ est $0 - Smax_i^h + k_i^h T_i$. Comme elle subit un délai $d \geq Smin_i^h$ pour atteindre h et que son instant d'arrivée dans h est toujours 0, on a :

$$\begin{aligned} -Smax_i^h + k_i^h T_i + d &= 0 \\ -Smax_i^h + k_i^h T_i + Smin_i^h &\leq -Smax_i^h + k_i^h T_i + d = 0 \\ -Smax_i^h + k_i^h T_i + Smin_i^h &\leq 0 \\ k_i^h T_i &\leq Smax_i^h - Smin_i^h = J_i^h \\ k_i^h T_i &\leq J_i^h \end{aligned}$$

- Concentrons-nous sur $f_{i,1}$, la première trame arrivant dans h strictement après 0. Elle est générée à l'instant $0 - Smax_i^h + (k_i^h + 1)T_i$ dans le nœud $first_i$ et subit un délai égal à $Smin_i^h$ pour atteindre h :

$$\begin{aligned} 0 &< -Smax_i^h + (k_i^h + 1)T_i + Smin_i^h \\ Smax_i^h - Smin_i^h &< (k_i^h + 1)T_i \\ J_i^h &< (k_i^h + 1)T_i \end{aligned}$$

Ce qu'il fallait démontrer. □

L'instant d'arrivée de la trame $f_{i,1}$ est déduit du corollaire suivant.

Corollaire 1 [68] *En considérant le scénario pire cas décrit dans le théorème 1, l'instant d'arrivée de la première trame du flux $v_i \in \Gamma^h$ dans le nœud h strictement après l'instant 0 (dénotee $f_{i,1}$ dans la Fig. 5.4) est : $(k_i^h + 1)T_i - J_i^h$.*

Démonstration : C'est un résultat direct des théorèmes 1 et 2, puisque l'instant de génération de $f_{i,1}$ sur son nœud source $first_i$ est égal à $-Smax_i^h + (k_i^h + 1)T_i$ et qu'elle met un délai égal à $Smin_i^h$ pour arriver sur h . □

Dans une file d'attente FIFO, 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 maximisé en considérant le scénario décrit par le théorème 1.

Dans la section suivante, nous exprimons ce scénario pire cas en utilisant la *Request Bound Function* (rbf) pour déterminer l'arriéré de travail maximum dans les files d'attente.

5.3.3 Fonction cumulative et Request Bound Function

La *Request Bound Function* $rbf_i^h(t)$ définit le temps de transmission totale des trames générées par le flux $v_i \in \Gamma^h$ et arrivant durant un intervalle de temps de longueur t dans le nœud h . Nous en déduisons, $W^h(t)$, une borne supérieure de la fonction cumulative des temps de transmission de tous les flux traversant un nœud h durant un intervalle de temps de longueur t :

$$W^h(t) = \sum_{v_i \in \Gamma^h} rbf_i^h(t) \quad (5.5)$$

L'expression de $rbf_i^h(t)$ peut être directement dérivée du scénario pire cas défini dans le théorème 1.

Théorème 3 Dans un nœud FIFO h , la Request Bound Function (rbf) du flux v_i dans un intervalle de temps de durée t modélisant le scénario du théorème 1 est donnée par :

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

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 : Plusieurs cas sont étudiés selon la valeur de t :

- Pour $t = 0$, $(1 + k_i^h)$ trames de v_i arrivent à l'instant 0, comme il a été prouvé dans le théorème 1 :

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

- Pour $0 \leq t < (k_i^h + 1)T_i - J_i^h$, il ne peut y avoir de nouvelle trame puisque :

$$\begin{aligned} rbf_i^h(t) &= \left(1 + \left\lfloor \frac{t + J_i^h}{T_i} \right\rfloor \right) C_i^h \\ &< \left(1 + \left\lfloor \frac{(k_i^h + 1)T_i}{T_i} \right\rfloor \right) C_i^h \\ &< (k_i^h + 2)C_i^h \end{aligned}$$

- La prochaine trame de v_i arrive à l'instant $(k_i^h + 1)T_i - J_i^h$. Finalement, toutes les trames suivantes arrivent avec un écart de T_i (voir la Fig. 5.4).

En résumant ces cas, la formule (5.6) est déduite. $\square$

À noter que si la formule (5.6) est appliquée dans le nœud source, elle correspond à la rbf de tâches non-préemptives sur un système mono-processeur (c-à-d. $J_i^{first_i} = 0$ car $Smin_i^{first_i} = Smax_i^{first_i} = 0$).

La fonction cumulative $W^h(t)$ calculée à partir des fonctions rbf peut être surestimée. Il est possible d'améliorer son calcul en tenant compte de l'effet de sérialisation.

5.3.4 L'effet de sérialisation

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 ne peuvent pas arriver simultanément dans le nœud suivant : leurs instants d'arrivée sont au moins séparés par leur temps de transmission à un taux maximal de r . C'est ce qui est communément appelé *l'effet de sérialisation* [52].

À titre d'exemple, considérons la Fig. 5.4. Les $(k_i^h + 1)$ premières trames de $v_i \in I^h$ ne peuvent arriver simultanément dans h à l'instant 0 puisqu'elles partagent le même

lien d'entrée. Ainsi, avec la prise en compte de la sérialisation, la fonction cumulative des temps de transmission générée par un flux peut être fortement optimisée. À noter que la sérialisation n'a de sens que pour les trames arrivant sur un nœud, pas pour celles générées par le nœud lui-même qui sont envoyées en séquences.

Nous complétons le modèle de nœud afin de distinguer les flux selon leurs liens d'entrée. Ainsi, un nœud h dispose de n liens d'entrée dénotés IP_x^h ($1 \leq x \leq n$), comme illustré Fig. 5.6. De plus, l'ensemble des trames générées par h est dénoté I_0^h . Pour ces trames, il n'y aura pas d'effet de sérialisation par définition.

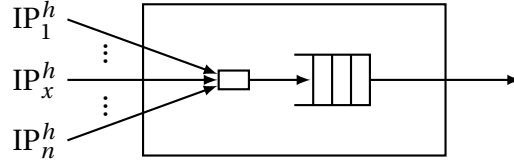


FIGURE 5.6 – Modèle de nœud avec des ports d'entrée.

La fonction cumulative $W^h(t)$ est alors obtenue en sommant les contributions des flux générés sur le nœud h ainsi que celle de chaque lien d'entrée :

$$W^h(t) = \sum_{v_i \in I_0^h} r b f_i^h(t) + \sum_{x=1}^n W_x^h(t) \quad (5.7)$$

Où $W_x^h(t)$ dénote le temps de transmission cumulatif de toutes les trames des flux arrivant dans h via le lien d'entrée IP_x^h durant un intervalle de durée t .

En considérant le débit d'un lien d'entrée physique IP_x^h (avec $1 \leq x \leq n$), il est possible de limiter la charge de travail (temps de transmission cumulés) entrant par ce lien. Le terme $W_x^h(t)$ est alors donné par le minimum entre les deux termes suivants :

- le temps de transmission cumulatif de toutes les trames générées durant un intervalle de temps de durée t , indépendamment de la sérialisation :

$$W_x^h(t) \leq r b f_x^h(t) = \sum_{v_i \in I_x^h} r b f_i^h(t) \quad (5.8)$$

Où I_x^h désigne l'ensemble des flux traversant le nœud h à travers le lien d'entrée IP_x^h (avec $1 \leq x \leq n$);

- la charge maximale qui peut être acheminée par le lien physique IP_x^h (avec $1 \leq x \leq n$) en tenant compte de son débit (cf. lemme 1).

Lemme 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 temps t , dénoté $W_x^h(t)$, est borné par :*

$$W_x^h(t) \leq \text{LinkRate}_{i,x}^h(t) = \frac{r^{h_x-1}}{r^h} t + \max_{v_i \in I_x^h} C_i^h \quad (5.9)$$

Où h_{x-1} dénote le nœud prédécesseur de h via IP_x^h et r^h (resp. r^{h_x-1}) est le taux de transmission du nœud h (resp. $h_x - 1$).

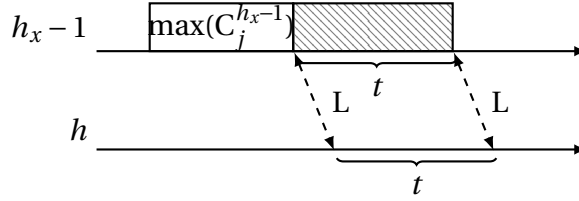


FIGURE 5.7 – La charge de travail maximale arrivant sur un nœud h via le lien d'entrée IP_x^h durant un intervalle de temps de longueur t .

Démonstration : Les nœuds h_x-1 et h sont directement connectés par un lien physique. Par conséquent, la quantité de trafic arrivant dans h via IP_x^h durant un intervalle de longueur t correspond au total des trames ayant été complètement servies au taux r^{h_x-1} , comme illustré Fig. 5.7 :

- (i) durant un intervalle de temps t , le trafic transmis avec un débit de r^{h_x-1} est $r^{h_x-1} \times t$ (la partie hachurée dans la Fig. 5.7) ;
- (ii) dans le pire cas, une trame de taille maximum est arrivée sur h au début de l'intervalle $r^{h_x-1} \times \max_{v_i \in I_x^h} C_i^{h_x-1}$.

En considérant la vitesse du nœud h , r^h , le temps total de transmission pire cas généré par le lien entrant IP_x^h (avec $1 \leq x \leq n$) est : $\frac{r^{h_x-1}}{r^h} \times t$ pour (i) plus le temps de transmission de la trame de taille maximum provenant de IP_x^h , qui est, par définition,

$$\frac{r^{h_x-1}}{r^h} \times \max_{v_i \in I_x^h} C_i^{h_x-1} = \max_{v_i \in I_x^h} C_i^h, \text{ pour le point (ii).} \quad \square$$

En tenant compte de l'effet de sérialisation, la fonction $W_x^h(t)$ est mise à jour dans le théorème 4.

Théorème 4 Le temps total de transmission cumulé atteignant h par le lien entrant IP_x^h ($1 \leq x \leq n$), durant un intervalle de temps t , est borné par :

$$W_x^h(t) = \min \left(r b f_x^h(t), \text{LinkRate}_{i,x}^h(t) \right) \quad (5.10)$$

Démonstration : Ce résultat est une application directe des formules (5.8) et (5.9) définissant deux bornes supérieures de $W_x^h(t)$. $\square$

5.3.5 L'arriéré de travail maximum

Durant un intervalle de temps $[0, t]$, le temps de transmission maximum cumulé $W^h(t)$ arrivant sur un nœud h est déterminé par la formule (5.7) et le théorème 4. Les nœuds sont conservatifs : dès qu'au moins une trame est en attente dans la file, il transmet. L'arriéré de travail maximum dans un nœud h s'obtient donc par la différence entre le temps de transmission maximum cumulé $W^h(t)$ et le temps de transmission cumulé émis par le nœud h durant l'intervalle de temps $[0, t]$, qui correspond à sa longueur t :

$$\text{Bklg}^h = \max_{t \geq 0} (W^h(t) - t) \quad (5.11)$$

5.3.6 Condition suffisante pour l'arrêt du calcul

Pour chaque nœud h , toutes les longueurs d'intervalle $t \geq 0$ doivent être testées pour déterminer l'arriéré de travail maximal $Bklg^h$. Cependant, nous ne pouvons pas tester toutes les valeurs de $t \geq 0$ à l'infini. Mais si pour une valeur de t donnée, l'arriéré de travail total généré durant $[0, t]$ a été totalement transmis en t , un instant oisif (cf. définition 3 du paragraphe 4.3.5) s'est produit et est détecté. 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 (l'instant de démarrage du nœud).

Remarque 1 *Cette condition d'arrêt est valable tant que la politique de service est conservative. C'est-à-dire qu'elle n'insère pas de temps creux si des trames sont prêtes à être transmises.*

En considérant les définitions des fonctions $W^h(t)$ et t , un instant oisif se produit pendant un intervalle de longueur t dans un nœud h si $W^h(t) \leq t$. C'est-à-dire que le temps total de transmission des trames générées durant l'intervalle $[0, t]$, est strictement inférieur à la quantité que peut émettre h durant le même intervalle.

Nous définissons maintenant une condition suffisante garantissant l'existence d'un instant oisif sur le nœud h , en se basant sur sa charge U^h .

Théorème 5 *Soit un nœud FIFO h . Si les trames arrivent selon le scénario défini dans le théorème 1 alors l'existence d'un instant oisif sur h est garantie si la condition suivante sur sa charge locale (U^h) est vérifiée :*

$$U^h = \sum_{v_i \in I^h} \frac{C_i^h}{T_i} < 1 \quad (5.12)$$

Démonstration : Le scénario d'arrivée de chaque flux $v_i \in I^h$ démarre avec une rafale initiale à l'instant 0 ($k_i^h + 1$ trames) et devient périodique à partir de l'instant $(k_i^h + 1)T_i - J_i^h$ (voir théorème 1). A l'instant $\max_{v_i \in I^h} (k_i^h + 1)T_i - J_i^h$, le système est périodique et, en particu-

lier, le scénario d'arrivée des trames sur h est cyclique. A cet instant, les rafales initiales de trames peuvent ne pas avoir été entièrement transmises par h .

La transmission de trames par un nœud peut être modélisée comme un problème d'ordonnancement non-préemptif monoprocesseur. Sur de tels systèmes, si $U^h < 1$ alors il existe au moins un instant oisif pendant un cycle.

Ainsi, si $U^h < 1$ et si les rafales de trames initiales n'ont pas été complètement transmises au début du régime périodique, ce surplus peut être transmis durant les instants oisifs des cycles et ce quelle que soit sa taille. En conclusion, un instant t oisif finira par se produire sur h . $\square$

5.3.7 Formules de calcul et algorithmes

Pour chaque flux $v_i \in I^h$, pour chaque nœud $h \in \mathcal{P}_i$, une borne supérieure du délai de traversée du réseau pire cas jusqu'à sa sortie de h , dénotée R_i^h , est calculée :

$$R_i^h = Smax_i^h + Bklg^h$$

où l'arriéré de travail maximum $Bklg^h$ est déterminé par

$$Bklg^h = \max_{t \geq 0} \{W^h(t) - t\} \quad (5.13)$$

avec $W^h(t)$ définie par la formule (5.7). Le calcul s'arrête dès la détection d'un instant oisif.

Le pseudo-code pour le calcul du délai de bout en bout pire cas d'un ensemble de flux dans un réseau est donné par [68] et mis à jour dans l'algorithme 1 pour inclure les modifications apportées par cette thèse telles que la condition d'arrêt du calcul et la généralisation du modèle réseau (débits différents, modélisation des nœuds). Il est pseudo-polynomial.

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 $v_i \in \Gamma$

begin

```

foreach flux  $v_i \in \Gamma$  do
   $Smin_i^{first_i} \leftarrow 0$ 
   $Smax_i^{first_i} \leftarrow 0$ 
  foreach nœud  $h$  do
    foreach flux  $v_i \in I^h$  do
       $C_i^h \leftarrow \frac{Fmax_i}{r^h}$ 
       $J_i^h \leftarrow Smax_i^h - Smin_i^h$ 
      foreach lien entrant  $IP_x^h$  avec  $x \in \{1, \dots, n\}$  do
         $W_x^h(t) \leftarrow \min \{rbf_x^h(t), \frac{r^{hx-1}}{r^h}t + \max_{v_i \in I_x^h} C_i^h\}$ 
       $W^h(t) \leftarrow \sum_{v_i \in I_0^h} rbf_i^h(t) + \sum_{x=1}^n W_x^h(t)$ 
       $Bklg^h \leftarrow 0$ 
       $t \leftarrow 0$ 
      do
         $Bklg^h \leftarrow \max \{Bklg^h, W^h(t) - t\}$ 
         $t \leftarrow Next(TimeToTest)$ 
      while  $W^h(t) > t$ 
      foreach flow  $v_i \in I^h$  do
        if  $h \neq last_i$  then
           $Smin_i^{h+1} \leftarrow Smin_i^h + C_i^h + L$ 
           $Smax_i^{h+1} \leftarrow Smax_i^h + Bklg^h + L$ 
        else
           $R_i \leftarrow Smax_i^h + Bklg^h$ 

```

Algorithme 1 : Calcul du délai de bout en bout pire cas avec la méthode FA.

Le terme $rbf_x^h(t)$ de la formule (5.10) définit une fonction en escalier pour laquelle, chaque marche correspond à l'arrivée d'une nouvelle trame dans le nœud h . Cependant, pour définir les valeurs de t à tester, il faut également tenir compte du terme de prise en compte de la sérialisation (cf. $LinkRate_{i,x}^h(t)$ de la formule (5.10)). Ainsi, l'algorithme 2 présente la construction de l'ensemble des instants t pour lesquels, dans un nœud h , le calcul du délai doit être effectué par FA.

Input : Un nœud h , traversé par un ensemble de flux, dénoté I^h

Pour chaque flux $v_i \in I$: C_i^h , T_i et J_i^h

Result : *TimeToTest*

begin

TimeToTest $\leftarrow \emptyset$

foreach $x \in \{0, \dots, n\}$ **do**

foreach flux $v_i \in I_x^h$ **do**

foreach $k \in \mathbb{N}$ **do**

$t \leftarrow -J_i + kT_i$

TimeToTest $\leftarrow \text{Add}(t)$

if $x > 0$ **then**

$ser(t) \leftarrow LinkRate_{i,x}^{h-1}(rbf_x^h(t))$

if $ser(t) > t$ **then**

TimeToTest $\leftarrow \text{Add}(ser(t))$

TimeToTest $\leftarrow \text{Sort}(\text{TimeToTest})$

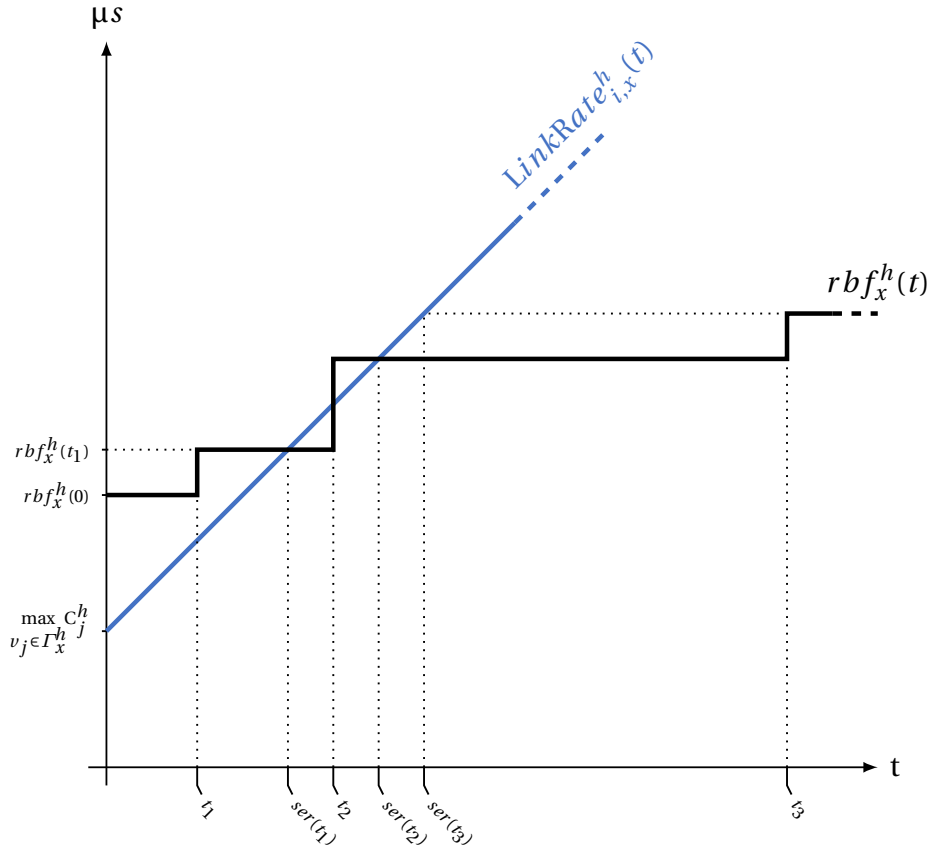
Algorithme 2 : Ensemble des instants t à tester.

Pour tous les flux v_i de I^h , tous les instants d'arrivée d'une nouvelle trame (les instants $-J_i + kT_i$ avec $k \in \mathbb{N}$ selon la définition de la fonction $rbf_i^h(t)$) sont à tester. Ensuite, chaque flux $v_i \in I_x^h$ (avec $x > 0$) rentre dans h via le lien IP_x^h (avec $x \in \{1, \dots, n\}$). Nous calculons alors $ser(t)$ qui est égal à $LinkRate_{i,x}^{h-1}(rbf_{i,x}^h(t))$. Cet instant définit la valeur (unique par construction de $LinkRate_{i,x}^h$) pour laquelle la fonction $LinkRate_{i,x}^h$ croise la nouvelle marche d'escalier de la fonction rbf_x^h . En pratique, nous obtenons :

$$ser(t) = \frac{r_x^h}{r_x^h - 1} \left(rbf_x^h(t) - \max_{v_j \in I_x^h} C_j^h \right). \text{ Si cet instant est situé après } t \text{ alors il doit être testé.}$$

Finalement, nous trions les instants à tester par ordre croissant.

Ainsi, considérons l'exemple illustré dans la Fig. 5.8. Pour un lien entrant IP_x^h d'un nœud h , les courbes des fonctions $rbf_x^h(t)$ et $LinkRate_{i,x}^h(t)$ ont été tracées. A l'instant t_1 une nouvelle trame arrive dans h , cet instant est à analyser par définition. Cette nouvelle marche d'escalier intercepte la fonction $LinkRate_{i,x}^h(t)$ à l'instant $ser(t_1)$ avec $ser(t_1) > t_1$ donc $ser(t_1)$ doit aussi être testé. Le même raisonnement est aussi valable pour t_2 (et $ser(t_2)$). Pour finir, t_3 correspond à l'arrivée d'une nouvelle trame et est à tester. Cependant, l'instant $ser(t_3)$ (instant d'intersection de la marche d'escalier avec $LinkRate_{i,x}^h(t)$) est inférieur à t_3 et n'a pas à être analysé.

FIGURE 5.8 – Les instants t à tester.

5.4 Expérimentations

Parmi les méthodes d'analyses présentées dans le paragraphe 4.3, nous comparons, uniquement, la méthode FA avec le Network Calculus (NC) et la méthode des trajectoires (TA) sur de configurations AFDX. En effet, la simulation ne détermine pas une borne supérieure garantie sur le délai et le temps de calcul nécessaire au *Model Checking* n'est pas raisonnable (polynomial ou pseudo-polynomiale). De plus, ni la méthode holistique ni la méthode *Compositional Performance Analysis* n'ont été adaptées à l'analyse d'un réseau AFDX complet (cf. paragraphe 4.3).

NC est basé sur la version [52] 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" paradigme [32]). La version implémentée de TA est celle qui a été corrigée [72] suite au problème d'optimisme.

Nous commençons ces expérimentations par présenter la transposition de l'AFDX dans notre modèle réseau dans le paragraphe 5.4.1 et nous détaillerons le calcul de la borne supérieure sur le délai de bout en bout d'un flux dans le paragraphe 5.4.2. Pour finir, nous appliquons la nouvelle version de FA et les deux autres méthodes sur deux configurations différentes pour comparer les bornes sur le délai de bout en bout : un cas d'étude de quelques nœuds et flux dans le paragraphe 5.4.3 et une configuration industrielle dans le paragraphe 5.4.4. À priori, la précision des bornes sur le délai ne peut pas être une métrique pour comparer les deux versions de FA car le principe de calcul n'a pas

été modifié. La différence réside principalement dans la généralisation du modèle réseau et de la condition d'arrêt du calcul.

5.4.1 Transposition de l'AFDX au modèle réseau de FA

La méthode FA ayant été développée pour la validation des réseaux AFDX, la transposition de ce protocole dans notre modèle (*cf.* paragraphe 5.2) est naturelle.

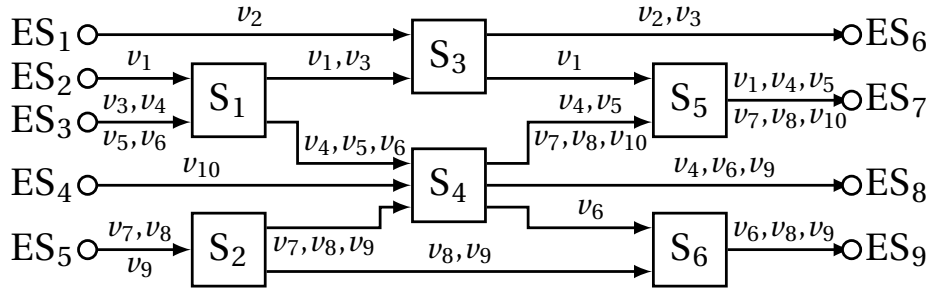


FIGURE 5.9 – Une configuration AFDX.

Les commutateurs AFDX récupèrent les trames en mode *store & forward* : la trame est totalement récupérée avant d'être analysée et routée selon ses données de configuration. Le délai dit de "commutation" (analyse et routage) est fixé à $16\mu s$ (*cf.* paragraphe 3.5.1 et [52]), il est reporté dans la latence technologique $L = 16\mu s$.

Les nœuds de notre modèle correspondent aux points de multiplexage du réseau. Ainsi, chaque port de sortie d'un commutateur est modélisé par un nœud. Prenons l'exemple du commutateur S_3 de la configuration illustrée dans la Fig. 5.9. S_3 a deux ports de sortie (l'un à destination de ES_6 et l'autre à destination de S_5). S_3 se modélise alors en deux nœuds nommés respectivement S_{31} et S_{32} (*cf.* Fig. 5.10). Les ports d'entrée du commutateur sont transposés aux nœuds modélisant les ports de sortie. Les ES ne disposent pas de ports 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 3.5.1) par une taille de trame maximale F_{max} et le délai minimum entre la génération de deux trames est défini par son BAG ($T_i = BAG$). Les VLs sont émis par un ES source et suivent un chemin statique. Cependant, les VLs peuvent avoir, par définition, plusieurs destinataires à la différence des flux de notre modèle. Ainsi, 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. 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. 5.9 est représentée par la Fig. 5.10.

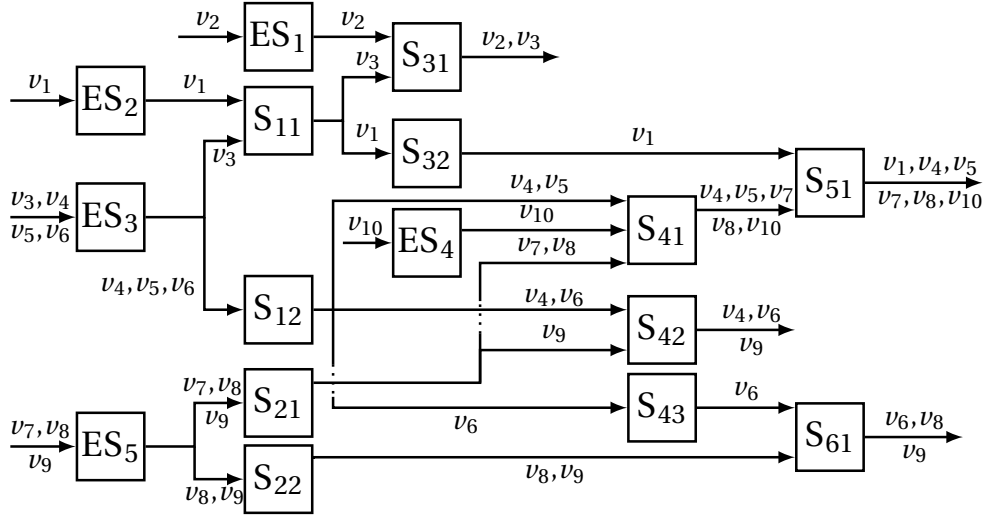


FIGURE 5.10 – La modélisation de la configuration représentée par la Fig. 5.9.

	Caractéristiques		Délai de bout en bout	
	$Fmax_i$ (bits)	T_i (μs)	NC (μs)	FA (μs)
v_1	12144	4000	4270.48	3577.04
v_2	12144	4000	3678.54	3270.40
v_3	8256	8000	5404.05	4957.60
v_4 (ES ₇)	8256	8000	6709.98	6007.60
v_4 (ES ₈)			2955.28	2901.76
v_5 (ES ₇)	8256	8000	6709.98	6007.60
v_6 (ES ₈)	2048	1000	2955.28	2901.76
v_6 (ES ₉)			3009.81	2880.26
v_7	2048	1000	4447.91	3752.88
v_8 (ES ₇)	512	1200	4447.91	3752.88
v_8 (ES ₉)			973.43	796.42
v_9 (ES ₈)	2304	500	693.20	647.04
v_9 (ES ₉)			973.43	796.42
v_{10}	512	250	3915.79	3232.56

TABLEAU 5.1 – Les caractéristiques des flux et leurs délais de bout en bout pire cas (par NC et FA)

5.4.2 Exemple de calcul en FIFO

Afin d'illustrer le calcul des bornes sur les délais par la méthode FA, nous analysons le délai de bout-en-bout du flux v_9 , à destination de ES₉, de la configuration dépeinte Fig. 5.9 et modélisée Fig. 5.10. v_9 est généré sur ES₄ et traverse successivement les nœuds S₂₂ et S₆₁ (le ES destination ES₉ n'est pas compté dans le calcul). La vitesse de service des nœuds est de 10Mbps pour ES₄ et S₂₂ ainsi que 100Mbps pour S₆₁. Les caractéristiques des flux sont résumées dans le tableau 5.1.

Sur le nœud source, v_9 croise les flux v_7 et v_8 . Par définition, du nœud source, $Sm\dot{in}_i^{ES_4} =$

$Smax_i^{ES_4} = 0$ (pour $i \in \{7, 8, 9\}$). Nous calculons dans ce qui suit l'arriéré de travail maximum $Bklg^{ES_4}$ donné par formule (5.11). Le maximum est atteint en $t = 0$ avec :

$$W^{ES_4}(0) = rbf_9^{ES_4}(0) + rbf_7^{ES_4}(0) + rbf_8^{ES_4}(0) = 486,4$$

Sur le nœud suivant, S_{22} , v_8 et v_9 sont les deux seuls flux. Nous commençons par calculer leurs gignes respectives :

$$\begin{aligned} J_8^{S_{22}} &= Smax_8^{S_{22}} - Smin_8^{S_{22}} = 502,4 - 67,2 = 435,2 \\ J_9^{S_{22}} &= Smax_9^{S_{22}} - Smin_9^{S_{22}} = 502,4 - 246,4 = 256 \end{aligned}$$

Les deux flux v_8 et v_9 partagent le même lien d'entrée de S_{22} , la fonction $W^{S_{22}}$ est donc :

$$\begin{aligned} W^{S_{22}}(t) &= \min \left(\frac{10t}{10} \cdot \frac{r^{ES_4}}{r^{S_{22}}} + C_9^{ES_4}, \left(1 + \left\lfloor \frac{t + J_8^{S_{22}}}{T_8} \right\rfloor \right) C_8^{S_{22}} + \left(1 + \left\lfloor \frac{t + J_9^{S_{22}}}{T_9} \right\rfloor \right) C_9^{S_{22}} \right) \\ &= \min \left(t + 23,04, \left(1 + \left\lfloor \frac{t + 435,2}{1200} \right\rfloor \right) 51,2 + \left(1 + \left\lfloor \frac{t + 256}{500} \right\rfloor \right) 23,04 \right) \end{aligned}$$

$Bklg^{S_{22}}$ est déterminé pour $t = 0$ et nous obtenons $Bklg^{S_{22}} = 230,4$.

Sur le dernier nœud, S_{61} , v_9 croise le flux v_6 venant d'un autre lien et v_8 avec les gignes respectives : $J_9^{S_{61}} = 256$, $J_6^{S_{61}} = 2538,88$ et $J_8^{S_{61}} = 614,4$. La formule de $W^{S_{61}}(t)$ est :

$$\begin{aligned} W^{S_{61}}(t) &= \min \left(\frac{10t}{100} \cdot \frac{r^{S_{22}}}{r^{S_{61}}} + C_9^{S_{61}}, \left(1 + \left\lfloor \frac{t + J_8^{S_{61}}}{T_8} \right\rfloor \right) C_8^{S_{61}} + \left(1 + \left\lfloor \frac{t + J_9^{S_{61}}}{T_9} \right\rfloor \right) C_9^{S_{61}} \right) \\ &\quad + \min \left(t \cdot \frac{r^{S_{43}}}{r^{S_{61}}} + C_6^{S_{61}}, \left(1 + \left\lfloor \frac{t + J_6^{S_{61}}}{T_6} \right\rfloor \right) C_6^{S_{61}} \right) \end{aligned}$$

$Bklg^{S_{61}}$ est obtenu à l'instant 40,96 et est égal à 47,62 μs . Finalement, le délai de bout en bout pire cas R_9 est déduit :

$$R_9 = Smax_9^{S_{61}} + L + Bklg^{S_{61}} = 732,8 + 16 + 47,62 = 796,42 \mu s$$

Où $L = 16 \mu s$ définit la latence technologique.

La borne obtenue par la méthode NC est donnée dans le tableau 5.1.

5.4.3 Cas d'étude en FIFO

Nous analysons la configuration AFDX représentée par la Fig. 5.9 détaillée dans le paragraphe 5.4.2. Les caractéristiques des flux sont données dans le tableau 5.1.

Les délais de bout en bout pire cas obtenus par TA ne figurent pas dans le tableau 5.1, car la configuration AFDX représentée par la Fig. 5.9 ne peut pas être étudiée avec TA.

En effet, la charge globale du trafic ($\sum \frac{C_i^h}{T_i}$) de certains flux (e.g. v_7) sur leur chemin est strictement supérieure à 1. Par conséquent, le calcul de TA ne converge pas.

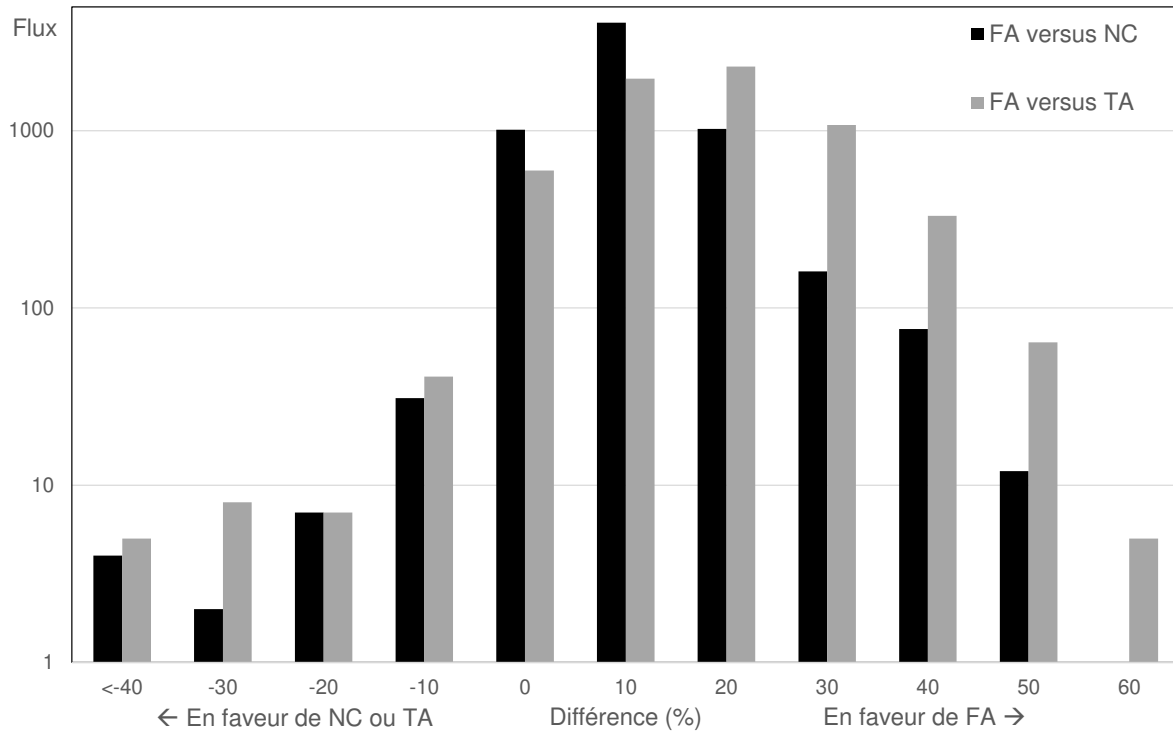


FIGURE 5.11 – Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA).

Les délais calculés par NC et FA sont présentés dans les deux dernières colonnes du tableau 5.1. Les délais de bout en bout pire cas sont différenciés pour chaque ES destinataire dans le cas de flux multicasts. Par exemple, deux délais sont déterminés par chaque méthode pour le flux ν_4 : un délai pour le nœud destinataire ES_7 et un deuxième pour ES_8 .

Les délais obtenus par FA sont inférieurs à ceux obtenus en utilisant NC. La différence entre les deux méthodes atteint même 22.2% pour le flux ν_8 (à destination de ES_9).

5.4.4 Configuration industrielle en FIFO

A présent, nous appliquons les trois méthodes (FA, NC and TA) sur une configuration AFDX industrielle utilisée pour l'A380 et présentée dans [21]. Elle est composée de 96 ESs reliés entre eux par 983 VLs multicast (soit 6412 flux) et 8 commutateurs (24 ports chacun). Le temps d'exécution de FA sur cette configuration 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 statistiques sont présentés dans la Fig. 5.11 sous forme d'histogramme de fréquence.

Pour chaque flux (chemin de chaque VL), 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, la taille de la barre verticale correspond au nombre de flux (chemins de VLs) qui ont une telle différence de délai : une valeur positive signifie que FA a obtenu un délai plus serré (plus petit) que l'autre approche et inversement pour des

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 obtenus en utilisant FA plus petit de 20 à 30%, que ceux déterminés par TA.

Dans [23], 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, aucune des trois méthodes n'est dominante pour cette configuration dès lors que nous obtenons des valeurs positives et négatives dans l'histogramme dans la Fig. 5.11. 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. 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.

5.5 Conclusion

Dans ce chapitre, nous avons présenté la méthode *Forward end-to-end delay Analysis* (FA) et le modèle de réseau sur lequel elle s'applique. La méthode a été définie dans [68]. Cependant, nous avons complété la méthode FA en reformulant et/ou complétant, démonstrations formelles à l'appui, certains points. Nous avons, ainsi, reformulé le scénario pire cas d'arrivée des trames (cf. théorème 1) et généralisé le modèle de nœud afin qu'il puisse, simultanément, générer et transférer des trames. Nous avons, également, permis des configurations comprenant des liens physiques avec des débits différents. Finalement, la condition d'arrêt du calcul a été généralisée à n'importe quelle politique d'ordonnancement conservative.

Pour conclure ce chapitre, FA a été comparée à deux approches existantes (Network Calculus et l'approche par Trajectoires) sur un cas d'étude d'une part et une configuration industrielle d'autre part. Nous avons démontré que FA était applicable sur des configurations industrielles. Nous avons aussi démontré que la méthode FA est pertinente pour la validation de tels réseaux. En effet, les bornes garanties déterminées par FA sont inférieures à celles de NC. FA n'a pas de contrainte sur la charge globale des flux, à la différence de TA (cf. paragraphe 4.3.5).

Ces travaux ont donné lieu à une conférence internationale [66] et une revue internationale [30]. Ils ont servi de base pour l'extension de FA; notamment à la politique de service FPFIFO détaillée dans le chapitre suivant.

Chapitre 6

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	Le calcul du délai pire cas d'une trame en FP/FIFO	82
6.3.1	Extension de FA, sans sérialisation, à FP/FIFO	82
6.3.2	L'effet de sérialisation en FP/FIFO	85
6.4	Expérimentations en FP/FIFO	91
6.4.1	Exemple de calcul en FP/FIFO	92
6.4.2	Cas d'étude en FP/FIFO	93
6.4.3	Configuration industrielle en FP/FIFO	94
6.5	Conclusion	96

6.1 Introduction

Dans le chapitre 5, la méthode FA a été présentée pour des réseaux de paquets utilisant la politique de service FIFO. Cependant, les industriels souhaitent 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. La politique de service FIFO ne permet pas cette différenciation car la priorité d'une trame est dynamique et dépend de son ordre d'arrivée dans un nœud. La différenciation entre les flux doit être gérée par un mécanisme statique qui garantit toujours une priorité plus forte aux flux critiques. Cette stratégie est définie par la politique *Fixed Priority/FIFO* (FP/FIFO). Un niveau de priorité statique est attribué à chaque flux et à l'intérieur de chaque classe, les flux sont toujours ordonnancés selon la stratégie FIFO. Par exemple, le réseau AFDX de l'A380 implémente deux classes de trafic : *HIGH* (*resp.* *LOW*) pour les flux plus (*resp.* moins) prioritaires. À noter, que FIFO est un cas particulier de FP/FIFO pour lequel le même niveau de priorité est attribué à tous les flux du réseau.

Le Network Calculus et la méthode des Trajectoires ont déjà implémenté la politique FP/FIFO. Dans la suite de ce chapitre, nous complétons le modèle réseau décrit dans le paragraphe 6.2 pour prendre en compte la politique de service FP/FIFO. Les formules de FA et, plus précisément, la fonction cumulative $W^h(t)$ sont, ensuite, étendues à la politique FP/FIFO dans le paragraphe 6.3. Finalement, une comparaison avec NC et TA est détaillée dans le 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 listées dans le paragraphe 5.2 sont identiques en FP/FIFO.

Dans ce chapitre, nous considérons la politique d'ordonnancement FP/FIFO avec $n \in \mathbb{N}^*$ niveaux de priorité différents. En plus, de $Fmax_i$, T_i et $\mathcal{P}_i$, un flux $v_i \in \Gamma$ est caractérisé par son niveau de priorité statique P_i . À noter que le niveau de priorité P_i le plus petit correspond à la plus haute priorité et les flux de même priorité font partie de la même classe de flux.

Le modèle du nœud est mis à jour afin de dénombrer dorénavant n files d'attente, une par niveau de priorité (*cf.* Fig. 6.1). Dans un nœud h , un flux $v_i \in \Gamma^h$ peut être retardé par trois ensemble de flux :

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

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

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

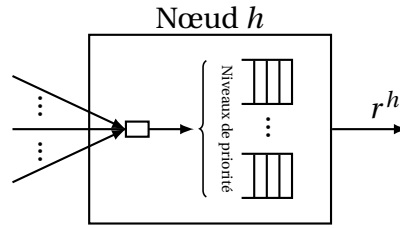


FIGURE 6.1 – Modèle nœud en FP/FIFO.

Remarque 2 En plus de ces ensembles, pour un flux v_i , sur un nœud h , nous définissons par shp_i^h l'union de sp_i^h et hp_i^h :

$$shp_i^h = sp_i^h \cup hp_i^h$$

Remarque 3 La fonction $W^h(t)$ a été définie dans le paragraphe 5.3.3 comme une borne supérieure de la fonction cumulative des temps de transmission de tous les flux traversant un nœud h durant un intervalle de temps de longueur t . Cependant, cette fonction qui était générale pour tous les flux d'un nœud en FIFO est individualisée en $W_i^h(t)$ pour chaque flux $v_i \in I^h$ en FP/FIFO et de même pour l'arriéré total $Bklg_i^h$.

6.3 Le calcul du délai pire cas d'une trame en FP/FIFO

L'extension des formules de FA au FP/FIFO se fait en deux temps : la version non sérialisée dans le paragraphe 6.3.1 puis en considérant l'effet de sérialisation (cf. formule (5.7) et le théorème 4) dans le paragraphe 6.3.2.

6.3.1 Extension de FA, sans sérialisation, à FP/FIFO

La méthode FA se base sur le concept de période d'activité définie en ordonnancement, afin de calculer les délais. Nous devons, par conséquent, étendre les définitions de l'instant oisif (défini dans la définition 3) et de la période d'activité (définie dans la définition 2) aux priorités statiques :

Définition 7 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 avant cet instant, ont été complètement servies. Autrement dit, il n'y a pas de trame de priorité au moins P_i en attente dans une des files du nœud h .

Définition 8 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 .

Nous étendons, maintenant, le théorème 1 du paragraphe 5.3 à la stratégie FP/FIFO. A la différence du cas FIFO, la longueur de la période d'activité est dépendante du flux v_i analysé (et de son temps de transmission maximum C_i^h).

Théorème 6 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 $v_j \in I^h$ 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 $\max_{v_j \in I^h} (C_j^h)$, 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 maximisée 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 retarde la transmission de f_i dans h depuis l'instant 0.

Démonstration :

(i) a déjà été démontré dans le théorème 1. Les autres items du Théorème maximisent les interférences en fonction du niveau de priorité des flux.

Nous analysons une période d'activité de niveau P_i démarrant à l'instant 0. La politique de service non-préemptive implique qu'une seule trame de priorité inférieure à P_i arrivant sur h juste avant l'instant 0 peut retarder f_i . L'interférence correspondante, $\max_{v_j \in I^h} (C_j^h) - \varepsilon$, est maximisée en considérant la trame avec le plus long temps de transmission (parmi les moins prioritaires) arrivant sur h juste avant l'instant 0 (à $0 - \varepsilon$, avec $\varepsilon \rightarrow 0^+$). Ce qui démontre (ii).

(iii) est démontré en considérant que la trame f_i est également retardée par toutes les trames générées par les flux de sp_i^h qui sont arrivées dans h avant elle (avant l'instant t). Par conséquent, cette interférence est à maximiser durant l'intervalle de temps $[0, t]$.

(iv) se démontre de la même façon avec les flux de hp_i^h . Toute trame générée par un flux de hp_i^h arrivant sur h avant le début de la transmission de f_i , exactement à l'instant $dep_i^h(t)$ tel que $dep_i^h(t) = W_i^h(t) - C_i^h$, la retarde. Cette interférence est à maximiser sur l'intervalle $[0, dep_i^h(t)[$. $\square$

Nous exprimons dans le théorème 7, la plus longue période d'activité déduite du théorème 6, à l'aide des *Request Bound Functions* (rbf). L'interférence pour un flux v_i est différenciée selon le niveau relatif de priorité des flux de I^h . Ainsi, la fonction $W_i^h(t)$ est décomposée en une somme de trois fonctions pour respectivement les flux de plus petite, de même et de plus haute priorité :

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

Théorème 7 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 (6.2), si la condition sur la charge $\sum_{v_j \in I^h} \frac{C_j}{T_j} < 1$ est respectée :

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

Où

$$\begin{aligned} WLP_i^h(t) &= \max_{v_j \in lp_i^h} C_j^h \\ WSP_i^h(t) &= \sum_{v_j \in sp_i^h} rbf_j^h(t) \\ WHP_{i,t}^h(k) &= \sum_{v_j \in hp_i^h} rbf_j^h(W_{i,t}^h(k) - C_i^h). \end{aligned}$$

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

Démonstration : L'interférence des flux appartenant à lp_i^h , $WLP_i^h(t)$, est directement obtenue à partir de l'élément (i) du théorème 6.

Pour les flux de même (sp_i^h) et de plus haute priorité (hp_i^h), nous utilisons les fonctions rbf_j^h définies dans la formule (5.6), pour maximiser les interférences. Ainsi, pour les flux de sp_i^h (resp. hp_i^h), les fonctions rbf_j^h sont utilisées sur l'intervalle $[0, t]$ (resp. $[0, W_{i,t}^h(t) - C_i^h]$).

Cependant, ce dernier terme introduit une cyclicité dans le calcul. En effet, la longueur de la période d'activité est dépendante de l'interférence des flux de plus haute priorité qui est elle-même dépendante de la longueur de la période d'activité. Le calcul est initialisé à C_i^h car f_i appartient nécessairement à la période d'activité de niveau P_i et est itéré jusqu'à ce qu'un point fixe soit obtenu. L'existence de ce point fixe est garantie si la charge du nœud h est strictement inférieure à 1.

□

L'arriéré temporel pire cas, subi par la trame f_i générée par le flux v_i dans le nœud h , dénoté $Bklg_i^h$, est ainsi défini par :

$$Bklg_i^h = \max_{t \geq 0} \{W_{i,t}^h(t) - t\} \quad (6.3)$$

Cependant, le théorème 7 fournit des résultats pessimistes car l'effet de sérialisation n'est pas pris en compte. C'est l'objectif du paragraphe suivant.

6.3.2 L'effet de sérialisation en FP/FIFO

Comme dans le paragraphe 5.3.4, en plus de l'ensemble I_0^h , le nœud h dispose d'un ensemble de n liens d'entrée IP_x^h (avec $1 \leq x \leq n$). Les notations sur les ensembles de flux en fonction du niveau de priorité sont étendues pour tenir compte du lien entrant par lequel ils atteignent le nœud h :

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

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

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

$shp_{i,x}^h = sp_{i,x}^h \cup hp_{i,x}^h$.

Par extension, $sp_{i,0}^h$ désigne l'ensemble des flux de priorité P_i générés par le nœud h et nous décomposons l'interférence $WSP_i^h(t)$ en différenciant les flux par lien entrant :

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

L'objectif est d'utiliser l'effet de sérialisation (introduit dans le paragraphe 5.3.4) afin de limiter l'arriéré entrant par chacun de ces liens.

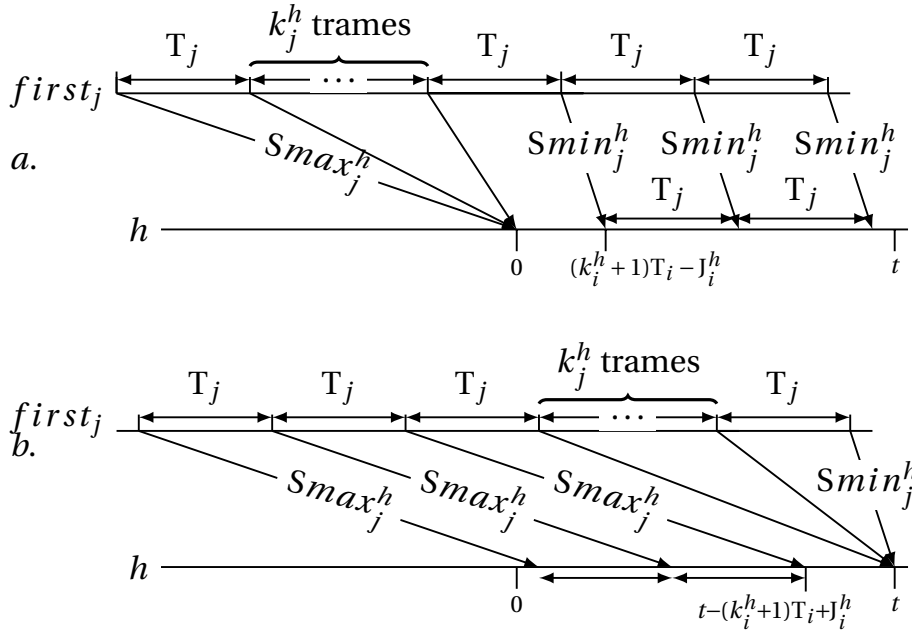


FIGURE 6.2 – Les scénarios d'arrivée pire cas des trames (tel que $v_j \in I^h$) où la rafale de $(1 + k_j^h)$ trames arrive à l'instant 0 (a) ou à l'instant t (b).

Dans FA, les interférences générées par les flux, retardant une trame f_i dans un nœud h , sont exprimées via les fonctions rbf_j^h (voire le paragraphe 5.3.3). Cependant, ces fonctions ne fournissent aucune information sur les instants d'arrivée de trames dans un nœud, elles donnent seulement le nombre de trames qui interfèrent, dans le pire cas,

	ν_1	ν_2	ν_3	ν_4
C_i (μ s)	10	10	40	10
T_i (μ s)	50	50	1000	1000
J_i^h (μ s)	100	100	0	0
P_i	1	2	3	2

TABLEAU 6.1 – Les caractéristiques des flux de l'exemple dans la Fig. 6.3

avec f_i . Ainsi, deux scénarios possibles d'arrivée de trames sont illustrés dans la Fig. 6.2. Ils correspondent 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).

Il a été formellement démontré respectivement dans [30] (et dans le chapitre 5) pour le scénario (a) et [68] pour (b), que ces deux scénarios correspondent à la même interférence exprimée par les fonctions rbf_j^h . 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 .

Néanmoins, avec la politique FP/FIFO, le scénario d'arrivée des trames influe sur le calcul de l'arriéré. Ainsi si la trame f_i arrivant dans h à l'instant t , peut être retardée par des trames de priorité supérieure même si elles arrivent après t , alors que les trames de même priorité ne peuvent retarder f_i que si elles arrivent avant t . Par conséquent, l'ordre d'arrivée des trames générées par sp_i^h et hp_i^h peut impacter le calcul du temps d'attente pire cas de f_i .

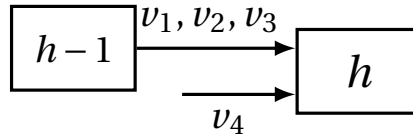
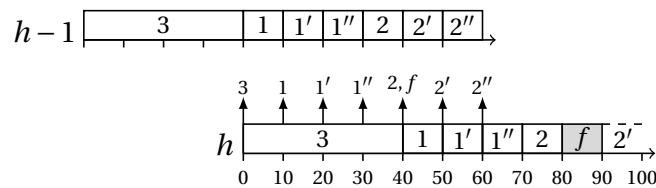


FIGURE 6.3 – Un exemple de configuration AFDX pour illustrer l'impact de l'ordre d'arrivée des trames avec la politique FP/FIFO.

Nous illustrons l'influence des instants d'arrivée de trame sur le temps d'attente pire cas d'une trame dans un nœud avec la configuration représentée dans la Fig. 6.3 et composée de quatre flux. Les flux ν_1 , ν_2 et ν_3 arrivent dans h à travers le même lien entrant, relié au nœud $h-1$. De plus, ν_4 atteint h par un autre lien. La latence technologique est fixée à $L = 0$ et tous les liens ont le même débit donc le temps de transmission, dénoté C_i , d'une trame de chaque flux ν_i est identique pour tous les nœuds. Nous analysons le délai de bout en bout de la trame f générée par le flux ν_4 et qui arrive dans h à l'instant $t = 40 \mu$ s.


 FIGURE 6.4 – Le scénario d'arrivée en plaçant la rafale de trames à l'instant $t = 0$.

Considérons le scénario de la Fig. 6.2.(a) pour lequel la rafale de trames arrive dès le début de l'intervalle $[0, 40]$. Comme dépeint par la Fig. 6.4, quatre trames retardent f dans le nœud h : trois sont plus prioritaires car générées par v_1 (elles arrivent les unes après les autres car la gigue est importante), plus une trame de même priorité générée par v_2 . Notons que la rafale de trames arrivant à partir de $h - 1$ est sérialisée par le lien entrant. Ainsi, les trames $2'$ et $2''$ générées par le même flux v_2 (donc de même priorité de f) arrivent sur h après l'instant $40 \mu s$ ne peuvent donc pas retarder f . Finalement, f est complètement transmise à $90 \mu s$.

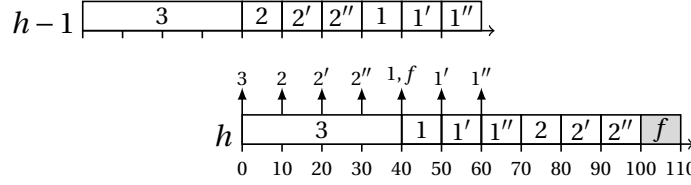


FIGURE 6.5 – Le scénario d'arrivée de la rafale de trames arrive à l'instant 0 pour les flux de sp_i^h et à l'instant $t = 40 \mu s$ pour ceux de hp_i^h .

Si nous retardons, maintenant, l'arrivée des trames générées par v_1 à la fin de l'intervalle $[0, 40]$ comme illustré par la Fig. 6.2.(b). L'arrivée des trames de v_1 étant retardée, un total de six trames interfèrent avec f . En effet, deux trames supplémentaires de v_2 arrivent avant f . La fin de transmission de f est décalée à $110 \mu s$.

En considérant les différents scénarios d'arrivée de trames, nous définissons une borne supérieure moins pessimiste (comparée à celle calculée par le théorème 1) de la plus longue période d'activité, en prenant compte de l'effet de sérialisation.

Théorème 8 *Pour une trame f_i générée par un flux v_i arrivant dans un FP/FIFO nœud 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 $v_j \in I^h$ sont générées périodiquement chaque T_j dans leur nœud source ($f_i r s t_j$);*
- (ii) *l'arrivée de la plus longue trame générée par l'ensemble des flux de priorité inférieure $\max_{v_j \in I^h} (C_j^h)$, se produit juste avant l'instant 0 (à l'instant $0 - \epsilon$, avec $\epsilon \rightarrow 0^+$);*
- (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 globalement 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.

Chaque lien entrant IP_x^h (avec $1 \leq x \leq n$) est partagé, notamment, entre des flux de $sp_{i,x}^h$ et de $hp_{i,x}^h$. Pour qu'une trame de $sp_{i,x}^h$ arrive sur le nœud h et soit comptabilisé dans l'arriéré, il faut que les trames plus prioritaires (de $hp_{i,x}^h$) laissent un créneau libre durant l'intervalle $[0, t]$.

Les flux v_j de $hp_{i,x}^h$ génèrent des trames au rythme maximum d'une trame toutes les T_j unités de temps. L'arriéré maximum pour chaque flux est maximisé sur l'intervalle

$[0, dep_i^h(t)[$ (cf. théorème 6). Cependant, nous devons distinguer les trames arrivées avant t (car $0 \leq t < dep_i^h(t)$) de celles arrivées après. En effet, toute trame générée par un flux de $hp_{i,x}^h$ et arrivant sur h durant l'intervalle $[0, t[$ occupe un créneau qui devient indisponible pour les trames générées par les flux de $sp_{i,x}^h$. Ainsi, afin de maximiser la longueur de la période d'activité, les arrivées des trames de $hp_{i,x}^h$ doivent, autant que possible, être retardées après l'instant t .

□

Nous déterminons, dans le lemme suivant, le nombre minimum de trames à décomposer dans l'intervalle $[0, t[$.

Lemme 2 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 $BHP_j^h(t)$ est égal à :

$$BHP_j^h(t) = rbf_j^h(t) - rbf_j^h(0)$$

Démonstration : Le résultat est direct car :

- $rbf_j^h(t)$ dénote le nombre total de trames générées par v_j et arrivées sur h dans l'intervalle $[0, t[$;
- $rbf_j^h(0)$ exprime exactement la taille de la rafale quel que soit son instant de génération (cf. Fig. 6.2).

□

Nous modifions, maintenant, le calcul de point fixe du théorème 7 à partir de la période d'activité définie dans le théorème 8 afin d'exprimer un arriéré de travail pire cas en prenant compte la sérialisation des flux.

Théorème 9 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 (6.4) si la condition sur la charge $\sum_{v_j \in \Gamma^h} \frac{C_j}{T_j} < 1$ est respectée.

$$\begin{cases} W_{i,t}^h(0) = C_i^h \\ W_{i,t}^h(k+1) = WLP_i^h + WSP_i^h(t) + WHP_{i,t}^h(k) \end{cases} \quad (6.4)$$

Où

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

avec

$$\begin{aligned}
 WSP_{i,x}^h(t) &= \min \left(rbf_{i,x}^h(t), \text{LinkRate}_{i,x}^h(t) \right) \\
 rbf_{i,x}^h(t) &= \sum_{v_j \in sp_{i,x}^h} rbf_j^h(t) \\
 \text{LinkRate}_{i,x}^h(t) &= \frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h - \text{BHP}_{i,x}^h(t) \\
 \text{BHP}_{i,x}^h(t) &= \sum_{v_j \in hp_{i,x}^h} \left(rbf_j^h(t) - rbf_j^h(0) \right).
 \end{aligned}$$

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

Démonstration : L'initialisation et l'interférence des flux moins prioritaires $WLP_i^h(t)$ et plus prioritaires $WHP_i^h(t)$ de la formule (6.4) restent identiques à celles de la formule (6.2).

L'effet de sérialisation ne concerne que l'intervalle $[0, t]$ pour les flux de priorité P_i ou de priorité supérieure. En effet, une seule trame de priorité inférieure retarde f_i et elle arrive seule au tout début de l'intervalle. De plus, les trames retardant f_i et arrivant sur h après l'instant t sont de priorités supérieures. Il n'y a pas d'effet de sérialisation car quel que soit le nombre de ces trames, si elles arrivent dans h avant l'instant $dep_i^h(t) = W_i^h(t) - C_i^h$, elles retarderont f_i .

La contribution des flux de priorité P_i et de priorités supérieures a, ainsi, à 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 8 et le lemme 2, l'interférence des flux plus prioritaires peut être reformulées sous la forme suivante :

$$\sum_{v_j \in hp_i^h} \left(\underbrace{\text{BHP}_j^h(t)}_{[0,t[} + \underbrace{rbf_j^h(W_i^h(t) - C_i^h)}_{[t, dep_i^h(t)[} - \text{BHP}_j^h(t) \right)$$

Si nous nous focalisons sur l'arriéré généré par les flux de priorité P_i ou supérieure entrant par le lien IP_x^h ($1 \leq x \leq n$) durant l'intervalle $[0, t[$, nous obtenons selon le théorème 8 :

$$\underbrace{\sum_{v_j \in sp_{i,x}^h} rbf_j^h(t)}_{=rbf_{i,x}^h(t)} + \underbrace{\sum_{v_j \in hp_{i,x}^h} \text{BHP}_j^h(t)}_{=\text{BHP}_{i,x}^h(t)}$$

À partir de la formule (5.9), nous pouvons limiter la formule ci-dessus par un min, afin de prendre en compte la sérialisation des flux. Ainsi, en sommant sur tous les liens entrants, nous obtenons :

$$\sum_{x=1}^n \min \left(rbf_{i,x}^h(t) + \text{BHP}_{i,x}^h(t), \frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h \right)$$

Si nous analysons l'intervalle $[t, dep_i^h(t)[$, les termes restants sont :

$$\sum_{v_j \in hp_i^h} rbf_j^h(dep_i^h(t)) - \underbrace{\sum_{v_j \in hp_i^h} BHP_j^h(t)}_{= \sum_{x=1}^n BHP_{i,x}^h(t)}$$

$WSP_{i,x}^h(t)$ peut alors s'écrire :

$$WSP_{i,x}^h(t) = \min \left(rbf_{i,x}^h(t) + BHP_{i,x}^h(t), \frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h \right) - BHP_{i,x}^h(t)$$

L'expression de $WSP_i^h(t)$ peut être factorisée :

$$WSP_i^h(t) = \min \left(rbf_{i,x}^h(t), \frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h - BHP_{i,x}^h(t) \right)$$

D'où le résultat du théorème 9, en intégrant le calcul de point fixe. $\square$

Corollaire 2 Pour une configuration dans laquelle tous les flux ont la même priorité, l'expression de l'arriéré de travail pire cas retardant une trame f_i générée par le flux v_i , définie en FP/FIFO par la formule (6.2) coïncide avec celle définie pour la politique FIFO par la formule (5.10).

Démonstration :

Le résultat est direct car, par définition, nous avons $lp_i^h = hp_i^h = \emptyset$ et donc $WLP_i^h(t) = WHP_{i,t}^h(k) = BHP_i^h(t) = 0$. $\square$

Remarque 4 Le calcul du délai de bout en bout en FP/FIFO avec FA est semblable au calcul avec la politique d'ordonnancement FIFO :

- L'algorithme 1 peut être étendu à la stratégie FP/FIFO. A la seule différence que la fonction de calcul de l'arriéré total $W_i^h(t)$ est calculé pour chaque flux $v_i \in \Gamma$.
- L'ensemble des points à tester est construit sur le même principe que l'algorithme 2 mais cet ensemble doit être calculé par niveau de priorité en raison du terme constant de $LinkRate_{i,x}^h$ (qui change par niveau de priorité).
- De plus, la condition d'arrêt du paragraphe 5.3.6 reste toujours valide.

Ainsi, l'ensemble des instants à tester est calculé par niveau de priorité P_i et est noté $TimeToTestP_i$. Le calcul défini par l'algorithme 3 est semblable à celui défini par l'algo-

rithme 2.

Input : Un nœud h , traversé par un ensemble de flux, dénoté I^h
 Pour chaque flux $v_i \in I^h$: C_i^h , T_i , sa priorité P_i et J_i^h

Result : $TimeToTestP_i$

begin

$TimeToTestP_i \leftarrow \emptyset$

foreach $x \in \{0, \dots, n\}$ **do**

foreach flux $v_i \in sp_{i,x}^h$ **do**

foreach $k \in \mathbb{N}$ **do**

$t \leftarrow -J_i + kT_i$

$TimeToTestP_i \leftarrow \text{Add}(t)$

if $x > 0$ **then**

$ser(t) \leftarrow \text{LinkRate}_{i,x}^{h-1} (rbf_{i,x}^h(t))$

if $ser(t) > t$ **then**

$TimeToTestP_i \leftarrow \text{Add}(ser(t))$

$TimeToTestP_i \leftarrow \text{Sort}(TimeToTestP_i)$

Algorithme 3 : Ensemble des instants t à tester en FP/FIFO.

6.4 Expérimentations en FP/FIFO

Dans ce paragraphe, nous appliquons FA sur la configuration décrite par la Fig. 6.6.

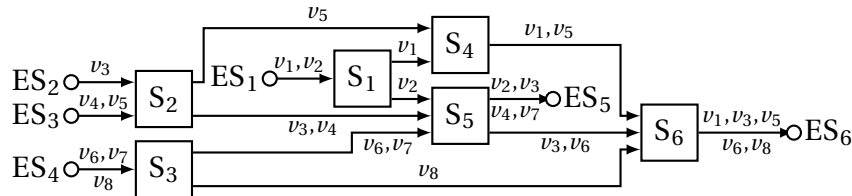


FIGURE 6.6 – Configuration AFDX utilisée pour tester la version FP/FIFO de FA.

Elle est composée de 6 End Systems, reliés par 6 commutateurs et qui s'échangent des données au travers de 8 VLs. Tous les liens ont le même débit (100 Mbps). Ainsi les durées de transmissions pour une trame sont égales sur tous les nœuds traversés. Les caractéristiques des flux sont résumées par le tableau 6.2. La latence technologique reste fixée à $L = 16 \mu s$.

Nous comparons les délais obtenus avec FA à ceux déterminées par la méthode des Trajectoires (TA) et le Network Calculus (NC). La version de NC est celle décrite dans [33]

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$Fmax_i$ (bits)	1000	1000	1000	2000	2000	1000	1000	2000
T_i (μs)	100	60	60	80	60	80	100	80
P_i	1	1	2	3	1	2	2	1

TABEAU 6.2 – Les caractéristiques des flux de la configuration dans la Fig. 6.6

tandis que pour TA, nous nous basons sur les travaux de [24] en utilisant la méthode corrigée FIFO définie dans [72].

Nous commençons par une analyse détaillée du délai du flux v_7 dans le paragraphe 6.4.1. Ensuite le paragraphe 6.4.2, nous comparons nos résultats avec ceux obtenus avec le Network Calculus (NC) et la méthode des Trajectoires (TA). Finalement dans le paragraphe 6.4.3, nous appliquons ces trois approches et les comparons sur une configuration industrielle.

6.4.1 Exemple de calcul en FP/FIFO

Nous étudions le flux v_7 pour illustrer le calcul du délai de bout en bout pire cas de la méthode FA en FP/FIFO qui suit le chemin suivant : $ES_4 - S_3 - S_5$.

Dans son nœud source ES_4 , v_7 est généré comme les flux v_6 et v_8 . Par définition, $Smin_i^{ES_4} = Smax_i^{ES_4} = 0 \mu s$ (pour $i \in \{6, 7, 8\}$), et donc leurs giques sont nulles, $J_i^{ES_4} = 0 \mu s$.

La priorité de v_7 étant de 2, nous obtenons les ensembles suivants : $hp_7^{ES_4} = \{v_8\}$, $sp_7^{ES_4} = \{v_6, v_7\}$ et $lp_7^{ES_4} = \emptyset$. Ainsi, comme il n'y a pas de sérialisation dans le nœud ES_4 (uniquement un nœud source), nous avons :

$$\begin{cases} WLP_7^{ES_4}(t) = 0 \\ WSP_7^{ES_4}(t) = rbf_6^{ES_4}(t) + rbf_7^{ES_4}(t) \\ WHP_{7,t}^{ES_4}(k) = rbf_8^h(W_{7,t}^h(k) - C_7^{ES_4}) \end{cases}$$

Ainsi, pour tout instant t , le calcul itératif est :

$$\begin{cases} W_{7,t}^{ES_4}(0) = C_7^{ES_4} \\ W_{7,t}^{ES_4}(k+1) = rbf_7^{ES_4}(t) + rbf_6^{ES_4}(t) + rbf_8^{ES_4}(W_{7,t}^{ES_4}(k) - C_7) \end{cases}$$

Le temps d'attente maximum (cf. formule (6.3)) est obtenu à l'instant $t = 0 \mu s$, $Bklg_7^{ES_4} = 40 \mu s$ et donc $Smax_7^{S_3} = 40 + 16 = 56 \mu s$.

Seuls les flux v_6 et v_7 traversent le nœud suivant, S_3 (avec $J_7^{S_3} = J_6^{S_3} = 56 - 26 = 30 \mu s$). Ils y arrivent en se partageant le même lien entrant. Ainsi, en sachant que $hp_7^{S_3} = lp_7^{S_3} = \emptyset$ and $sp_7^{S_3} = \{v_6, v_7\}$, nous obtenons :

$$\begin{cases} W_{7,t}^{S_3}(0) = C_7^{S_3} \\ W_{7,t}^{S_3}(k+1) = \min(rbf_6^{S_3}(t) + rbf_7^{S_3}(t), t + C_7^{S_3}) \end{cases}$$

Le délai pire cas encouru dans S_3 est déterminé à l'instant $t = 0 \mu s$, avec $Bklg_7^{S_3} = 10 \mu s$ et donc, $Smax_7^{S_5} = Smax_7^{S_3} + 10 + 16 = 82 \mu s$.

Dans le dernier nœud, v_2 arrive seul par un lien, v_3 et v_4 par un autre et v_7 par un troisième lien entrant avec les giques respectives de $J_2^{S_5} = 10 \mu s$, $J_3^{S_5} = 20 \mu s$, $J_4^{S_5} = 30 \mu s$ et $J_7^{S_5} = 40 \mu s$.

De plus, $hp_7^{S_5} = \{v_2\}$, $sp_7^{S_5} = \{v_3, v_7\}$ et $lp_7^{S_5} = \{v_4\}$, donc nous avons :

$$\left\{ \begin{array}{l} \text{WLP}_7^{S_5}(t) = C_4^{S_5} \\ \text{WSP}_7^{S_5}(t) = \min \left(\left\lfloor \frac{t - \alpha_2^{S_5}}{T_2} \right\rfloor C_2^{S_5}, t + C_2^{S_5} \right) - \left\lfloor \frac{t - \alpha_2^{S_5}}{T_2} \right\rfloor C_2^{S_5} \\ \quad + \min \left(rbf_3^{S_5}(t), t + C_3^{S_5} \right) \\ \quad + \min \left(rbf_7^{S_5}(t), t + C_7^{S_5} \right) \\ \text{WHP}_{7,t}^{S_5}(k) = rbf_2^{S_5} \left(W_{7,t}^{S_5}(k) - C_7^{S_5} \right) \end{array} \right.$$

Le temps d'attente maximum obtenu dans le nœud S_5 pour le flux v_7 est $Bklg_7^{S_5} = 50 \mu\text{s}$.

Finalement, le délai de bout en bout pire cas du flux v_7 est égal à $R_7 = 132 \mu\text{s}$. Les résultats des méthodes NC et TA sont présentés dans le tableau 6.3.

6.4.2 Cas d'étude en FP/FIFO

Nous appliquons FA sur la configuration AFDX représentée dans la Fig. 6.6 et dont les caractéristiques des flux sont résumées dans le tableau 6.2. Les délais obtenus avec FA sont comparés à ceux déterminés par la méthode NC et la méthode TA.

Les délais de bout en bout pire cas calculés avec ces trois approches, pour chaque chemin de chaque flux, sont regroupés dans le tableau 6.3. Cependant, il n'y a pas de délai déterminé par TA pour les flux v_5 et v_6 car la charge globale de ces flux est strictement supérieure à 1.

Pour affiner nos comparaisons (en particulier avec NC), nous avons différencié dans le tableau 6.3, les délais obtenus avec et sans sérialisation.

Ainsi, sans sérialisation, les délais obtenus par FA sont plus petits que ceux obtenus par NC et TA. Pour les flux à forte charge (*p.ex.* v_3 à destination de l'ES₆), les délais avec TA sont plus du double que ceux obtenus avec FA (et NC). La prise en compte de la sérialisation a peu d'influence sur les écarts entre les délais de TA et FA. Néanmoins, comme énoncé en conclusion du chapitre 5, il n'y a pas de relation de dominance entre FA et TA (aucune approche n'obtient des bornes plus fines que l'autre pour tous les flux de toutes configurations).

Flux	Non sérialisés			Sérialisés	
	NC	TA	FA	TA	FA
$\nu_1(\text{ES}_6)$	198	178	168	168	158
$\nu_2(\text{ES}_5)$	104	92	92	92	92
$\nu_3(\text{ES}_5)$	151	142	122	142	122
$\nu_3(\text{ES}_6)$	366	728	288	728	278
$\nu_4(\text{ES}_5)$	242	–	152	–	152
$\nu_5(\text{ES}_6)$	218	188	198	188	188
$\nu_6(\text{ES}_6)$	409	–	308	–	288
$\nu_7(\text{ES}_5)$	194	162	142	162	132
$\nu_8(\text{ES}_6)$	159	152	142	142	132

TABLEAU 6.3 – Les délais de bout en bout pire cas (en μs) de la configuration dans la Fig. 6.6.

À noter que la sérialisation a peu d’impact sur les délais déterminés par TA car elle est déjà partiellement (pour les flux partageant le même lien entrant que la trame analysée) prise en compte dans la version non sérialisée. De plus dans cette configuration, il y a peu de flux de même priorité partageant un même lien.

Finalement, en tenant compte de la sérialisation, les délais de bout en bout pire cas déterminés par FA sont améliorés, en moyenne de la durée de transmission d’une trame.

6.4.3 Configuration industrielle en FP/FIFO

Nous appliquons FA, TA et NC sur la même configuration industrielle AFDX que celle présentée dans le paragraphe 5.4.4. Cependant, cette configuration utilisée pour l’A380 est définie pour la politique d’ordonnancement FIFO. Ainsi, elle ne comporte pas de niveau de priorité. Nous l’avons modifié afin d’introduire, de manière aléatoire, 6 niveaux de priorité.

Le temps d’exécution de 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.

Nous comparons FA avec chacune des deux autres méthodes séparément. Nous utilisons la même étude statistique sur les délais que celle menée dans le paragraphe 5.4.4. Ces résultats sont résumés dans l’histogramme de fréquence de la Fig. 6.7.

Comme pour l’analyse de la configuration industrielle détaillée dans le paragraphe 5.4.4, l’histogramme est construit par intervalle de 10%. Chaque barre verticale correspond au nombre de flux dont la différence entre les délais déterminés par FA et NC (ou TA) est donnée par la valeur inscrite sur l’axe des abscisses. Une barre positive correspond à un délai plus petit déterminé par FA que pour l’une des deux autres approches et inversement pour une barre négative. Ainsi, la barre grise correspondant à 40% signifie que pour environ 100 flux, le délai obtenu par FA est de 30 à 40% plus petit que celui obtenu en appliquant TA.

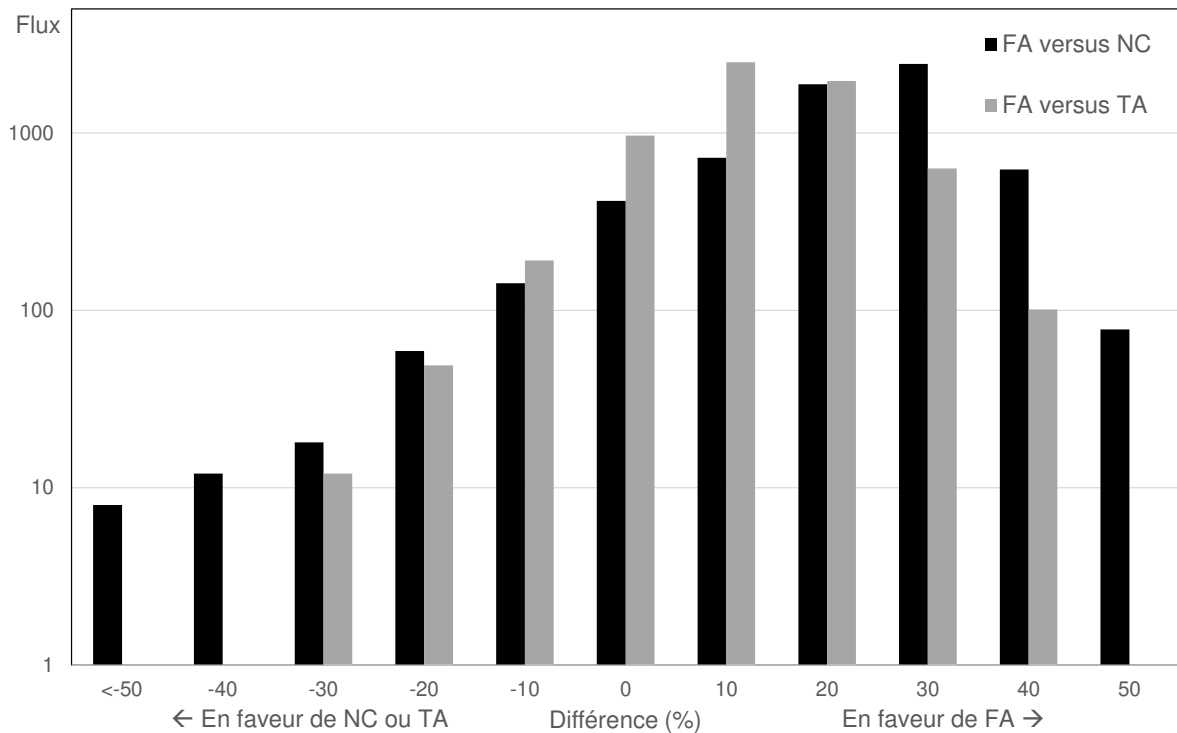


FIGURE 6.7 – Comparaisons statistiques entre les délais pire cas déterminés par FA et par chacune des deux autres approches (NC et TA) en FP/FIFO.

Comme en FIFO, en FP/FIFO, il n'y a pas de méthode dominante entre FA, TA et NC. Cependant, sur le 6412 flux, 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 plus petits de l'ordre de 17.55% par rapport à NC et de 8.01% par rapport à TA.

6.5 Conclusion

Dans les réseaux temps réel tels que l'AFDX, les commutateurs intègrent la possibilité de différencier le service en fonction des flux. Elle s'exprime par l'introduction d'une priorité statique affectée à chacun des flux. Les éléments réseaux servent, ensuite, les flux avec la politique d'ordonnancement FP/FIFO. Dans ce chapitre, la méthode FA initialement développée pour la politique d'ordonnancement FIFO a été étendue à la stratégie FP/FIFO. Cette extension inclut une démonstration formelle de nos résultats, et elle intègre la prise en compte de l'effet de sérialisation.

Les expérimentations menées sont composées d'un exemple détaillé du calcul du délai par FA, d'une comparaison des délais déterminés avec FA avec ceux obtenus par le Network Calculus et la méthode des Trajectoires sur une simple configuration et sur une configuration industrielle.

Ainsi, nous avons démontré que FA est applicable sur des configurations de taille industrielle même pour la politique d'ordonnancement FP/FIFO. De plus, les bornes déterminées par FA sont globalement meilleures que celles déterminées par les deux autres approches. Cependant, aucune des trois approches ne domine totalement les deux autres.

Pour de futurs travaux, certaines sources de pessimisme identifiées dans le calcul des délais déterminés par FA sont à corriger. Par exemple, pour de petites valeurs de t , dès lors que plusieurs trames de priorités différentes partagent un même lien entrant, l'arriéré arrivant par ce lien est pessimiste.

Ces travaux ont donné lieu à une publication [29].

Chapitre 7

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

Sommaire

7.1 Introduction	99
7.2 Modèle réseau en CBS	99
7.3 Le calcul du délai pire cas d'une trame en CBS	101
7.3.1 Interférence maximum générée par les classes moins prioritaires	101
7.3.2 Interférence pire cas générée par la classe X_i	102
7.3.3 Interférence maximum générée par les classes plus prioritaires	104
7.3.4 Le temps d'attente maximum dû au réapprovisionnement de crédit	105
7.3.5 Formules sur le délai calculé par FA en CBS	106
7.4 Expérimentations en CBS	107
7.4.1 Transposition de l'AVB au modèle réseau de FA	109
7.4.2 Exemple de calcul en CBS	110
7.4.3 Cas d'étude industriel en CBS	112
7.5 Conclusion	115

7.1 Introduction

Le réseau *Audio/Vidéo Bridging Switched Ethernet* (AVB) a été standardisé [6]. AVB est un réseau de paquets 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. Cet algorithme introduit un crédit pour chaque classe de priorité de flux et selon la valeur de ce crédit, une trame peut être transmise ou retardée.

Plusieurs approches basées sur la simulation [69, 99, 12] ont été développées pour l'AVB. Dans [74] et [98] des simulations modélisent l'IEEE 802.1 AVB sur l'*INET-Framework* pour *OMNeT++* [99]. Les simulations de l'AVB ont été appliquées au domaine de l'automobile. Par exemple, dans [12] les auteurs proposent une configuration réseau pour l'*Advanced Driver Assistance* (ADAS) et l'*infotainment* en utilisant l'AVB avec différentes topologies. Cependant, la simulation ne peut pas garantir une borne supérieure sur le délai car elle n'est pas exhaustive.

L'approche *Compositional Performance Analysis* (CPA) a été implémentée pour calculer les délais de traversée du réseau AVB [45]. Ce calcul itératif est propagé de nœud en nœud en se basant sur les périodes d'activités et des scénarios d'arrivée pire cas des flux. Bordoloi *et al.* [31] présentent un moyen pour prendre en compte de l'impact du CBS uniquement sur les flux de plus haute priorité. Cependant le calcul n'est présenté que sur un seul nœud et doit être couplé avec une approche holistique pour être appliqué sur un réseau.

Des approches développées pour le calcul de délais dans les réseaux Ethernet tels que l'AFDX ont été étendues à l'AVB en prenant en compte l'algorithme CBS. Par exemple, dans [41] les auteurs proposent une méthode d'analyse de l'AVB basée sur le Network Calculus (NC). La méthode des trajectoires a, elle aussi, été étendue au réseau AVB en prenant compte l'algorithme CBS [73]. Dans ce chapitre, nous étendons le modèle du nœud FP/FIFO défini dans le paragraphe 6.2 pour intégrer la modélisation de l'algorithme CBS dans le paragraphe 7.2. Ensuite, paragraphe 7.3, nous présentons le calcul d'une borne supérieure sur le délai de traversée du réseau AVB par la méthode FA. Ces travaux sont basés sur les résultats du chapitre 6 et modifiés pour prendre en compte CBS. Finalement, les résultats obtenus avec FA sont comparés à ceux déterminés par CPA dans le paragraphe 7.4.

7.2 Modèle réseau en CBS

Dans ce paragraphe, nous détaillons l'algorithme CBS puis, nous complétons le modèle réseau détaillé dans le paragraphe 6.2.

Nous considérons que le réseau est toujours composé d'un ensemble de flux sporadiques (mono-émetteur et mono-destinataire) Γ et d'un ensemble de nœuds S . L'ensemble des caractéristiques du réseau listées dans le paragraphe 6.2 est toujours valable en CBS. L'ensemble des flux du réseau est réparti en m (avec $m \geq 1$) classes de priorités fixes (une priorité unique par classe).

La politique de service à l'intérieur de chaque classe est basée sur la stratégie FIFO mais, en supplément, l'algorithme *credit based traffic shaper* (CBS) est utilisé dans chaque nœud. Ainsi, chaque classe a son propre crédit. Les trames d'une classe ne peuvent être

servies que si c'est leur file d'attente qui a la priorité dans le port de sortie (selon FP/FIFO) et si le crédit de la classe est suffisant.

Le crédit de chaque classe X_i (avec $1 \leq i \leq m$) dans un nœud donné varie dans le temps : il diminue à un taux constant appelé *send slope* et dénoté $\sigma_{X_i}^{h,-}$ quand les trames d'un flux de la classe X_i sont transmises par le nœud, et il est incrémenté à un taux constant appelé *idle slope* et dénoté $\sigma_{X_i}^{h,+}$ quand il y a des trames présentes dans la file d'attente de la classe X_i mais qu'aucune trame de cette classe n'est en cours de transmission. Le crédit est initialisé à zéro, et il ne peut être positif que si des trames de la classe X_i sont présentes dans sa file d'attente du nœud.

Une trame de la classe X_i (avec $1 \leq i \leq m$) peut être transmise si toutes les conditions suivantes sont remplies :

- le crédit de la classe X_i est supérieur ou égal à zéro ;
- il n'y a pas de trame d'une classe de plus haute priorité présente dans sa file d'attente avec un crédit positif ou nul ;
- pas d'autres trames ne sont en cours de transmission (ordonnancement non-pré-emptif).

Le crédit de la classe X_i ne peut pas rester positif si aucune trame de la classe X_i est en attente de transmission. Il est remis à zéro.

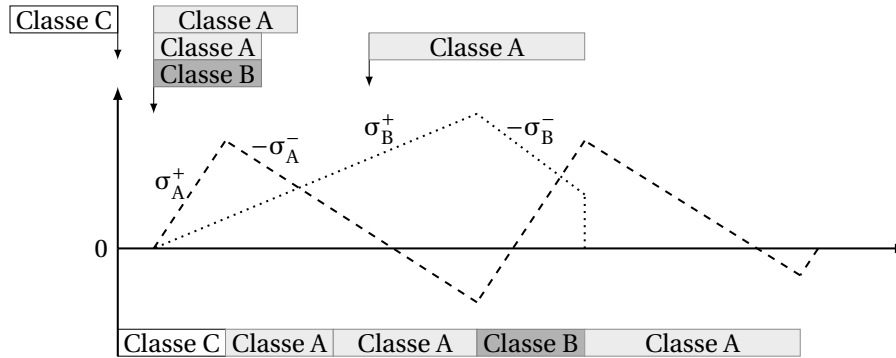


FIGURE 7.1 – Un exemple d'ordonnancement suivant le régulateur CBS.

Un exemple d'ordonnancement avec CBS est illustré dans la Fig. 7.1. Les trames de trois classes sont ordonnancées (la classe A est plus prioritaire que la classe B qui est elle-même plus prioritaire que la classe C). Une trame de classe C est transmise immédiatement dès l'instant 0 (car à cet instant, il n'y a qu'une seule trame). Peu après le début de cette transmission, trois trames arrivent : deux de classe A et d'une de la classe B. Les crédits de l'ensemble des classes SR sont incrémentés jusqu'à la fin de la transmission de la trame de classe C. Aussitôt une trame de la classe A est transmise, suivie par une autre trame de la classe A car le crédit est toujours positif et que la classe A est plus prioritaire. En parallèle, le crédit de la classe B continue d'augmenter. A la fin de la transmission de sa seconde trame, le crédit de classe A est devenu négatif. La troisième trame de classe A arrivée entre temps est bloquée dans l'attente que le crédit de sa classe soit au moins égal à zéro. Ainsi, la trame de classe B est donc transmise immédiatement. A la fin de cette transmission, le crédit de la classe A est positif et la dernière de ses trames est transmise.

L'*idle slope* d'une classe correspond à sa réservation de bande passante. Soit r^h le débit

du port de sortie du nœud h , alors la condition suivante est respectée :

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

Pour une classe X_i ($1 \leq i \leq m$), le *send slope* correspond, par conséquent, à la bande passante allouée aux autres classes. Ainsi, il peut être simplement déduit de son *idle slope* :

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

Dans ce chapitre, nous analysons le délai de bout en bout d'une trame f_i d'un flux v_i appartenant à une classe de flux X_i traversant un nœud CBS h . Nous dénotons par $HP_{X_i}^h$ (*resp.* $LP_{X_i}^h$) l'ensemble des classes strictement plus (*resp.* moins) prioritaires que la classe X_i . Nous définissons également $SHP_{X_i}^h = HP_{X_i}^h \cup X_i$. Soient Y une classe SR et h un nœud, alors nous dénotons par Y^h l'ensemble des flux traversant h et appartenant à la classe Y : $Y^h = \{v_j \in Y\} \cap I^h$. La relation entre $HP_{X_i}^h$, $SP_{X_i}^h$, $LP_{X_i}^h$ et $HP_{X_i}^h$ (définis par le graphe 6.2) et les ensembles de classes sont données par :

- $hp_i^h = \{v_j \in Y^h \mid Y \in HP_{X_i}^h\}$;
- $shp_i^h = \{v_j \in Y^h \mid Y \in HP_{X_i}^h \cup X_i\}$;
- $sp_i^h = X_i \cap I^h$;
- $lp_i^h = \{v_j \in Y^h \mid Y \in LP_{X_i}^h\}$

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

La définition du temps oisif défini définition 7 doit être étendue au contexte CBS pour prendre en compte le paramètre *idle slope*.

Définition 9 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 le crédit est revenu à 0.

En considérant cette définition, un nouveau terme $WNIL_i^h(t)$ doit être rajouté à la formule (6.1) du calcul de la plus longue période d'activité retardant la trame f_i , $W_i^h(t)$:

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

Paradoxalement, $WNIL_i^h$ est une partie de l'interférence quand aucune trame n'est transmise par le port de sortie du nœud, durant les phases de réapprovisionnement de crédit.

7.3.1 Interférence maximum générée par les classes moins prioritaires

L'interférence due aux classes de priorités inférieures, dénotée WLP_i^h dans le cas de CBS, est identique à celle définie pour la stratégie FP/FIFO dans le théorème 9 :

Théorème 10 L'interférence, $WLP_i^h(t)$, générée par les flux de priorités inférieures et retardant une trame f_i de la classe X_i en traversant un nœud CBS h durant un intervalle de temps de longueur t est égal à :

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

Démonstration : Plusieurs trames de priorités inférieures peuvent être transmises durant la période d'activité de niveau P_i à cause des temps de réapprovisionnement du crédit de X_i . Cependant, Bordoloi *et al.* [31] ont démontré qu'en utilisant l'algorithme CBS, l'impact de la non-préemption des flux moins prioritaires ne peut pas excéder la durée de transmission de la plus longue trame de plus petite priorité. En effet, la transmission des autres trames de priorité inférieure est nécessairement déjà comptabilisée dans le temps de réapprovisionnement du crédit de la classe X_i . $\square$

7.3.2 Interférence pire cas générée par la classe X_i

Nous considérons dorénavant l'interférence des flux de la classe X_i (flux de même priorité que v_i). L'algorithme CBS a un effet négatif (attente d'un crédit suffisant) sur les instants de transmission des trames dans un nœud CBS. Cependant, la limitation de la bande passante allouée, à l'entrée du nœud, peut aussi avoir un effet positif. Dans la formule (6.1), $WSP_{i,x}^h$ est limité seulement par le débit du lien de sortie. Nous introduisons un troisième terme dans l'expression du min, correspondant à l'application de l'algorithme CBS dans chaque nœud précédent h_{x-1} pour chaque lien entrant IP_x^h (avec $1 \leq x \leq n$) :

$$WSP_{i,x}^h(t) = \min \left[rbf_{i,x}^h(t), LinkRate_{i,x}^h(t), ShaperRate_{i,x}^h(t) \right] \quad (7.4)$$

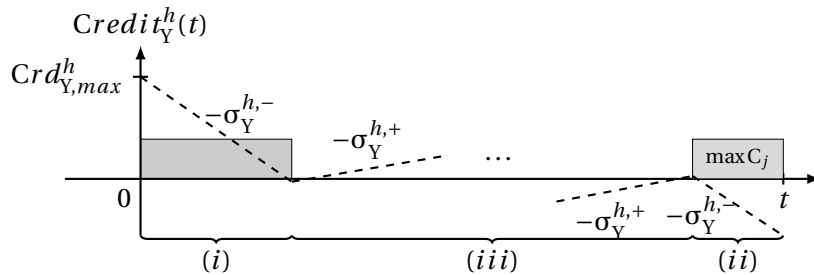


FIGURE 7.2 – Scénario pire cas défini et prouvé par Bordoloi *et al.* [31] maximisant le temps de transmission total généré par une classe Y sur un nœud CBS durant un intervalle de temps de durée t .

Bordoloi *et al.* [31] établit une borne supérieure au temps de transmission total des trames générées par la classe SR de plus haute priorité. Cette borne est basée sur la valeur du crédit maximum que peut accumuler cette classe. Nous étendons ce calcul à n'importe quelle classe SR.

Lemme 3 Pour une classe Y, le crédit maximum accumulé dans le nœud h est :

$$Crd_{Y,max}^h = \left(\max_{v_j \in LP_Y^h} C_j^h + \mathcal{BP}_Y^h \right) \sigma_Y^{h,+} \quad (7.5)$$

Où $\mathcal{BP}_Y^h$ désigne la plus longue période d'activité générée par l'ensemble des classes plus prioritaires HP_Y^h , obtenu par le plus petit point fixe de :

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

Remarque 5 Le point fixe existe si, et seulement si, $\sum_{v_j \in \text{HP}_Y^h} \frac{C_j^h}{T_j} \leq 1$ (assurée par construction).

Démonstration : Le crédit de la classe Y augmente si une trame f de cette classe est, en attente de transmission, dans la file d'attente du nœud h . La transmission peut être retardée par au plus :

- la plus longue trame générée par LP_Y^h , transmise juste avant l'arrivée de f ;
- suivie de la plus longue période d'activité $\mathcal{BP}_Y^h$, générée par les flux des classes plus prioritaires HP_Y^h .

Le calcul de la plus longue période d'activité a été établi pour les systèmes monoprocesseur [92, 95] dans le contexte des tâches non-préemptives sporadiques ou périodique. $\square$

Par conséquent, l'extension de l'interférence pire cas déterminée par Bordoloi pour toute classe SR Y est :

Lemme 4 Le temps de transmission total généré par une classe SR Y sur un nœud CBS durant un intervalle de durée t est borné par $\text{Bord}_Y^h(t)$:

$$\text{Bord}_Y^h(t) = \frac{\text{Crd}_{Y,max}^h}{\sigma_Y^{h,-}} + \max_{v_j \in Y^h} C_j^h + \left[\left(t - \frac{\text{Crd}_{Y,max}^h}{\sigma_Y^{h,-}} - \max_{v_j \in Y} C_j^h \right) \frac{\sigma_Y^{h,+}}{\sigma_Y^{h,+} + \sigma_Y^{h,-}} \right]^+$$

Où $[\text{Expr.}]^+ = \max(0, \text{Expr.})$

Démonstration :

Bordoloi *et al.* [31] ont dépeint un scénario pire cas (cf. Fig. 7.2) pour un flux d'une classe SR Y dans un nœud CBS h , basé sur la consommation du crédit. Ils ont prouvé que le temps de transmission total des trames générées par Y durant un intervalle de temps de durée t est maximisé quand :

- (i) à l'instant 0, le crédit de la classe Y est à son maximum et il est entièrement consommé par la transmission des trames de la classe Y (partie (i) de la Fig. 7.2) pour une durée de $\frac{\text{Crd}_{Y,max}^h}{\sigma_Y^{h,-}}$;
- (ii) une trame de taille maximale $\max_{v_j \in Y} C_j^h$ est transmise juste avant la fin de l'intervalle $[0, t]$ (élément (ii) de la Fig. 7.2) avec un crédit égal à ϵ (avec $\epsilon \rightarrow 0^+$) ;
- (iii) durant le temps restant, la classe Y génère l'interférence maximum contrainte par l'algorithme CBS ((iii) de la Fig. 7.2).

La valeur de $\text{Crd}_{Y,max}^h(t)$ est déterminée par le lemme 3, et la longueur de l'intervalle de temps restant est au plus égale à $\left(t - \frac{\text{Crd}_{Y,max}^h}{\sigma_Y^{h,-}} - \max_{v_j \in Y} C_j^h \right)$, mais ne peut pas être inférieure à 0. Durant cet intervalle, puisque le crédit commence et se termine par une valeur nulle, le temps de transmission total généré par Y est :

$$\left[\left(t - \frac{\text{Crd}_{Y,max}^h}{\sigma_Y^{h,-}} - \max_{v_j \in Y} C_j^h \right) \frac{\sigma_Y^{h,+}}{\sigma_Y^{h,+} + \sigma_Y^{h,-}} \right]^+$$

$\square$

Théorème 11 *L'interférence maximale, $WSP_i^h(t)$, générée par les flux de même priorité que la classe X_i et retardant une trame f_i de la classe X_i dans un nœud CBS h durant un intervalle de temps de durée t est :*

$$WSP_{i,x}^h(t) = \min \left[rbf_{i,x}^h(t), LinkRate_{i,x}^h(t), ShaperRate_{i,x}^h(t) \right] \quad (7.6)$$

$$\text{Avec : } ShaperRate_{i,x}^h(t) = Bord_Y^{h_x-1} \left(\frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h \right).$$

Démonstration : L'interférence déterminée dans le contexte FP/FIFO due aux fonctions RBF et au débit des liens est toujours valable (voir $WSP_{i,x}^h(t)$ définie dans le théorème 9). Le troisième terme prend en compte l'algorithme CBS en se basant sur les résultats du lemme 4, appliqué sur un intervalle de longueur $\frac{r^{h_x-1}}{r^h} t + \max_{v_j \in shp_{i,x}^h} C_j^h$. Il correspond la longueur de l'intervalle à analyser dans le nœud précédent $h_x - 1$ pour déterminer l'interférence arrivant sur le nœud h durant l'intervalle $[0, t]$ (cf. Fig. 5.7). $\square$

7.3.3 Interférence maximum générée par les classes plus prioritaires

Une trame qui arrive à l'instant t sur un nœud h peut être retardée par des trames plus prioritaires jusqu'à son instant de départ. Cet instant de départ est impacté par le CBS.

Lemme 5 *L'instant de départ au plus tard d'une trame f_i générée par un flux v_i d'une classe SRX_i arrivée à l'instant t dans un nœud h est $dep_i^h(t) = W_i^h(t) - \left(1 + \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} \right) C_i^h$.*

Démonstration : La définition de $W_i^h(t)$ inclut les temps d'attente insérés par l'algorithme CBS. Cependant, le temps de récupération dû à la transmission de la trame f_i elle-même n'a pas d'influence sur son instant de départ. Comme le crédit consommé par f_i dans le nœud h est égal à $\sigma_{X_i}^{h,-} \cdot C_i^h$, le temps de réapprovisionnement de crédit correspondant est alors $\frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}} C_i^h$. $\square$

Dans le contexte FP/FIFO, l'interférence maximum générée par les classes plus prioritaires est déterminée par les fonctions RBF (les contrats de trafic) et de l'instant de départ au plus tard (voir $WHP_i^h(t)$ défini dans le théorème 9). Dans le théorème 11, nous avons démontré comment l'interférence maximum des flux de même priorité que v_i liée au CBS peut être évaluée. Nous utilisons la même démarche pour borner l'interférence des flux de priorité supérieure :

Théorème 12 *Pour une trame f_i générée par un flux v_i appartenant à une classe X_i , l'interférence pire cas générée par les classes de priorités supérieures, dénotée $WHP_i^h(t)$, retardant f_i sur un nœud CBS h durant un intervalle de temps de durée t est maximisée par :*

$$\min \left[\sum_{v_j \in h p_i^h} rbf_j^h \left(dep_i^h(t) \right), \sum_{Y \in H P_{X_i}^h} Bord_Y^{h_x-1} \left(dep_i^h(t) + \max_{v_j \in SHP_Y^h} C_j^h \right) \right] \quad (7.7)$$

Démonstration : Le premier terme 7.7 correspond à l'interférence déduite des fonctions RBF en FP/FIFO (cf. théorème 9) en tenant compte de la nouvelle expression de l'instant de départ au plus tard de f_i défini dans le lemme 5. La seconde partie de 7.7 correspond au temps de transmission cumulé maximum, généré par les classes SR de priorités supérieures à X_i (HP_{X_i}), selon le lemme 4. Les deux termes étant des bornes garanties, $WHP_i^h(t)$ est donc le minimum entre les deux termes. Puisque la formule (7.7) dépend de $W_i^h(t)$ (via l'expression de $dep_i^h(t)$), il doit être déterminé par un calcul de point fixe comme en FP/FIFO. $\square$

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

L'algorithme *Credit Based Shaping* est une politique de service non-conservative : 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, si son crédit est négatif. Nous dénotons ce délai par $WNIL_i^h(t)$, selon la formule (7.3).

Théorème 13 *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 :*

$$WNIL_i^h(t) = \left(ready_i^h(t) - \sum_{v_j \in hp_i^h} \left\lfloor \frac{ready_j^h(t)}{T_j} \right\rfloor C_j^h \right)^+$$

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

Démonstration : Pour une trame f_i (avec un temps de transmission C_i^h) générée par un flux v_i d'une classe SR X_i , le temps de réapprovisionnement du crédit consommé pour la transmission de f_i par le nœud h est égal à $C_i^h \cdot \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}}$ (cf. la démonstration du lemme 5). La trame f_i est donc retardée au plus tard jusqu'au réapprovisionnement total correspondant à la transmission de toutes les trames de mêmes priorités que f_i et arrivées sur h avant t : $(WSP_i^h(t) - C_i^h) \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}}$.

Ce temps d'attente perdu pour la classe X_i , ne l'est pas forcément pour les autres classes SR ayant un crédit positif ou nul. Ainsi, une partie de ce qui est comptabilisé dans le terme $WHP_i^h(t)$ peut être transmise pendant que la classe X_i est en attente d'un crédit suffisant. Ce temps compté deux fois, ne peut l'être que sur l'intervalle $[0, ready_i^h(t)]$, soit avant que le crédit soit suffisant pour transmettre toutes les trames de X_i^h jusqu'à f_i incluse. Si aucune trame de X_i^h n'est transmise, le crédit augmente et il devient suffisant après un délai de $ready_i^h(t) = (WSP_i^h(t) - C_i^h) \frac{\sigma_{X_i}^{h,-}}{\sigma_{X_i}^{h,+}}$. En effet, après $ready_i^h(t)$, le crédit nouvellement accumulé par la classe X_i ne sera pas utilisé avant $dep_i^h(t)$.

Comme le scénario pire cas d'un flux sporadique est de le considérer strictement périodique, les flux de plus hautes priorités (que v_i) accumulent durant l'intervalle $[0, t]$ un temps de transmission total égal à $\sum_{v_j \in hp_i^h} \left\lfloor \frac{t}{T_j} \right\rfloor C_j^h$.

$WNIL_i^h(t)$ ne pouvant pas être négatif, ceci conclut notre démonstration. $\square$

7.3.5 Formules sur le délai calculé par FA en CBS

Le délai pire cas subi par une trame en traversant un nœud est obtenu lorsque la différence entre son instant d'arrivée dans le nœud sous étude et son instant de départ est maximale :

Théorème 14 *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 (7.8) si la condition sur la charge $\sum_{v_j \in I^h} \frac{C_j}{T_j} < 1$ est respectée.*

$$\begin{cases} W_{i,t}^h(0) = C_i^h \\ W_{i,t}^h(k+1) = WLP_i^h + WSP_i^h(t) + WHP_{i,t}^h(k) + WNIL_i^h(t) \end{cases} \quad (7.8)$$

Où

$$\begin{aligned} WLP_i^h(t) &= \max_{v_j \in lp_i^h} C_j^h \\ WSP_i^h(t) &= \sum_{v_j \in I_{0,i}^h} rbf_j^h(t) + \sum_{x=1}^n WSP_{i,x}^h(t) \\ WHP_{i,t}^h(k) &= \min \left[\sum_{v_j \in hp_i^h} rbf_j^h(dep_i^h(t)), \sum_{Y \in HP_{X_i}^h} Bord_Y^{h_x-1} \left(dep_i^h(t) + \max_{v_j \in SHP_Y^h} C_j^h \right) \right] \\ WNIL_i^h(t) &= \left(ready_i^h(t) - \sum_{v_j \in hp_i^h} \left\lfloor \frac{ready_i^h(t)}{T_j} \right\rfloor C_j^h \right)^+ \end{aligned}$$

avec

$$WSP_{i,x}^h(t) = \min \left[rbf_{i,x}^h(t), LinkRate_{i,x}^h(t), ShaperRate_{i,x}^h(t) \right]$$

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

Démonstration : Le résultat est une application directe des Théorèmes 10, 11, 12 et 13. $\square$

Remarque 6 *Le calcul du délai de bout en bout en CBS avec FA est semblable à celui détaillé pour la politique d'ordonnancement FP/FIFO dans le chapitre 6. Cependant, l'ensemble des points à tester défini dans l'algorithme 3 doit être mis à jour afin de prendre en compte la fonction $ShaperRate_{i,x}^h$.*

De plus, la condition d'arrêt du paragraphe 5.3.6 reste toujours valide, si nous prenons en compte la définition de l'instant oisif en CBS défini dans la définition 9.

Ainsi, l'ensemble des points à tester est calculé pour chaque classe X_i . Par rapport à l'algorithme 3, dans l'algorithme 4, nous rajoutons les points $shr(t)$ qui correspondent à l'intersection de la nouvelle marche d'escalier générée par l'arrivée d'une trame de X_i et de la fonction $ShaperRate_{i,x}^h$ (si ce point se situe après t).

Input : Un nœud h , traversé par un ensemble de flux, dénoté Γ^h

Pour chaque flux $v_i \in \Gamma: C_i^h, T_i$, sa classe X_i et J_i^h

Result : $TimeToTestX_i$

begin

$TimeToTestX_i \leftarrow \emptyset$

foreach $x \in \{0, \dots, n\}$ **do**

foreach flux $v_i \in X_i^h \cap \Gamma_x^h$ **do**

foreach $k \in \mathbb{N}$ **do**

$t \leftarrow -J_i + kT_i$

$TimeToTestX_i \leftarrow \text{Add}(t)$

if $x > 0$ **then**

$ser(t) \leftarrow \text{LinkRate}_{i,x}^h{}^{-1}(rbf_{i,x}^h(t))$

$shr(t) \leftarrow \text{ShaperRate}_{i,x}^h{}^{-1}(rbf_{i,x}^h(t))$

if $ser(t) > t$ **then**

$TimeToTestX_i \leftarrow \text{Add}(ser(t))$

if $shr(t) > t$ **then**

$TimeToTestX_i \leftarrow \text{Add}(shr(t))$

$TimeToTestX_i \leftarrow \text{Sort}(TimeToTestX_i)$

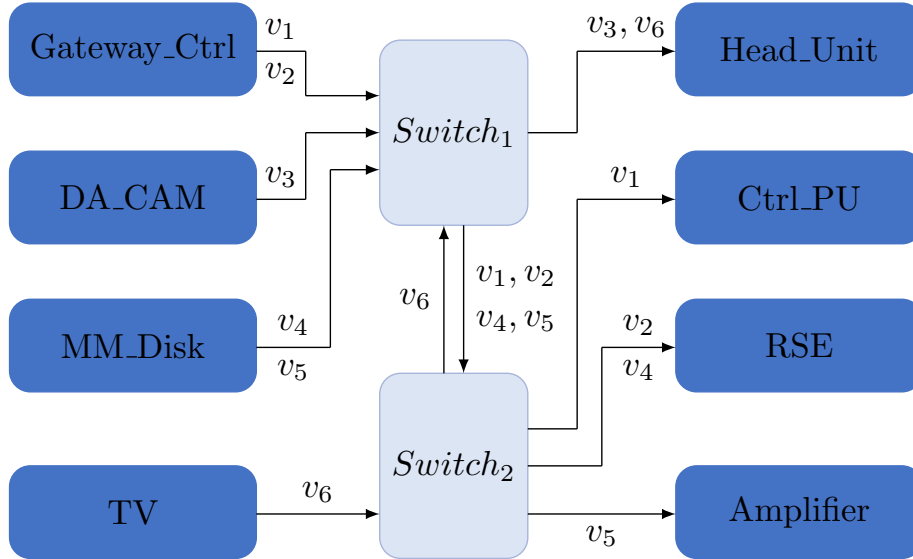
Algorithme 4 : Ensemble des instants t à tester en CBS.

7.4 Expérimentations en CBS

Dans cette section, nous appliquons la méthode FA à la configuration AVB développée pour le système *Advanced Driver Assistance System* (ADAS) et l'*Infotainment* [12, 99]. Nous n'avons pas encore implémenté les méthodes Network Calculus (NC) et Trajectoires (TA). Une telle comparaison sera dans les perspectives de la thèse. Ainsi, nous comparerons nos résultats avec ceux obtenus avec l'approche *Compositional Performance Analysis* (CPA) [45].

La configuration est illustrée dans la Fig. 7.3. Elle est composée de 8 ECUs inter-connectés par deux commutateurs AVB $Switch_1$ ans $Switch_2$. Tous les ports de sortie ont un débit constant de 100Mbps. De plus, la latence technologique est fixée à $L = 8 \mu s$. Les caractéristiques des flux sont résumées dans le tableau 7.1. Les flux liés au système ADAS lui-même sont critiques et ont donc la plus haute priorité (classe A) tandis que les flux de l'*Infotainment* sont moins prioritaires (classe B).

Dans la suite, nous présentons dans le paragraphe 7.4.1 la transposition de l'AVB à notre modèle. Ensuite, nous présentons le calcul détaillé de notre approche sur un flux. Pour finir, paragraphe 7.4.3, nous appliquons FA sur un cas d'étude industriel développé pour le système ADAS de l'automobile. Nous comparons nos résultats à ceux obtenus avec CPA.

FIGURE 7.3 – Configuration d'un réseau AVB pour les applications ADAS et *Infotainment*.

ν_i	$T_i(\mu s)$	$C_i^h(\mu s)$	Classe AVB	$R_i(\mu s)$	Destination
ν_1	5000	5, 12	Class A	2762,72	Ctrl_CPU
ν_2	5000	5, 12	Class A	2994	RSE
ν_3	125	32, 64	Class A	188,96	Head_Unit
ν_4	280	115, 68	Class B	631,9	RSE
ν_5	1400	115, 68	Class B	618,67	Amplifier
ν_6	560	115, 68	Class B	395,68	Head_Unit

TABLEAU 7.1 – Les caractéristiques des flux et les délais pire cas calculés par FA pour la configuration dans la Fig. 7.3.

7.4.1 Transposition de l'AVB au modèle réseau de FA

L'ensemble des flux AVB est réparti en plusieurs classes de priorités fixes. La politique d'ordonnancement FIFO régit chaque classe. Il y a deux types de classes de trafic dans l'AVB : 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 au moins deux classe SR : A et B (la classe A ayant une priorité plus haute que 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.

Dans les systèmes de l'automobile (cf. paragraphe 2.5) les ES correspondent aux *Electronic Control Unit* (ECU). Les nœuds du réseau modélisent ainsi toujours les ports de sortie des ES et des commutateurs AVB.

Selon le protocole IEEE 802.1Qat, à l'entrée du réseau, les flux sont définis par trois paramètres : le *Class Measurement Interval* (CMI), le *Maximum Interval Frames* (MIF) et le *Maximum Frame Size* (MFS). Selon ce modèle, un flux génère au plus MIF trames durant un intervalle de temps de longueur CMI. Par exemple, le standard CMI pour la classe SR A (resp. classe B) est de 125 μ s (resp. 250 μ s). Ainsi, la bande passante allouée aux flux AVB ne peut pas excéder $\frac{MIF \times MFS}{CMI}$.

Le contrat de trafic défini par les paramètres CMI, MIF et MFS à son entrée dans le réseau est le point de départ de l'analyse du délai pire cas. A partir de ces valeurs, nous pouvons redéfinir les paramètres de chaque flux $v_i \in I^h$ d'un nœud donné h du 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 ;
- T_i , la période est égale au CMI quand MIF = 1 ;
- P_i , tous les flux d'une classe SR X ont la même priorité.

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 mais peuvent avoir, par définition un ou plusieurs destinataires. Par conséquent, comme pour les VLs, un délai est calculé par flux et dès que le flux se subdivise, le délai est propagé vers chaque destinataire.

Chaque flux v_i appartient à une classe X_i . Chaque classe X_i est ainsi définie par son niveau de priorité P_i et des deux paramètres CBS : l'*idle slope* $\sigma_{X_i}^+$ et le *send slope* $\sigma_{X_i}^-$.

Dans une configuration AVB typique, 75% de la bande passante totale est réservée aux classes SR (c-à-d. : $\sigma_{X_1}^+ + \dots + \sigma_{X_m}^+ \leq 0.75 \times r$). Ces paramètres (*idle slope* et *send slope*) peuvent être configurés manuellement (sous réserve que la contrainte (7.1) soit respectée) ou automatiquement via le *Stream Reservation Protocol* IEEE 802.1Qat.

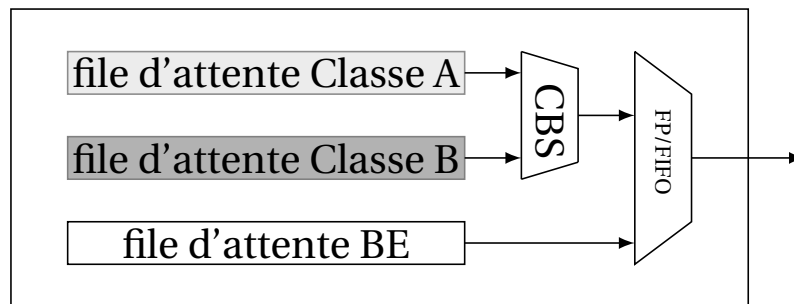


FIGURE 7.4 – Port de sortie AVB

L'algorithme *credit based traffic shaper* (CBS) est appliqué uniquement aux classes SR dans chaque nœud (port de sortie d'un commutateur ou d'un *End System*) du réseau (cf. Fig. 7.4). Les flux de la classe BE ont tous la plus petite priorité. La classe BE n'a pas de paramètre CBS. Cependant, pour simplifier la notation, nous la modélisons comme une classe SR avec la plus petite priorité et une bande passante maximum (*idle slope*) égale au débit du lien de sortie du nœud r^h et un *send slope* égal à zéro. Comme il n'y a pas de valeur pour les paramètres MFS et CMI pour les flux BE, nous pouvons supposer, dans le pire cas, une taille de trame maximum égale à la trame Ethernet maximum (1518 octets). De plus, sans période, nous ne pouvons pas calculer de délai pour le trafic BE. Cependant, l'impact d'une classe BE sur les classes de plus haute priorité est limité, à cause de la non-préemption) à une unique trame. Par conséquent, pour l'analyse des classes SR, nous supposons que $T_i = +\infty$ pour l'ensemble des flux BE v_i .

Ainsi, la configuration ADAS dépeinte par la Fig. 7.3 est modélisée selon la Fig. 7.5. Dans chaque nœud, la bande passante de chaque classe SR est calculée. Par exemple, quatre flux traversent le nœud S_{11} , la bande passante réservée par les classes A et B sur ce nœud est respectivement : $\sigma_A^{S_{11},+} = \frac{C_1 * R}{T_1} + \frac{C_2 * R}{T_2} = 0.2048$ et $\sigma_B^{S_{11},+} = \frac{C_4 * R}{T_4} + \frac{C_5 * R}{T_5} = 49.58$.

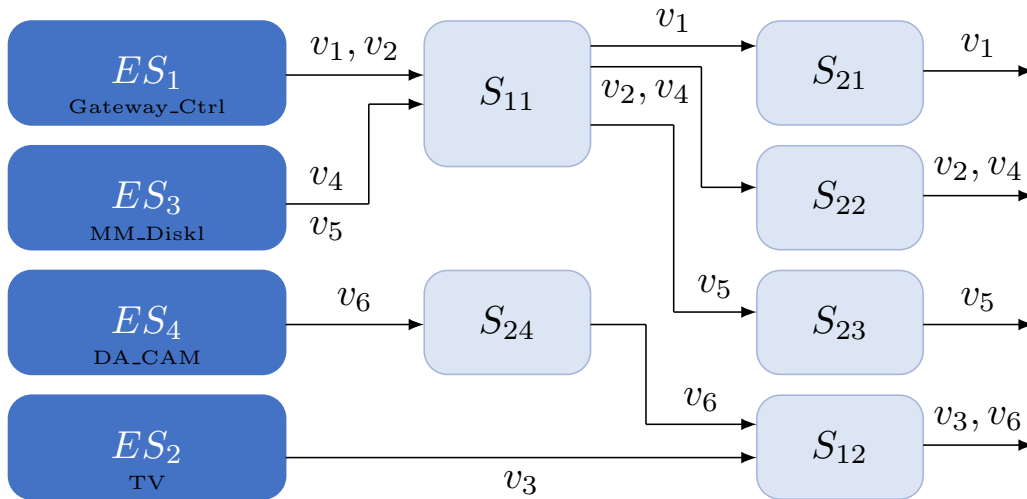


FIGURE 7.5 – Modélisation de la configuration représentée dans la Fig. 7.3.

7.4.2 Exemple de calcul en CBS

Nous analysons avec FA, le délai de traversée du réseau par une trame f_4 du flux v_4 de classe B. Le premier nœud à analyser est le nœud source de v_4 , *MM_Disk* (ES_3). Deux flux v_4 et v_5 sont générés par l' ES_3 . Nous avons alors : $\sigma_B^{ES_3,+} = 49.58$ et $J_4^{ES_3} = J_5^{ES_3} = 0 \mu s$ (pas de gigue initiale). Il n'y pas d'effet de sérialisation sur un nœud source et $lp_5^{ES_3} = hp_5^{ES_3} = \emptyset$. Ainsi, pour tout $t \geq 0$, l'interférence maximale subit par f_4 sur ES_3 est :

$$\begin{aligned} W_{4,t}^{ES_3} &= WSP_4^{ES_3}(t) + WNIL_4^{ES_3}(t) \\ &= \left(rbf_4^{ES_3}(t) + rbf_5^{ES_3}(t) \right) + (WSP_4^{ES_3} - C_4) \left(\frac{\sigma_B^-(ES_3)}{\sigma_B^+(ES_3)} \right) \end{aligned}$$

L'arriéré temporel maximal est obtenu en $t = 0 \mu s$: $Bklg_4^{ES_3} + C_4 = W_4^{ES_3}(0) - 0 = 349 \mu s$. Ainsi, $Smax_4^{S_{11}} = 349 + L = 357 \mu s$.

Les flux v_4 and v_5 partagent le même lien $IP_2^{S_{11}}$ jusqu'au prochain nœud de $\mathcal{P}_4$, S_{11} . Deux autres flux v_1 et v_2 arrivent pas un même autre lien $IP_1^{S_{11}}$ et sont de la classe de plus haute priorité A. Ainsi, sur ce nœud, nous devons appliquer l'effet de sérialisation et l'algorithme CBS. Nous avons $slp_4^{S_{11}} = \emptyset$ et donc, l'interférence maximale subit par f_4 se formalise ainsi :

$$W_4^{S_{11}}(t) = WSP_4^{S_{11}}(t) + WHP_4^{S_{11}}(t) + WNIL_4^{S_{11}}(t)$$

Où

$$\begin{aligned} WSP_4^{S_{11}}(t) &= \min [rbf_4^{IP_2^{S_{11}}}(t) + rbf_5^{IP_2^{S_{11}}}(t), LinkRate_{4,2}^{S_{11}}(t), \\ &\quad ShaperRate_{4,2}^{S_{11}}(t)] \\ WHP_4^{S_{11}}(t) &= \min [rbf_1^{IP_1^{S_{11}}}(W_4^{S_{11}}(t) - C_4) + rbf_2^{IP_1^{S_{11}}}(W_4^{S_{11}}(t) - C_4), \\ &\quad Bord_A^{ES_{11}}(W_4^{S_{11}}(t) - C_4 + \max_{v_j \in \{v_4, v_5, v_1, v_2\}} C_j)] \\ WNIL_4^{S_{11}}(t) &= \left(ready_4^{S_{11}}(t) - \sum_{v_j \in hp_4^{S_{11}}} \left\lfloor \frac{ready_4^{S_{11}}(t)}{T_j} \right\rfloor C_j \right)^+ \\ ready_4^{S_{11}}(t) &= \left(WSP_4^{S_{11}}(t) - C_4 \right) \frac{\sigma_B^{S_{11},-}}{\sigma_B^{S_{11},+}} \end{aligned}$$

avec : $J_1^{S_{11}} = J_2^{S_{11}} = 2500 \mu s$.

L'arriéré temporel pire cas rencontré par f_4 sur S_{11} est borné par $Bklg_4^{S_{11}} = 138 \mu s$. Cette borne est déterminée en $t = 3138.77 \mu s$. Par conséquent, sur le dernier nœud de $\mathcal{P}_4$, S_{22} , nous obtenons pour f_4 : $Smax_4^{S_{22}} = 349 + L + 138 + L = 503 \mu s$.

Dans S_{22} , le seul flux qui interfère avec v_4 est le flux de plus haute priorité v_2 . Ils arrivent sur S_{22} via le même lien entrant $IP_1^{S_{22}}$. Alors $W_4^{S_{22}}(t)$ est défini par :

$$W_{4,t}^{S_{22}}(k+1) = WSP_4^{S_{22}}(t) + WHP_4^{S_{22}}(t) + WNIL_4^{S_{22}}(t)$$

Où

$$\begin{aligned} WSP_4^{S_{22}}(t) &= \min [rbf_4^{IP_1^{S_{11}}}(t), LinkRate_{4,2}^{S_{11}}(t), ShaperRate_{4,1}^{S_{22}}(t)] \\ WHP_4^{S_{22}}(t) &= \min [rbf_2^{IP_1^{S_{22}}}(W_4^{S_{22}}(t) - C_4), \\ &\quad Bord_A^{S_{11}}(W_4^{S_{22}}(t) - C_4 + \max_{v_j \in \{v_4, v_2\}} C_j)] \\ WNIL_4^{S_{22}}(t) &= \left(ready_4^{S_{22}}(t) - \sum_{v_j \in hp_4^{S_{22}}} \left\lfloor \frac{ready_4^{S_{22}}(t)}{T_j} \right\rfloor C_j \right)^+ \\ ready_4^{S_{22}}(t) &= \left(WSP_4^{S_{22}}(t) - C_4 \right) \frac{\sigma_B^{S_{22},-}}{\sigma_B^{S_{22},+}} \end{aligned}$$

Avec $J_2^{S_{22}} = 2731.36 \mu s$.

Ainsi, l'arriéré temporel pire cas retardant f_4 sur S_{22} est obtenu à l'instant $t = 2655.3 \mu s$ et est égal à $128.9 \mu s$.

Finalement, nous obtenons un délai pire cas pour la trame f_4 de $R_4 = 631.9 \mu s$.

7.4.3 Cas d'étude industriel en CBS

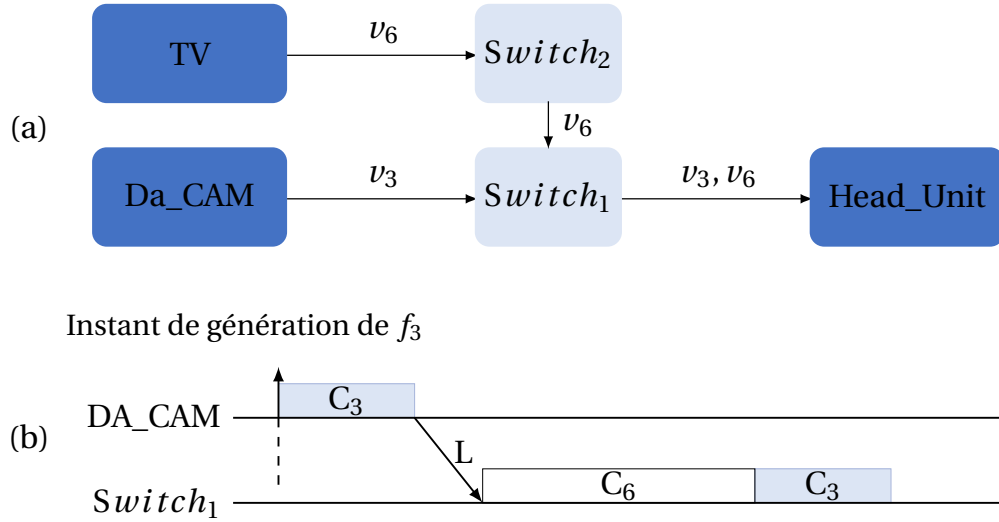
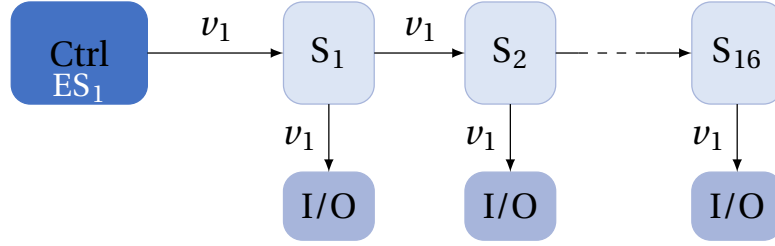


FIGURE 7.6 – Scénario pire cas pour le flux v_3 .

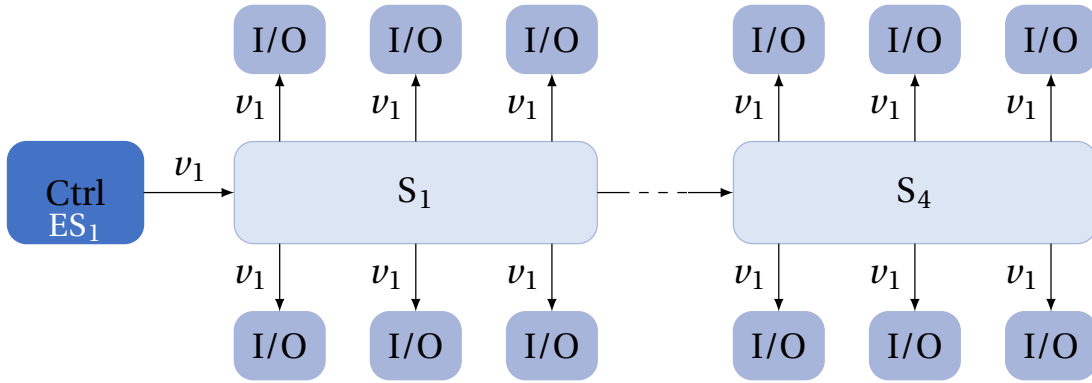
Nous analysons la configuration ADAS illustrée par la Fig. 7.3. Dans [73], les auteurs ont appliqué la méthode des Trajectoires (TA) pour calculer les délais de traversée de la configuration ADAS pour chacun des flux. Cependant, nous ne pouvons pas encore comparer nos résultats à ceux de TA présentés dans [73]. En effet, prenons l'exemple du flux v_3 de la configuration ADAS (cf. Fig. 7.3). Le délai pire cas obtenu par TA est égal à $194.7 \mu s$ [73]. Pourtant, nous obtenons $188.69 \mu s$ si nous essayons d'appliquer TA nous-même. De plus, le pire scénario subit par v_3 est simple comme illustré par la Fig. 7.6.b : v_3 est seul dans son nœud source et l'unique interférence se produit dans le Switch₁, avec le flux de plus faible priorité v_6 (qui arrive par un lien différent). Ainsi, une unique trame de v_6 (qui est de priorité plus faible) peut retarder v_3 et le délai pire cas de traversée du réseau de v_3 est : $2 \times 32.64 + 115.68 + 8 = 188.96 \mu s$. Comme il n'existe actuellement aucun outil de calcul accessible au public, nous préférons ne pas comparer les chiffres que nous sommes incapables de discuter.

Dans [73], les délais pire cas obtenus avec TA sont comparés à des résultats de simulation afin d'estimer le pessimisme de TA, en utilisant le modèle *CoRE4Inet* [80, 98]. *CoRE4Inet* est une extension du modèle *INET* développé pour la plateforme de simulation *OMNet++*. Cependant, l'outil ne gère pas correctement les périodes des flux SR. Ainsi, les *idleSlopes* sont toujours calculés en utilisant les périodes standards AVB : $125 \mu s$ pour la classe A et $250 \mu s$ pour la classe B. A notre connaissance, la version courante de *CoRE4Inet* n'est pas adaptée pour une configuration comme celle de l'ADAS avec des périodes différentes pour chaque flux d'une même classe. Ainsi, le tableau 7.1 regroupe les délais déterminés à l'aide la méthode FA mais nous ne pouvons comparer nos résultats à une autre approche.

Cependant, afin de comparer FA et CPA, nous appliquons FA sur les mêmes configurations que celles sur lesquelles ont été appliquées CPA dans [45]. Elles sont composées de classes SR et de trafic BE.



a. Topologie linéaire



b. Topologie en *Cluster*

FIGURE 7.7 – Configurations AVB.

Ainsi, nous avons appliqué FA sur les configurations AVB utilisées dans [73, 45] et illustrées dans la Fig. 7.7. Ces deux configurations ont des topologies différentes mais elles partagent les mêmes caractéristiques de flux. Dans les deux cas, le réseau est composé d'un unique contrôleur envoyant des messages périodiques de classe A à un ensemble de nœuds I/O avec une gigue initiale égale à $250\mu\text{s}$ (égale à la période). Les trames émises ont une taille maximale de 336 bits. De plus pour chaque nœud h , le débit du port de sortie est égal à $r^h = 1\text{ Gbps}$ et la latence technologique est fixée à $L = 0.33\mu\text{s}$. En plus du flux de classe A, du trafic *Best-effort* (BE) est ajouté dans chaque commutateur et est égal à une taille de trame Ethernet maximale 12144 bits.

La configuration linéaire dans la Fig. 7.7.a est composée d'un unique contrôleur et de 16 nœuds I/O. Le flux de classe A v_1 a plusieurs destinataires : il est émis par le nœud ES_1 et a 16 nœuds I/O destinataires : 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. 7.7.b), v_1 a 24 destinataires regroupés en étoile sur quatre commutateurs (six nœuds I/O par commutateur).

Dans une topologie linéaire, le délai de traversée de bout en bout pire cas devrait subir une augmentation importante à chaque commutateur traversé, comme nous pouvons le

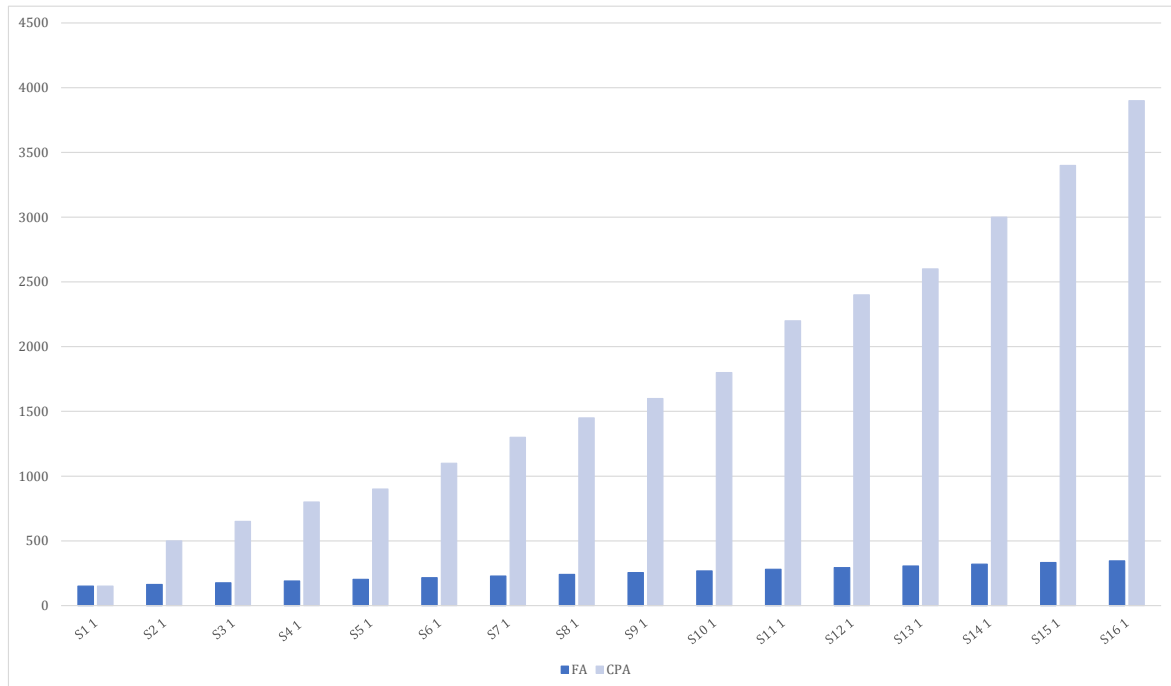


FIGURE 7.8 – Comparaison des résultats obtenus par FA et CPA pour la configuration représentée par la Fig. 7.7.a.

remarquer dans les bornes déterminées avec CPA regroupées dans la Fig. 7.8. Cependant, l'interférence arrivant sur un nœud a été régulée (effet de sérialisation et algorithme CBS) par le nœud précédent. Ainsi, nous obtenons des délais en utilisant la méthode FA plus faibles que ceux déterminés par CPA. Par conséquent, même avec une bande passante allouée plus restreinte, la méthode FA est capable de calculer une borne supérieure sur chaque délai, en tenant compte du régulateur tout au long du chemin du flux. A chaque commutateur, le délai subit l'interférence d'une trame de plus faible priorité 12144 bits avec un débit de sortie des nœuds de 1 Gbps (*c-à-d.* 12.144 μ s).

Pour la topologie en étoile, les nœuds I/O sont interconnectés en ligne à travers quatre commutateurs (six nœuds I/O par commutateur). Par conséquent, les délais déterminés pour chaque nœud connecté à un même commutateur sont égaux et nous regroupons ces délais par commutateur (*cf.* Fig. 7.9). Il y a des différences importantes entre CPA et FA. En effet, même si pour le premier commutateur les délais sont égaux, à partir du second commutateur, les délais diffèrent rapidement en faveur de FA. La principale raison vient de la prise en compte de la régulation via l'algorithme CBS dans les nœuds précédents.

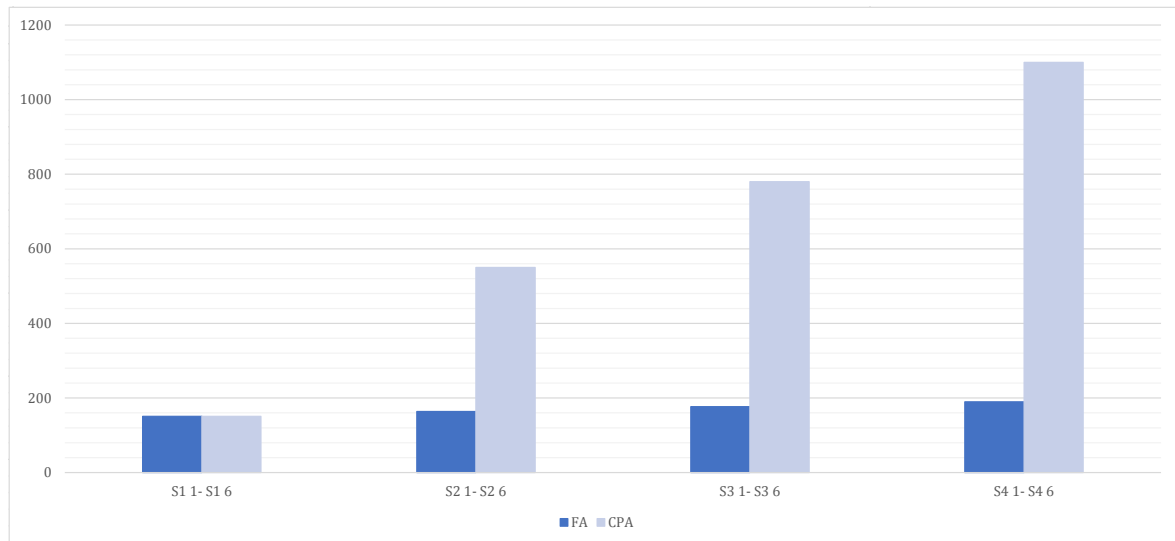


FIGURE 7.9 – Comparaison des résultats obtenus par FA et CPA pour la configuration représentée par la Fig. 7.7.b.

7.5 Conclusion

Le réseau AVB est un standard basé sur Ethernet pour les applications automobiles grâce à sa bande passante élevée et sa garantie de qualité de service. Il est destiné à être utilisé dans les systèmes temps réel distribués critiques. Une borne supérieure du délai de bout en bout de traversée de l'AVB pour chaque flux est à déterminer afin de garantir le déterminisme du réseau.

Dans ce chapitre, nous avons présenté l'extension de la méthode FA à l'AVB et l'algorithme CBS pour calculer le pire délai de bout en bout de chaque flux d'un réseau AVB. Cette approche est basée sur celle développée pour FP/FIFO dans le chapitre 6. Dans le calcul selon FA, le délai de bout en bout d'un flux est composé de la somme de la latence technologique entre les nœuds sur son chemin et de la somme des délais variables subis dans chaque nœud. Ce délai variable dépend de l'interférence de tous les flux traversant le nœud.

Avec FP/FIFO comme politique de service, l'impact des flux dépend uniquement de son niveau de priorité. Dans un réseau AVB, en plus de FP/FIFO, l'algorithme CBS basé sur une consommation de crédit régule les classes de plus hautes priorités afin de limiter leur influence. Nous avons proposé une méthode pour prendre en compte cet algorithme supplémentaire et ainsi comptabilisé le temps additionnel induit pour la classe sous analyse quand elle ne possède pas suffisamment de crédit pour transmettre une trame en attente. Nous avons appliqué FA sur une configuration industrielle développée pour le système *Advanced Driver Assistance System* (ADAS) et l'*Infotainment* [12, 99]. De plus, nous avons comparés les délais déterminés avec FA à ceux calculés à l'aide l'approche *Compositional Performance Analysis* (CPA) [45] sur deux petites configurations avec respectivement une topologie linéaire et en étoile. FA détermine des bornes plus serrées que CPA.

AVB est en constant développement vers un nouveau réseau standardisé pour les systèmes temps réel critiques, le standard *Time Sensitive Networking*. En TSN, un troisième type de trafic (en plus des classes *Stream Reservation* et du trafic *Best Effort*) critique avec

un algorithme de régulation *Time Aware Shaper* (TAS) est ajouté. Dans [20]), les auteurs ont proposé une analyse des délais dans un réseau TSN pour lequel l'ordonnancement serait hors-ligne pour les flux critiques (avec la plus haute priorité) et les autres flux sont ordonnancés comme des flux AVB classiques. La principale différence entre TAS et CBS, est que pour les flux critiques de plus haute priorité, il n'y a pas d'interférence due à la transmission d'une trame de priorité inférieure car la préemption est autorisée pour eux. L'extension de FA au protocole TSN fait, ainsi, partie des travaux futurs.

Ces travaux ont donné lieu à une publication dans la conférence internationale [28].

Chapitre 8

Dimensionnement des Files d'attente

Sommaire

8.1	Introduction	119
8.2	Etat de l'art et motivation	120
8.3	Dimensionnement des files d'attente FIFO, sans s�rialisation	121
8.3.1	Sc�nario d'arriv�e des trames	122
8.3.2	Minimisation du nombre de trames transmises	123
8.3.3	Calcul de NbF^h	124
8.4	Dimensionnement des files d'attente FIFO, avec s�rialisation	124
8.5	Exp�rimentations	125
8.5.1	Exemple de calcul de taille de files	126
8.5.2	Cas d'�tude sur les tailles de files d'attente	128
8.5.3	Cas d'une configuration industrielle	128
8.6	Conclusion	131

8.1 Introduction

Dans ce dernier chapitre de contribution, nous nous focalisons sur le problème connexe au délai, celui du dimensionnement des files d'attente. En effet, le déterminisme d'un réseau temps réel implique, notamment, la garantie qu'aucune trame ne se perde. En particulier, les files d'attente doivent être dimensionnées suffisamment grandes pour garantir le non débordement d'une d'entre elles.

Il existe deux possibilités pour exprimer la taille d'une file d'attente :

en nombre de bits : quand l'allocation de la mémoire est dynamique (les demandes d'allocation sont de tailles variables et seul le nombre exact de bits nécessaires est alloué);

en nombre de trames : quand l'allocation est statique (quelle que soit la demande, la taille de l'espace mémoire alloué est toujours le même).

Dans les systèmes temps réel critiques tels que les systèmes avioniques, et afin de réduire la complexité des processus de certification, l'allocation statique d'espace mémoire de taille fixe est privilégiée. Ainsi, dans la suite de ce chapitre, nous déterminons une borne supérieure sur la taille des files d'attente en nombre de trames.

Le nombre maximum de trames présentes dans une file d'attente peut être déduit du nombre de bits maximum, en le divisant par la taille de trame minimale. Cette approche sera dénommée dans la suite approche naïve. Cependant, ce calcul moyen conduit à surestimer l'espace mémoire requis pour les files d'attente. Dans ce chapitre, nous présentons une méthode permettant un calcul moins pessimiste de la taille des files d'attente en nombre de trames pour la politique de service FIFO. Nous avons besoin en entrée :

- une topologie réseau;
- un contrat de trafic : un contrat de trafic maximum garanti à l'entrée du réseau;
- une méthode d'évaluation pire cas des délais.

Ainsi, cette approche est agnostique dans le sens où elle n'est pas développée uniquement pour (ou basée sur) FA mais peut se généraliser à toute méthode d'évaluation de délai pire cas. Le contrat de trafic est formalisé pour un flux ν_i par sa période T_i et par la taille maximale des trames générées par ν_i sur son nœud source, $Fmax_i$.

La suite de ce chapitre est organisée comme suit. Dans le paragraphe 8.2, nous présentons les principaux travaux dans la littérature pour le dimensionnement des files d'attente dans les réseaux commutés. Dans ce même paragraphe, nous justifions l'intérêt de développer une approche spécifique pour le dimensionnement d'une file d'attente en nombre de trames. Dans le paragraphe 8.3, nous présentons notre contribution sur le dimensionnement de files d'attente en termes de trames. Dans la partie expérimentation (le paragraphe 8.5), nous détaillons, dans un premier temps, notre méthode de calcul sur une configuration AFDX avec l'approche FA. Puis nous montrons la pertinence de nos résultats en comparaison avec la méthode naïve sur une configuration AFDX industrielle.

8.2 Etat de l'art et motivation

Dans la littérature, seules quelques méthodes ont été développées pour le dimensionnement des files d'attente et spécialement dans le contexte AFDX. Généralement, ces approches 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 termes de nombre de bits, à partir du plus long temps d'attente qu'elle construit.

Le Network Calculus (cf. paragraphe 4.3.3) a été la première approche développée pour l'AFDX [52]. Elle est, également, celle sur laquelle Airbus se base pour dimensionner ses files d'attente de l'AFDX. La taille maximum de la file d'attente se calcule en déterminant l'écart maximum vertical entre la courbe d'arrivée agrégée des flux en entrée et la courbe de service modélisant la file.

Une autre approche, basée sur la méthode des Trajectoires (cf. paragraphe 4.3.5), calcule, dans un contexte AFDX avec plusieurs niveaux de priorité (une file d'attente par niveau de priorité), la taille maximum des files d'attente en nombre de trames [38]. Cependant, toutes les trames sont supposées de même taille.

La méthode FA détermine l'arriéré maximum temporel présent dans chaque file d'attente. Connaissant la vitesse de service en sortie de la file d'attente, la taille de la file d'attente en nombre de bits en est déduite.

Finalement, toutes ces approches se basent sur l'évaluation de délais de traversée du réseau pour déterminer l'arriéré pire cas en termes de bits. Cependant, nous pouvons démontrer à l'aide d'un contre-exemple que l'arriéré pire cas en termes de trames ne coïncide pas nécessairement avec l'arriéré pire cas en nombre de bits.

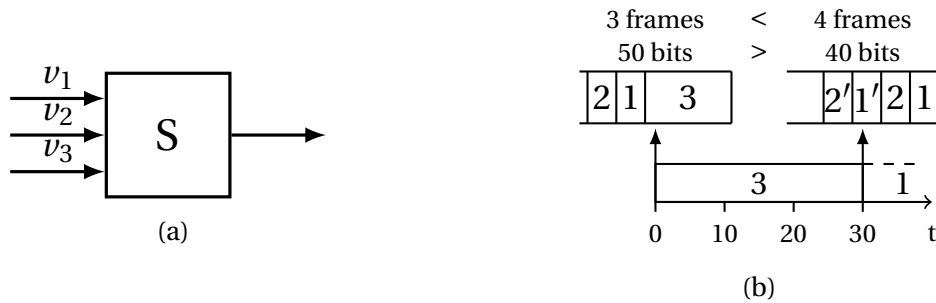


FIGURE 8.1 – Contre-exemple sur la différenciation entre les arriérés pire cas en bits et en trames.

Considérons la configuration représentée par la Fig. 8.1.(a) : trois flux v_1 , v_2 et v_3 , entrant via trois liens différents, traversent un nœud S. Les caractéristiques des flux sont les suivantes :

- $C_1^S = C_2^S = 10 \mu s$ et $C_3^S = 30 \mu s$
- $T_1 = T_2 = 30 \mu s$ et $T_3 = 100 \mu s$.

Avec C_i^S le temps de transmission maximum d'une trame du flux v_i et T_i sa période.

Par souci de simplification, considérons que la file d'attente sert les trames au rythme de 1 bit par μs (ainsi l'arriéré temporel est égal à l'arriéré en nombre de bits). Une trame de chaque flux arrive sur S à l'instant 0. Cet instant correspondant à l'arriéré pire cas en nombre de bits, 50 bits ($C_1^S + C_2^S + C_3^S$) pour trois trames. La trame 3 est servie en premier et à l'instant $t = 30$, elle quitte le nœud. Au même instant deux nouvelles trames 1' et 2'

respectivement générées par les flux ν_1 et ν_2 arrivent dans S (cf. Fig. 8.1b). L'arriéré est alors de seulement 40 bits mais composé de quatre trames (deux de ν_1 et deux de ν_2).

Ainsi, l'arriéré en nombre de trames est décorrélié de celui en nombre de bits. De plus, si nous divisons l'arriéré en bits (50 bits) par la taille de la plus petite trame (10 bits), l'arriéré en nombre de trames est de 5. Cette borne bien que garantie, surestime le dimensionnement nécessaire (4 trames). Par conséquent, déduire le pire cas en nombre de trames à partir de celui en nombre de bits induit du pessimisme dans les tailles calculées.

Déterminer la taille maximum d'une file d'attente en nombre de trames nécessite donc une approche spécifique qui sera développée dans les prochains paragraphes.

8.3 Dimensionnement des files d'attente FIFO, sans sérialisation

Dans ce paragraphe, nous proposons une nouvelle approche dédiée au dimensionnement pire cas des files d'attente FIFO en nombre de trames. La sérialisation n'est pas prise en compte pour le moment. Notons NBF^h le nombre maximum de trames présentes dans la file d'attente du nœud FIFO h . L'objectif principal est de maximiser le nombre de trames présentes dans la file d'attente en :

- maximisant le nombre de trames entrant sur le nœud h ;
- tout en minimisant, en parallèle, le nombre de trames sortant du nœud.

Dans un premier temps, nous nous focalisons sur la description d'un scénario d'arrivée des trames dans une file d'attente FIFO. Considérons, de nouveau, la configuration de la Fig. 8.1.(a) composée d'un seul nœud S et trois flux entrants. Cependant, nous changeons certaines caractéristiques de flux

- $C_1^S = 20 \mu s$, $C_2^S = 10 \mu s$ et $C_3^S = 30 \mu s$
- $T_1 = T_2 = T_3 = 100$.

Dans la Fig. 8.2.(a), nous présentons un scénario possible d'arrivée des trames sur S : une trame de chaque flux arrive au plus tôt, à l'instant 0. La trame 1 est transmise (lorsque plusieurs trames arrivent au même instant, c'est celle avec la plus petit indice qui est transmise) suivie par la trame 2 à l'instant $20 \mu s$. En fonction de leurs historiques dans les nœuds précédents, des trames d'un même flux peuvent arriver sur le nœud S avec un écart inférieur à la période. Ainsi, deux nouvelles trames (une pour le flux ν_1 et une pour ν_3) arrivent sur S respectivement aux instants $40 \mu s$ et $30 \mu s$. L'arriéré maximum du nœud S est de $NBF^S = 3$ trames obtenu aux instants 0 et 40.

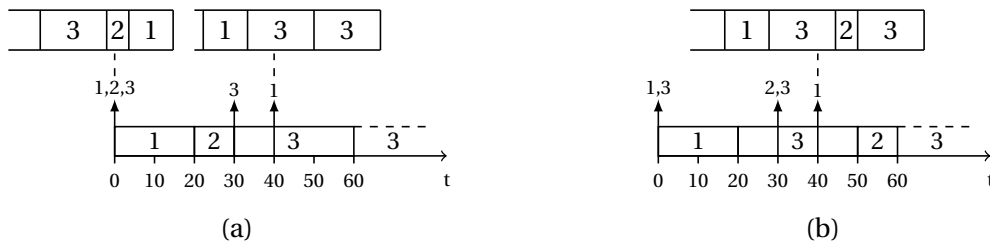


FIGURE 8.2 – Scénarios d'arrivée dans une file d'attente FIFO.

Cependant, il est possible de définir un scénario pire en décalant l'arrivée de la trame 2 à l'instant 30 (cf. Fig. 8.2.(b)), l'arriéré $NbF^S = 4$ est alors obtenu à l'instant 40.

À notre connaissance, il n'existe pas de travaux portant sur la description d'un scénario d'arrivée maximisant le nombre de trames présentes dans une file d'attente FIFO. Ainsi, nous proposons dans la suite, de séparer l'analyse du scénario d'arrivée des trames (cf. paragraphe 8.3.1), de la stratégie pour minimiser le nombre de trames transmises par le nœud dans le paragraphe 8.3.2. Finalement, le calcul complet est récapitulé dans le paragraphe 8.3.3.

8.3.1 Scénario d'arrivée des trames

L'arriéré de travail pire cas généré dans un nœud FIFO h durant un intervalle de longueur t peut être exprimé en utilisant les *Request Bound Function* $rbf_i^h(t)$ définies pour chaque flux $v_i \in \Gamma^h$ (définie dans le paragraphe 5.3.3) :

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

Remarque 7 Où J_i^h est la gigue de v_i dans le nœud h .

Les Request Bound Function ont été formalisées dans le cadre de l'approche FA. Cependant, elles se basent sur la gigue maximale J_i^h qui peut être calculée par n'importe quelle méthode d'évaluation du délai pire cas.

Comme démontré dans le chapitre 5, la fonction $rbf_i^h(t)$ maximise le nombre de trames générées par le flux v_i et arrivant sur le nœud h durant un intervalle de longueur t . Cependant, les instants d'arrivée des trames ne sont pas connus. Ainsi, nous avons démontré dans le paragraphe 6.3.2 qu'il est possible de définir plusieurs scénarios possibles et compatibles avec la fonction $rbf_i^h(t)$. Cependant, le scénario qui maximise le nombre de trames arrivant au plus tôt, dès l'instant 0 est illustré dans la Fig. 6.2.a du chapitre 6. Comme démontré, dans le théorème 3, à l'instant 0, $k_i^h + 1$ trames ($k_i^h = \left\lfloor \frac{J_i^h}{T_i} \right\rfloor$ cf. théorème 2) arrivent simultanément (pas de sérialisation), une nouvelle trame de v_i arrivera à l'instant $k_i^h T_i - J_i^h$, et les trames suivantes arriveront périodiquement toutes les T_i unités de temps. Ce scénario d'arrivée est représenté Fig. 8.3.

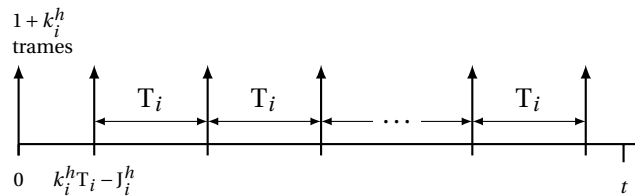


FIGURE 8.3 – Instants d'arrivée au plus tôt des trames de v_i dans une file d'attente FIFO h , en considérant $rbf_i^h(t)$.

Ce scénario d'arrivée, obtenu en utilisant les fonctions rbf_i^h , maximise le nombre de trames générées par un flux v_i arrivées au plus tôt sur un nœud h . Pour déterminer NbF^h , nous devons, maintenant, minimiser le nombre de trames servies par h dans le même intervalle de temps.

8.3.2 Minimisation du nombre de trames transmises

La construction d'un pire scénario qui mène à minimiser le nombre de trames transmises par un FIFO nœud est complexe, à cause du nombre de scénarios à analyser. Ainsi, nous proposons d'utiliser l'algorithme *Largest Processing Time First* LPT [51]. Nous démontrons que LPT est optimal parmi les algorithmes conservatifs (dès qu'au moins une trame est en attente dans la file, il transmet) et non-préemptifs, incluant FIFO, pour la minimisation du nombre de trames transmises (cf. théorème 15).

Théorème 15 *Étant donné un ensemble de flux indépendants générant des trames avec des instants d'arrivée arbitraires dans un nœud h , la politique de service LPT est optimale parmi les algorithmes conservatifs et non-préemptifs pour minimiser le nombre de trames transmises par h .*

Démonstrations : Ce théorème peut être prouvé avec un argument similaire à celui utilisé par Dertouzos [42] pour prouver l'optimalité de l'algorithme EDF pour déterminer un ordonnancement faisable dans les systèmes monoprocesseurs : l'intervention de deux tâches n'augmente pas le retard des tâches considérées.

Considérons un nœud h qui transmet des trames en utilisant une politique de service A différente de LPT. À tout instant t , nous dénotons par $\sigma(t)$ (resp. $LPT(t)$) la trame transmises par A (resp. LPT). La notation $t_{LPT(t)}$ désigne l'instant choisi par A pour transmettre la trame $LPT(t)$. Le temps de transmission des trames est indépendant de la politique de service utilisée.

La politique de service A étant différente de LPT, il existe nécessairement un instant où la décision prise par A diffère de celle de LPT. Nous dénotons par t_1 ($t_1 \geq 0$) le premier instant tel que $\sigma(t_1) \neq LPT(t_1)$. Nous prouvons que l'intervention des transmissions de $LPT(t_1)$ et $\sigma(t_1)$ ne peut pas augmenter le nombre de trames transmises.

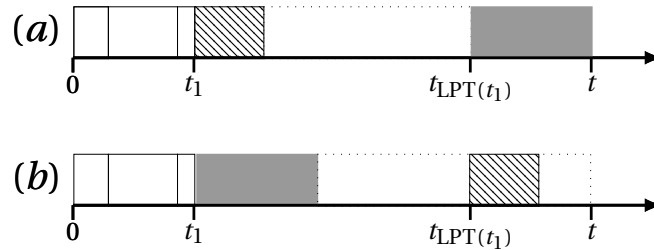


FIGURE 8.4 – Optimalité de LPT pour minimiser le nombre de trames transmises.

La Fig. 8.4 illustre l'intervention entre les transmissions de $LPT(t_1)$ et $\sigma(t_1)$. Dans la Fig. 8.4.(a), l'algorithme A sert la trame $\sigma(t_1)$ (trame hachurée) à l'instant t_1 et $LPT(t_1)$ (trame grisée) à l'instant $t_{LPT(t_1)}$ alors que la transmission de ces trames est inversée pour LPT (cf. Fig. 8.4.(b)).

Durant l'intervalle $[0, t]$ (où t désigne la fin de transmission de la trame $LPT(t_1)$ par A), dans les deux scénarios, le temps disponible pour transmettre les autres trames (espace en pointillés) est identique. Cependant, celui de la Fig. 8.4.(b) est scindé en deux. Ainsi, le nombre de trames transmises par LPT est au plus égal à celui de l'algorithme A .

Par itération, à chaque instant t_1 ($t_1 \geq 0$) tel que $\sigma(t_1) \neq LPT(t_1)$, nous interchangeons la transmission de ces deux trames afin de suivre la politique LPT. À la fin de la transmission de chaque trame interchangée, le nombre de trames transmises en utilisant A est toujours plus grand ou égal à celui obtenu par LPT.

L'algorithme LPT est optimal pour la minimisation du nombre de trames transmises. $\square$

8.3.3 Calcul de NbF^h

Nous récapitulons, dans ce qui suit l'algorithme qui détermine le nombre maximum de trames présentes dans un nœud h , NbF^h qui s'obtient comme la différence entre :

- la courbe d'arrivée cumulative incluant tous les flux $v_i \in I^h$. L'instant d'arrivée au plus tôt de chaque trame est déduit de sa fonction rbf_i^h (voir le paragraphe 8.3.1),
- la courbe cumulative de transmission des trames construite à partir de l'algorithme LPT (voir le paragraphe 8.3.2) et des trames présentes dans h selon la courbe d'arrivée cumulative (établie via les fonctions RBF).

Remarque 8 *La différence entre ces deux courbes est calculée pour tout instant d'arrivée d'une nouvelle trame. Le calcul prend fin au premier point d'intersection entre les deux courbes. Ce qui signifie qu'à cet instant, la file d'attente est vide.*

NbF^h est fixée à la différence maximum.

8.4 Dimensionnement des files d'attente FIFO, avec sérialisation

La sérialisation des flux n'a pas d'influence directe sur la vitesse de transmission des trames mais sur le nombre de trames entrant dans un nœud et donc sur celles qui candidatent à être transmises. En effet, un lien d'entrée limite l'arrivée des trames par sa capacité (cf. paragraphe 5.3.4). Ainsi, dans la suite, pour dimensionner les files d'attente en prenant en compte la sérialisation, le scénario de sortie des trames est complètement indépendant et ne se base pas sur le scénario d'arrivée.

Dans un nœud h , pour chaque lien entrant IP_x^h (avec $1 \leq x \leq n$), nous utilisons la fonction $rbf_i^h(t)$ définie pour chaque flux $v_i \in I_x^h$ pour déterminer les trames prêtes à arriver sur h . Ensuite, nous utilisons l'algorithme *Shortest Processing Time First* SPT [51] pour maximiser le nombre de trames arrivant au plus tôt sur chaque lien entrant.

En effet, maximiser le nombre de trames entrant via un lien IP_x^h (avec $1 \leq x \leq n$) revient à maximiser le nombre de trames transmises par le nœud précédent h (via ce même lien). De plus, selon le même argument que celui développé dans le paragraphe 8.3.2, en FIFO, il est très difficile de déterminer le scénario qui maximise le nombre de trames transmises par un nœud. Ainsi, nous démontrons dans la suite que quel que soit le scénario, l'algorithme SPT est optimal parmi les algorithmes conservatifs et non-préemptifs, incluant FIFO, pour la maximisation du nombre de trames transmises (cf. théorème 16).

Théorème 16 *Étant donné un ensemble de flux indépendants générant des trames avec des instants d'arrivée arbitraires dans un nœud h , la politique de service SPT est optimale parmi les algorithmes conservatifs et non-préemptifs pour maximiser le nombre de trames transmises par h .*

Démonstration : La preuve est similaire à celle du théorème 15. Nous utilisons l'argument d'intervention présenté par Dertouzos [42] et nous démontrons que tout algorithme A qui diffère avec SPT ne peut pas transmettre plus de trames que SPT lui-même. $\square$

Pour un nœud h , l'algorithme du calcul de NbF^h est mis à jour :

- pour chaque lien entrant IP_x^h (avec $1 \leq x \leq n$), la fonction rbf_i^h de chaque flux $v_i \in I_x^h$ est déterminée. Ensuite, l'algorithme SPT est appliqué pour chacun des liens afin de limiter les trames arrivant par la capacité du lien ;
- la fonction rbf_i^h de chaque flux v_i générée par h est calculée et l'instant d'arrivée au plus tôt de chaque trame en est déduit (voir le paragraphe 8.3.1) ;
- la courbe cumulative de transmission des trames construite à partir de l'algorithme LPT (voir le paragraphe 8.3.2) en se basant uniquement sur le scénario d'arrivée non sérialisé.

La condition d'arrêt du calcul reste inchangée. La complexité de l'algorithme est pseudo-polynomiale.

8.5 Expérimentations

Pour les expérimentations, nous appliquons notre méthode de dimensionnement de file d'attente en termes de trames sur une configuration réseau. En entrée, nous initialisons notre approche avec :

- une topologie réseau : la configuration AFDX illustrée dans la Fig. 8.5 ;
- un contrat de trafic : les caractéristiques résumées dans le tableau 8.1 ;
- une méthode d'évaluation pire cas des délais : FA.

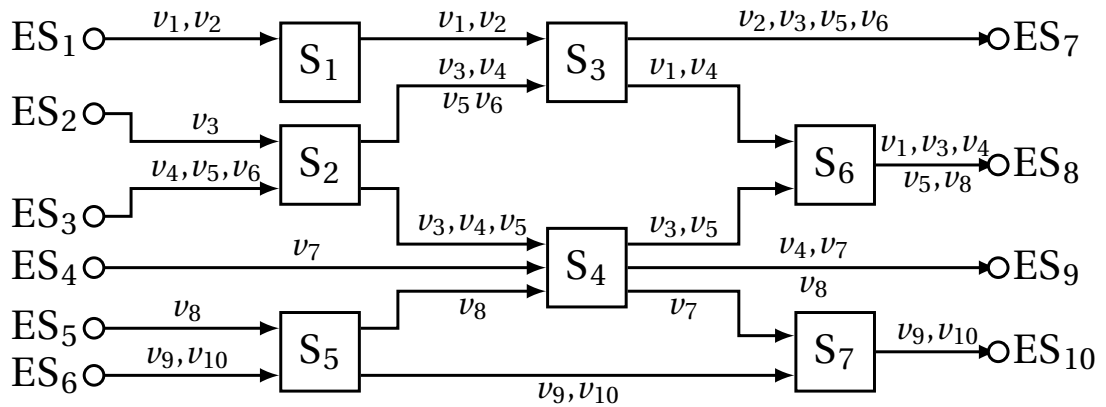


FIGURE 8.5 – Configuration AFDX pour l'analyse du dimensionnement de file d'attente.

Pour la méthode d'évaluation, nous utilisons celle au cœur de cette thèse, FA. De plus, nous analysons notre approche sur la configuration AFDX de la Fig. 8.5 dont la modélisation par FA est dépeinte dans la Fig. 8.6. La politique de service utilisée est FIFO et les caractéristiques des flux sont regroupées dans le tableau 8.1. Tous les liens ont une vitesse de 100Mbps.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
$Fmax_i$ (bits)	1000	2000	1000	1000	2000	2200	6400	2200	1000	1000
T_i (μs)	60	120	100	80	60	60	120	100	80	80

TABLEAU 8.1 – Caractéristiques des flux de la configuration dans la Fig. 8.5.

La seule méthode à laquelle nous pouvons comparer nos résultats est la méthode naïve basée sur le calcul de l'arriéré en termes de bits, divisé par la taille de trame minimum pour en déduire le nombre de trames maximum.

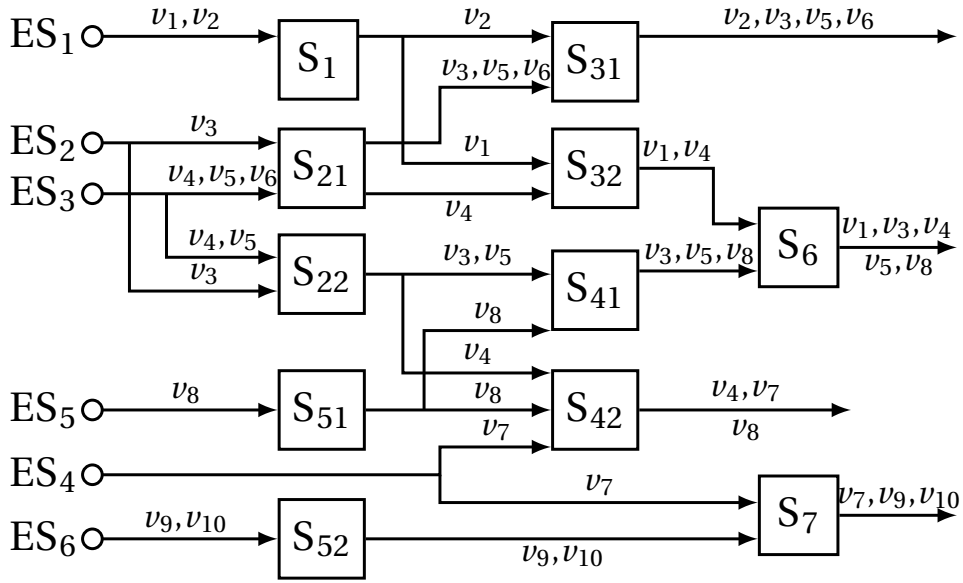


FIGURE 8.6 – La configuration AFDX représenté par la Fig. 8.5 transposé au modèle réseau de FA.

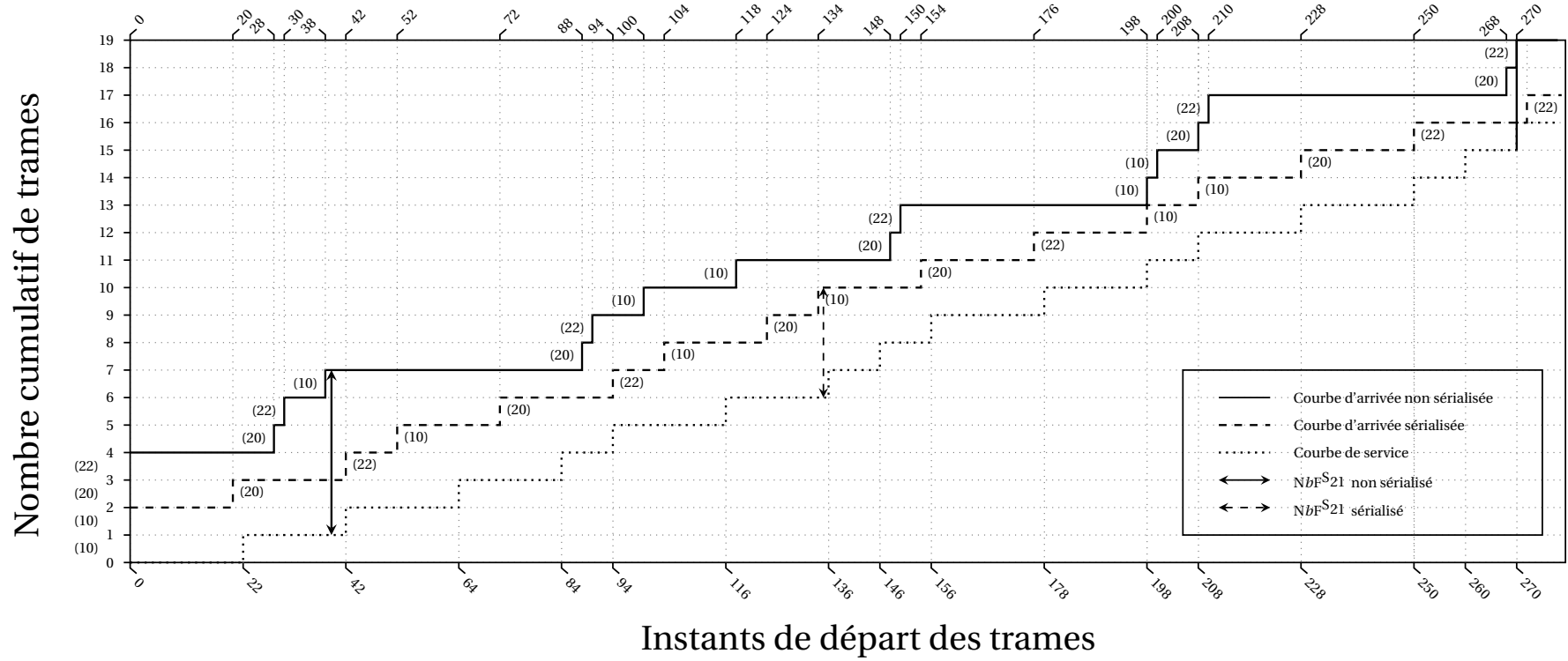
Un exemple détaillé du calcul de la taille d'une file d'attente est décrit dans le paragraphe 8.5.1. Une comparaison des résultats obtenus avec notre approche (version non sérialisée et sérialisée) et avec l'approche naïve est présentée dans le paragraphe 8.5.2 sur un cas d'étude et dans paragraphe 8.5.3 sur une configuration industrielle.

8.5.1 Exemple de calcul de taille de files

Pour détailler le principe de notre méthode, nous nous focalisons sur la file d'attente du nœud S_{21} (cf. Fig. 8.6). Ce nœud est traversé par quatre flux : v_3 , v_4 , v_5 et v_6 arrivant par deux liens différents. Les caractéristiques sont résumées dans le tableau 8.1. Selon FA, considérant la politique de service FIFO, les giques pire cas déterminées pour ces trois flux à leur arrivée dans S_{21} sont :

$$J_3^{S_{21}} = 0 \mu s \quad J_4^{S_{21}} = 42 \mu s \quad J_5^{S_{21}} = 32 \mu s \quad J_6^{S_{21}} = 30 \mu s$$

La Fig. 8.7 trace la courbe de service résultante de l'algorithme LPT ainsi que les courbes d'arrivée cumulatives : la courbe sérialisée suivant l'algorithme SPT, et la courbe déduite des fonctions rbf .

FIGURE 8.7 – Évolution du nombre de trames dans la file d'attente du nœud S_{21} .

En non sérialisé, une trame de chaque flux arrive dans S_{21} à l'instant 0. Lorsque les liens d'entrée sont sérialisés, il ne peut y avoir deux trames qui arrive à l'instant $t = 0$ (une trame par lien d'entrée). Et dans le cas de l'algorithme SPT, c'est la trame la plus petite qui arrive en premier. La courbe de service est quant à elle découplée de la courbe d'arrivée sérialisée. Elle applique l'algorithme LPT sur les trames arrivées suivant la courbe déduite de la RBF. Par conséquent, à l'instant $t = 0$, S_{21} transmet la plus grande trame sensée arrivée à cet instant qui est une trame de ν_6 (avec $C_6 = 22 \mu s$).

Le processus continue jusqu'à ce que la courbe de service croise la courbe d'arrivée sérialisée à l'instant $270 \mu s$, obtenant une différence maximale égale à $NbF^{S_{21}} = 4$ entre les instants 134 et 136. La différence maximale avec la courbe non sérialisée, est obtenue entre les instants 38 et 42 et est égale à $NbF^{S_{21}} = 4$. Avec la méthode naïve, l'arriéré de travail est divisé par la plus taille de trames parmi les flux dans $I^{S_{21}}$. L'arriéré de travail maximum en nombre de bits est égal à 5200 et la plus petite trame a une taille de 1000 bits. Le nombre de trames maximum est donc arrondi à 6 trames, ce qui est supérieure à ce que nous obtenons avec notre proposition. Les valeurs des bornes des autres nœuds de cette configuration sont données dans la prochain paragraphe.

8.5.2 Cas d'étude sur les tailles de files d'attente

Pour chaque nœud h de la configuration dépeinte dans la Fig. 8.6, nous avons déterminé (pour la version non sérialisée puis sérialisée) :

- l'arriéré maximum en nombre de bits obtenus par la méthode FA. Ce nombre a été divisé par la taille minimum des trames traversant chacun de ces nœuds. Ce résultat correspond à la méthode dite *naïve*;
- le nombre de trames pire cas obtenu à l'aide de notre approche NbF^h .

Les résultats de ces expérimentations sont synthétisés dans le tableau 8.2. Concernant les files d'attente des ESs, il n'y a pas d'amélioration des résultats avec la sérialisation car par définition, elle ne s'applique pas dans ce type de nœud. Cependant, pour les nœuds ES_1 et ES_3 , la taille de la file d'attente est plus serrée avec notre approche (ces nœuds génèrent des trames de taille différente).

Que ce soit en sérialisé ou non sérialisé, nous notons plusieurs différences entre la méthode naïve et la nôtre. La différence la plus importante est observée dans le nœud S_{31} et S_{42} car les variations entre les tailles de trame sont les plus importantes. Par exemple, sur le nœud S_{41} la taille des trames varie de 1000 bits à 6400 bits. C'est la raison principale de la surestimation de la méthode naïve.

Les tailles de files d'attente avec la version sérialisée sont inférieures avec ceux de la version non sérialisée car les giges calculées et utilisées dans la définition des fonctions RBF sont plus faibles par construction.

8.5.3 Cas d'une configuration industrielle

Nous avons également appliqué notre méthode de dimensionnement à la configuration AFDX utilisée dans l'A380 et présentée dans [21] et dans le paragraphe 5.4.4. Nous nous sommes une nouvelle fois basé sur la méthode d'évaluation des délais pire cas FA. Nous comparons une nouvelle fois nos résultats avec la méthode naïve. Les résultats sont résumés dans la Fig. 8.8. Chaque barre de l'histogramme exprime la différence en pourcentages entre les bornes obtenues par notre méthode et la méthode naïve. Par exemple,

	Approches par bits				Approches par trames	
	Non sérialisés		Sérialisés		Non sérialisés	Sérialisés
	Arriéré (bits)	Naïve (trames)	Arriéré (bits)	Naïve (trames)	NbF ^h (trames)	NbF ^h (trames)
ES ₁	3000	3	3000	3	2	2
ES ₂	1000	1	1000	1	1	1
ES ₃	5200	6	5200	6	3	3
ES ₄	6400	1	6400	1	1	1
ES ₅	2200	1	2200	1	1	1
ES ₆	2000	2	2000	2	2	2
S ₁₁	3000	3	2000	2	2	1
S ₂₁	7600	8	5200	6	6	4
S ₂₂	4000	4	3000	3	3	2
S ₃₁	13000	13	8200	9	9	7
S ₃₂	3000	3	2000	2	3	2
S ₄₁	6400	7	4200	5	4	3
S ₄₂	9800	10	9600	10	4	4
S ₄₃	6400	1	6400	1	1	1
S ₅₁	2200	1	2200	1	1	1
S ₅₂	2000	2	1000	1	2	1
S ₆₁	12400	13	9600	10	12	9
S ₇₁	8400	9	7400	8	5	3

TABEAU 8.2 – Comparaison des approches de dimensionnement de file d'attente FIFO en nombre de trames.

la taille de file d'attente calculée par notre méthode de dimensionnement est inférieure de 70%, à celle obtenue avec la méthode naïve pour environ 60 ports de sortie.

Pour un seul nœud, la méthode naïve obtient une taille de file d'attente inférieure à la nôtre. En effet, le découplage entre la quantité de trames en entrée et la quantité de trames en sortie induit à des scénarios irréalisables. Néanmoins, pour une configuration industrielle, ce pessimisme ne s'est produit que pour un seul nœud sur plus de 245. Le gain en version sérialisée est, en moyenne, de 54,7% pour notre approche par rapport à la méthode naïve.

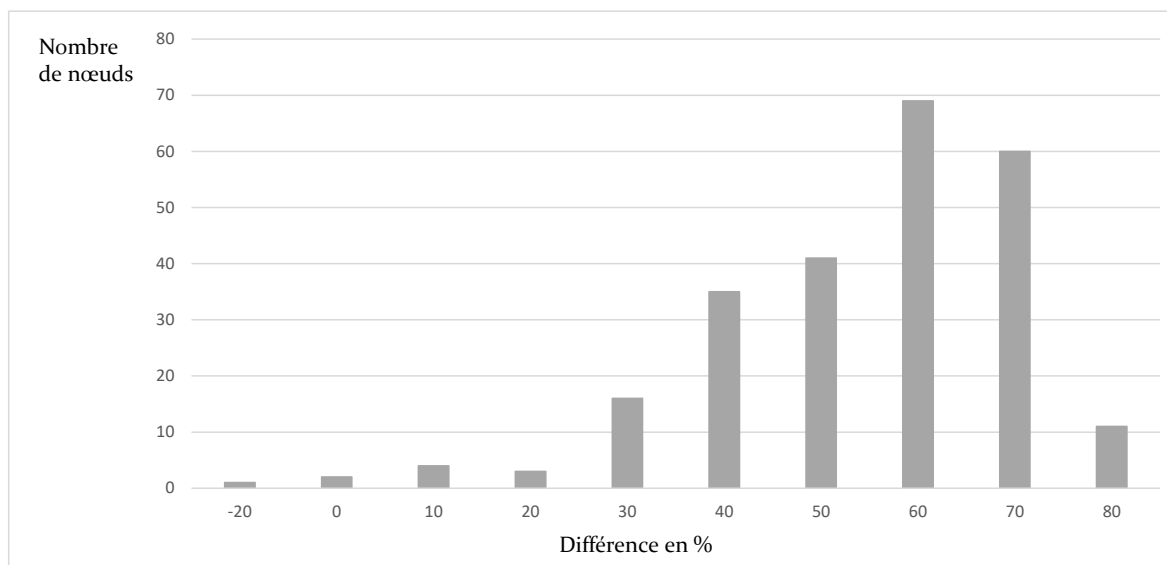


FIGURE 8.8 – Différence en pourcentage entre les bornes pire cas sur le nombre de trames présentes simultanément dans chaque port de sortie de la configuration de l'A380 entre obtenues par notre méthode de dimensionnement par trame et la méthode naïve.

8.6 Conclusion

Le dimensionnement de file d'attente est requis pour la certification dans les réseaux avioniques. L'arriéré pire cas en nombre de bits est utilisé pour l'allocation dynamique de mémoire. Cependant, dans les systèmes avioniques, la conception statique et l'allocation d'espace mémoire de taille fixe sont favorisées. Ainsi, le dimensionnement des files d'attente est exprimé en nombre de trames.

Nous avons développé une approche qui nécessite en entrée : une topologie réseau, un contrat de trafic en entrée pour chaque flux et une méthode d'évaluation des délais. Cette méthode fournit en sortie, une taille de file d'attente garantie. Cette taille a été optimisée en tenant compte de l'effet de sérialisation.

Nous avons appliqué notre approche sur un cas d'étude AFDX ainsi que sur une configuration industrielle. Nous avons comparé nos résultats à ceux de la méthode naïve. De manière générale les gains observés sont notoires en dimensionnement directement les files d'attente en nombre de trames via notre méthode.

Dans les travaux futurs, cette méthode peut être étendue à l'AVB. Cependant, le caractère non oisif de l'algorithme CBS doit être intégré.

Ces travaux ont donné lieu à une publication dans un *workshop* avec comité de sélection associé à une conférence internationale [27].

Chapitre 9

Conclusion

Sommaire

9.1 Principaux résultats	135
9.2 Perspectives	136

Un système temps réel distribué est composé d'un ensemble de sous-systèmes reliés par un réseau pour partager et échanger les données. Dans cette thèse, nous nous sommes focalisés sur l'analyse des réseaux temps réel utilisés dans les domaines de l'avionique et de l'automobile. Pour les systèmes critiques, le déterminisme du réseau doit être démontré : la garantie d'aucune perte et le non dépassement de délai dans le cas nominal (absence de panne).

Nous avons étudié, en particulier, le réseau AFDX [3] utilisé comme réseau cœur des systèmes avioniques développés par Airbus et le réseau AVB [6] qui est une solution d'avenir du domaine de l'automobile. Ces deux protocoles sont basés sur Ethernet commuté : les nœuds (définissant les points d'entrée/sortie) s'échangent des données via des liens physiques et un ensemble de commutateurs. Plusieurs approches ont été définies afin d'analyser les délais de bout-en-bout dans ces réseaux. Nous nous sommes concentrés sur la méthode *Forward ETE delay Analysis* (FA) [66] qui a été développée nativement pour le réseau AFDX et la politique de service FIFO.

9.1 Principaux résultats

Après un état de l'art sur les systèmes temps réel distribués, les réseaux temps réel et les méthodes d'évaluation des délais de bout en bout, nous avons proposé différentes contributions. Chacune d'elle est appuyée par une démonstration formelle et une expérimentation sur un cas industriel. Dans chaque élément d'extension de FA, nous avons intégré l'effet de sérialisation. Ce paradigme bien connu dans la littérature permet d'optimiser les résultats sur le calcul de délai. Nous pouvons les résumer comme suit :

- Initialement développée dans [65], nous avons généralisé le modèle réseau de la méthode FA. Ainsi, les nœuds peuvent simultanément émettre et acheminer des trames. Le réseau peut également être composé de liens physiques avec des débits différents. Nous avons également reformulé le scénario définissant l'interférence maximum subie par la trame sous analyse. Finalement, la condition d'arrêt du calcul selon FA a été réévaluée pour s'appliquer à toutes les politiques de service conservatives. Ce travail a donné lieu à une publication dans une conférence internationale [66] et une revue internationale [30].
- FA a intégré la prise en compte de la différenciation de service en fonction de la priorité des flux. Elle a ainsi été étendue à la politique de service FP/FIFO. Ces travaux ont permis une publication dans une conférence internationale [29].
- FA a été appliquée au standard AVB en incluant une prise en compte de l'algorithme CBS basé sur la consommation de crédit. Ces travaux ont été publiés dans une conférence internationale [28].
- Finalement, nous avons travaillé sur le problème de dimensionnement des files d'attente FIFO. Nous avons ainsi développé une approche. Elle nécessite de connaître la topologie du réseau, les contrats de trafic de chaque flux en entrée et d'une méthode d'évaluation des délais de bout-en-bout. Elle fournit en sortie une borne supérieure garantie de la taille de chaque file d'attente en nombre de trames. Cette approche intègre la prise en compte de l'effet de sérialisation afin d'affiner les résultats. Ce résultat a été publié dans [27].

9.2 Perspectives

L'un des principaux axes pour la poursuite des travaux est l'extension de la méthode FA à la politique de service *Time Aware Shaper* (TAS) employée dans le standard *Time Sensitive Networking* (TSN). Ce standard est principalement basé sur l'AVB et représente l'avenir des systèmes de l'automobile. Cette politique de service est basée sur l'algorithme CBS en incluant le protocole *Time Division Multiple Access* (TDMA) pour la classe des flux critiques les plus prioritaires. Ces flux sont ordonnancés durant des fenêtres temporelles définies statiquement hors-ligne à l'aide du protocole TDMA. Un autre axe intéressant serait de chercher à minimiser le pessimisme induit par FA notamment avec la politique de service FP/FIFO en identifiant plus précisément le flux entrant par chaque lien entrant. Finalement, la partie expérimentation de l'application de FA à l'AVB est assez pauvre en comparaison avec les techniques actuelles telles que TA ou CPA. Nous souhaitons donc prendre le temps d'échanger plus précisément avec les auteurs de ces approches afin de mener des études comparatives plus pertinentes.

Bibliographie

- [1] In vehicle networking solutions- common automotive technologies. <https://www.renesas.com/en-us/solutions/automotive/technology/networking.html>. vii, 31
- [2] I. 11898. *Road Vehicles — Interchange of Digital Information — Controller Area Network (CAN) for High Speed Communication*, 1993. 24, 30
- [3] A. 664. *ARINC 664, Aircraft Data Network, Parts 1,2 7. Technical report, ARINC specification 664.*, 2002-2005. 33, 135
- [4] 802.1AS. 802.1as - timing and synchronization for time-sensitive applications. Technical report, IEEE, 2006. 40, 46
- [5] I. 802.1AS. Ieee standard for local and metropolitan area networks - timing and synchronization for time-sensitive applications in bridged local area networks. *IEEE Std 802.1AS-2011*, pages 1–292, March 2011. 40
- [6] 802.1BA. Ieee standard for local and metropolitan area networks–audio video bridging (avb) systems. Technical report, IEEE, 2011. 33, 39, 99, 135
- [7] I. 802.1Q. Ieee standard for local and metropolitan area networks–bridges and bridged networks. *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)*, pages 1–1832, Dec 2014. 40
- [8] I. 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. *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pages C1–72, Jan 2009. 40, 41
- [9] I. 802.3. Ieee standard for ethernet. *IEEE Std 802.3-2012 (Revision to IEEE Std 802.3-2008)*, pages 1–3747, Dec 2012. 32
- [10] M. Adnan, J.-L. Scharbarg, J. Ermont, and C. Fraboul. An improved timed automata approach for computing exact worst-case delays of AFDX sporadic flows. In *Emerging Technologies and Factory Automation (ETFA)*, Krakow, pages 1–8. IEEE, septembre 2012. 47
- [11] M. Adnan, J.-L. Scharbarg, and C. 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, pages 294–301, <http://www.ieee.org/>, juin 2011. IEEE. 47
- [12] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, and L. L. Bello. 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*, pages 187–194. IEEE, 2012. 99, 107, 115

- [13] G. Alderisi, G. Iannizzotto, and L. L. Bello. Towards ieee 802.1 ethernet avb for advanced driver assistance systems : a preliminary assessment. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–4. IEEE, 2012. [32](#)
- [14] R. Alur. *Techniques for automatic verification of real-time systems*. PhD thesis, stanford university, 1991. [47](#)
- [15] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical computer science*, 126(2) :183–235, 1994. [47](#)
- [16] ARINC. *ARINC 651, Aeronautical Radio Inc. ARINC specification 651. Design Guidance for Integrated Modular Avionics*, 1991. [15](#)
- [17] ARINC. *ARINC 653, Aeronautical Radio Inc. ARINC specification 653 Avionics application Software Standard Interface*, 1997. [15](#)
- [18] ARINC. *Arinc 429. Aeronautical Radio Inc. ARINC specification 429. Digital Information Transfer System (DITS) parts 1,2,3.*, 2001. [29](#)
- [19] AS6802. *As6802 : Time-triggered ethernet*. *SAE International*, 2011. [33](#)
- [20] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, and L. L. Bello. Schedulability analysis of ethernet audio video bridging networks with scheduled traffic support. *Real-Time Systems*, 53(4) :526–577, 2017. [116](#)
- [21] H. Bauer. *Analyse pire cas de flux hétérogènes dans un réseau embarqué avion*. PhD thesis, Institut National Polytechnique de Toulouse, 2011. [76](#), [128](#)
- [22] H. Bauer, J.-L. Scharbarg, and C. 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*, pages 1–8. IEEE, septembre 2009. [53](#)
- [23] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Improving the worst-case delay analysis of an afdx network using and optimized trajectory approach. *IEEE Transactions on Industrial Informatics*, 6(4) :521–533, November 2010. [49](#), [53](#), [77](#)
- [24] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Applying trajectory approach with static priority queueing for improving the use of available afdx resources. *Real-Time Systems, Springer-Verlag*, 48(1) :101–133, january 2012. [49](#), [53](#), [92](#)
- [25] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Worst-case backlog evaluation of avionics switched ethernet networks with the trajectory approach. In *Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on*, pages 78–87. IEEE, 2012. [3](#), [53](#)
- [26] L. L. Bello. The case for ethernet in automotive communications. *ACM SIGBED Review*, 8(4) :7–15, 2011. [19](#)
- [27] N. Benammar, H. Bauer, F. Ridouard, and P. Richard. Tighter buffer dimensioning in AFDX networks. *SIGBED Review*, 13(4) :37–42, 2016. [131](#), [135](#)
- [28] N. Benammar, H. Bauer, F. Ridouard, and P. Richard. Timing analysis of AVB ethernet network using the forward end-to-end delay analysis. In Y. Ouhammou, F. Ridouard, E. Grolleau, M. Jan, and M. Behnam, editors, *Proceedings of the 26th International Conference on Real-Time Networks and Systems, RTNS 2018, Chasseneuil-du-Poitou, France, October 10-12, 2018*, pages 223–233. ACM, 2018. [116](#), [135](#)

- [29] N. Benammar, F. Ridouard, H. Bauer, and P. Richard. 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*, pages 1–8. IEEE, 2017. 96, 135
- [30] N. Benammar, F. Ridouard, H. Bauer, and P. Richard. Forward end-to-end delay for AFDX networks. *IEEE Transactions on Industrial Informatics*, 2018 :858–865, 2018. 77, 86, 135
- [31] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng. 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*, pages 1–10, 2014. viii, 99, 102, 103
- [32] J.-Y. L. Boudec and P. Thiran. *Network calculus : A Theory of Deterministic Queuing Systems for the Internet*, volume 2050. Springer Verlag, 2001. ISBN : 3-540-42184-X. 3, 47, 48, 53, 72
- [33] A. Bouillard, L. Jouhet, and E. Thierry. *Service curves in Network Calculus : dos and don'ts*. PhD thesis, INRIA, 2009. 48, 53, 91
- [34] J. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien. Tight worst-case response-time analysis for ethernet avb using eligible intervals. In *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, pages 1–8, May 2016. 51, 53
- [35] H. Charara. *Évaluation des performances temps réel de réseaux embarqués avioniques*. PhD thesis, Institut National Polytechnique de Toulouse, 2007. 46
- [36] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an afdx network. In *Real-Time Systems, 2006. 18th Euromicro Conference on*, pages 10–pp. IEEE, 2006. 46, 47, 53
- [37] CJLINFO. Les réseaux ethernet. <https://cjl-info.com/historique-des-reseaux-ethernet-c11-p91.html>. 24
- [38] R. Coelho, G. Fohler, and J.-L. Scharbarg. Dimensioning buffers for afdx networks with multiple priorities virtual links. In *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th*, pages 10A5–1. IEEE, 2015. 50, 120
- [39] R. L. Cruz. A calculus for network delay, part i : Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1) :114–131, January 1991. 3, 47
- [40] R. L. Cruz. A calculus for network delay, part ii : Network analysis. *IEEE Transactions on Information Theory*, 37(1) :132–141, January 1991. 3, 47
- [41] J. A. R. De Azua and M. Boyer. Complete modelling of avb in network calculus framework. In *22Nd International Conference on Real-Time Networks and Systems, RTNS '14*, pages 55–64. ACM, 2014. 48, 53, 99
- [42] M. L. Dertouzos. Control robotics :the procedural control of physical processors. *Proceedings of the IFIP congress*, pages 807–813, 1974. 123, 125
- [43] J. Diemer and P. Axer. pycpa : Compositional performance analysis in python. <http://code.google.com/p/pycpa/>, 2012. 51
- [44] J. Diemer, J. Rox, and R. Ernst. Modeling of ethernet avb networks for worst-case timing analysis. *MATHMOD, Austria*, 2012. 51, 53

- [45] J. Diemer, D. Thiele, and 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 2012, Karlsruhe, Germany, June 20-22, 2012*, pages 1–10, 2012. 51, 53, 99, 107, 113, 115
- [46] P. européen de recherche victoria. <http://www.euoproject-victoria.org>. 15
- [47] freescale semiconductors. In-vehicle networking. <https://www.nxp.com/docs/en/brochure/BRINVEHICLENET.pdf>. vii, 20
- [48] L. George, N. Rivierre, and M. Spuri. *Preemptive and non-preemptive real-time uni-processor scheduling*. PhD thesis, Inria, 1996. 49, 53
- [49] F. Geyer, E. Heidinger, S. Schneelee, and A. von Bodisco. Evaluation of audio/video bridging forwarding method in an avionics switched ethernet context. In *Computers and Communications (ISCC), 2013 IEEE Symposium on*, pages 000711–000716. IEEE, 2013. 47, 53
- [50] J. C. Gocłowski and M. Dynamics Research Corp., Wilmington. Digital avionics information system (dais) : Impact of dais concept on life cycle cost : Supplement. final report. *ERIC Clearinghouse*, 1981. Sponsoring Agency : Air Force Human Resources Lab., Brooks AFB, TX. 14
- [51] R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, 17(2) :416–429, March 1969. 123, 124
- [52] J. Grieu. *Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques*. PhD thesis, Université of Toulouse, 2004. 35, 66, 72, 73, 120
- [53] I. A. Grzemba. *MOST : the automotive multimedia network*. Franzis Verlag, 2012. 31
- [54] J. J. Gutiérrez, J. C. Palencia, and M. González Harbour. Response time analysis in afdx networks with sub-virtual links and prioritized switches. *XV Jornadas de Tiempo Real*, January-February 2012. Santander. 49, 53
- [55] M. S. Hans-Werner Schaal. Challenge of ethernet use in the automobile. https://vector.com/portal/medien/cmc/press/Vector/IP_HanserAutomotive_201304_PressArticle_EN.pdf, 2013. 17
- [56] D. Harel and A. Pnueli. On the development of reactive systems. In *Logics and models of concurrent systems*, pages 477–498. Springer, 1985. 11
- [57] F. He, L. Zhao, and E. Li. Impact analysis of flow shaping in ethernet-avb/tsn and afdx from network calculus and simulation perspective. *Sensors*, 17(5) :1181, 2017. 48
- [58] Y. He, G. Siganos, and M. Faloutsos. Internet topology. In *Encyclopedia of Complexity and Systems Science*, pages 4930–4947. Springer, 2009. 26
- [59] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst. System level performance analysis—the symta/s approach. *IEE Proceedings-Computers and Digital Techniques*, 152(2) :148–166, 2005. 51
- [60] O. Hotescu, K. Jaffrès-Runser, J.-L. Scharbarg, and C. Fraboul. Injection de flux non-critiques dans l’avionics full-duplex switched ethernet. -, -. 47, 53
- [61] J. Huysseume and P. Palmer. *Nevada - pamela - victoria*, 2001. 15

- [62] J. Imtiaz, J. Jasperneite, and L. Han. A performance study of ethernet audio video bridging (avb) for industrial real-time communication. In *ETFA*, pages 1–8, 2009. [39](#), [47](#), [53](#)
- [63] J.-B. Itier. A380 integrated modular avionics. http://www.artistembedded.org/docs/Events/2007/IMA/Slides/ARTIST2_IMA_Itier.pdf. [37](#)
- [64] M. Johas Teener, A. Fredette, C. Boiger, P. Klein, C. Gunther, D. Olsen, and K. Stanton. Heterogeneous networks for audio and video : Using ieee 802.1 audio video bridging. *Proceedings of the IEEE*, 101(11) :2339–2354, Nov 2013. [31](#)
- [65] G. Kemayo. *Evaluation et validation des systèmes distribués avioniques*. PhD thesis, Chasseneuil-du-Poitou, Ecole nationale supérieure de mécanique et d’aérotechnique, 2014. [12](#), [135](#)
- [66] G. Kemayo, N. Benammar, F. Ridouard, H. Bauer, and P. Richard. Improving afdx end-to-end delays analysis. In *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, pages 1–8. IEEE, 2015. [3](#), [52](#), [53](#), [77](#), [135](#)
- [67] G. Kemayo, F. Ridouard, H. Bauer, and P. Richard. Optimistic problems in the trajectory approach in fifo context. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–8. IEEE, 2013. [50](#)
- [68] G. Kemayo, F. Ridouard, H. Bauer, and P. Richard. A forward end-to-end delays analysis for packet switched networks. In *22nd International Conference on Real-Time Networks and Systems*, Versailles, France, October 2014. RTNS 2014. [52](#), [53](#), [59](#), [64](#), [65](#), [70](#), [77](#), [86](#)
- [69] M. Knezic, A. Ballesteros, and J. Proenza. Towards extending the omnet++ INET framework for simulating fault injection in ethernet-based flexible time-triggered systems. In A. Grau and H. Martínez, editors, *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation, ETFA 2014, Barcelona, Spain, September 16-19, 2014*, pages 1–4. IEEE, 2014. [99](#)
- [70] M. Lauer. *Une méthode globale pour la vérification d'exigences temps réel : application à l'avionique modulaire intégrée*. PhD thesis, INPT, 2012. [47](#)
- [71] J. Lehoczky. Fixed-priority scheduling of periodic task sets with arbitrary deadlines. *proc. Real-Time Systems Symposium*, pages 201–209, December 1990. [63](#)
- [72] X. Li, O. Cros, and L. 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*, pages 1–10. IEEE, 2014. [50](#), [72](#), [92](#)
- [73] X. Li and L. George. Deterministic delay analysis of AVB switched Ethernet networks using an extended trajectory approach. *Real-Time Systems*, 53(1) :121–186, 2017. [49](#), [53](#), [99](#), [112](#), [113](#)
- [74] H. Lim, D. Herrscher, M. J. Walzl, and F. Chaari. Performance analysis of the IEEE 802.1 ethernet audio/video bridging standard. In G. Riley, F. Quaglia, and J. Himmelsbach, editors, *International ICST Conference on Simulation Tools and Techniques, SIMUTOOLS '12, Sirmione-Desenzano, Italy, March 19-23, 2012*, pages 27–36. ICST/ACM, 2012. [99](#)
- [75] H.-T. Lim, L. Völker, and D. Herrscher. Challenges in a future ip/ethernet-based in-car network for real-time applications. In *Proceedings of the 48th Design Automation Conference*, pages 7–12. ACM, 2011. [47](#), [53](#)

- [76] N. Malcolm and W. Zhao. Hard real-time communication in multiple-access networks. *Real-Time Systems*, 8(1) :35–77, 1995. 29
- [77] S. Martin. *Mastering the time dimension of the quality of service in networks*. PhD thesis, Université de Paris XII, 2004. 3
- [78] S. Martin and P. Minet. Holistic and trajectory approaches for distributed non-preemptive fp/dp* scheduling. In *ICN*, pages 296–305, Berlin Heidelberg, 2005. Springer-verlag. 49, 53
- [79] S. Martin and P. 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*, pages 8–pp. IEEE, 2006. 49
- [80] P. Meyer, T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt. Core4inet. https://core4inet.core-rg.de/trac/wiki/CoRE4INET_Background. 112
- [81] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt. Extending iee 802.1 avb with time-triggered scheduling : A simulation study of the coexistence of synchronous and asynchronous traffic. In *Vehicular Networking Conference (VNC), 2013 IEEE*, pages 47–54. IEEE, 2013. 33
- [82] K.-R. Muller, T. Steinbach, F. Korf, and T. C. Schmidt. A real-time ethernet prototype platform for automotive applications. In *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, pages 221–225. IEEE, 2011. 32
- [83] . Q. Ieee time-sensitive networking task group. Technical report, IEEE, 2016. 33, 41
- [84] . Qbu. 802.1qbu - frame preemption. Technical report, IEEE, 2016. 42
- [85] . Qbv. 802.1qbv - enhancements for scheduled traffic. Technical report, IEEE, 2015. 41
- [86] R. Queck. Analysis of ethernet avb for automotive networks using network calculus. In *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, pages 61–67. IEEE, 2012. 48, 53
- [87] N. Rejeb, A. K. B. Salem, and S. B. Saoud. Afdx simulation based on ttethernet model under omnet++. In *Advanced Systems and Electric Technologies (IC_ASET), 2017 International Conference on*, pages 423–429. IEEE, 2017. 46, 53
- [88] M. R. Report. Automotive electronic control unit (ecu) market size, share and trend analysis report by application type (adas and safety system, chassis electronics, powertrain, infotainment), by region, and segment forecasts, 2018 - 2025. <https://www.grandviewresearch.com/industry-analysis/automotive-ecu-market>. vii, 19
- [89] K. Revsbech, H. Schiøler, T. K. Madsen, and J. J. Nielsen. Worst-case traversal time modelling of ethernet based in-car networks using real time calculus. In *Smart Spaces and Next Generation Wired/Wireless Networking*, pages 219–230. Springer, 2011. 51
- [90] F. Ridouard, J.-L. Scharbarg, and C. 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*, pages 1220–1227. IEEE, 2008. 48

- [91] F. Ridouard, J.-L. Scharbarg, and C. Fraboul. Stochastic upper bounds for heterogeneous flows using a static priority queueing on an afdx network. *Journées FAC*, 2008. 48
- [92] I. Ripoll, A. Crespo, and A. Mok. Improvement in feasibility testing for real-time tasks. *J. Real-Time Systems*, 11(1) :19–39, 1996. 103
- [93] J. Rox and R. Ernst. Formal timing analysis of full duplex switched based ethernet network architectures. Technical report, SAE Technical Paper, 2010. 51, 53
- [94] Y. Song, A. Koubaa, and F. Simonot. Switched ethernet for real-time industrial communication : Modelling and message buffering delay evaluation. In *Factory Communication Systems, 2002. 4th IEEE International Workshop on*, pages 27–35. IEEE, 2002. 32
- [95] M. Spuri. Analysis of deadline scheduled real-time systems. Technical Report 2772, INRIA, 1996. 103
- [96] J. A. Stankovic. Misconceptions about real-time computing : A serious problem for next-generation systems. *Computer*, 21(10) :10–19, 1988. 11
- [97] M. Staron. *Automotive Software Architectures : An Introduction*. Springer, 2017. 17, 19
- [98] T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt. 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, SIMUTools '11*, pages 375–382, ICST, Brussels, Belgium, Belgium, 2011. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 47, 53, 99, 112
- [99] T. Steinbach, H. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz. 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*, pages 1–5. IEEE, 2012. 99, 107, 115
- [100] C. Suthaputchakun, Z. 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*, pages 1–6. IEEE, 2015. 46, 53
- [101] R. Tarek. Automotive bus technologies. <https://www.slideshare.net/RadwaTarek7/automotive-bus-technologies>. vii, 18
- [102] D. Thiele, P. Axer, R. Ernst, and J. R. Seyler. Improving formal timing analysis of switched ethernet by exploiting traffic stream correlations. In *Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, New Delhi, India, October 2014. 47, 51
- [103] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, volume 4, pages 101–104. IEEE, 2000. 51
- [104] K. W. Tindell and J. Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocessing and Microprogramming*, 40(2-3) :117–134, Apr. 1994. 49, 53

- [105] R. Vehicles. *Flexray ISO 17458-1*. ISO, 2013. [31](#)
- [106] R. Vehicles. *LIN ISO 17987-1*. ISO, 2016. [31](#)
- [107] H. Zimmermann. Osi reference model - the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4) :425–432, April 1980. [27](#)

Liste des acronymes

ABS Anti-lock Brake System. [18](#)
ADAS Advanced Driver Assistance System. [19](#)
ADCN Aircraft Data Communication Network. [16](#)
AFDX Avionic Full Duplex Switched Ethernet. [3](#)
AVB Audio Video Bridging Switched Ethernet. [3](#)

BAG Bandwidth Allocation Gap. [34](#)
BE Best Effort. [39](#)

CAN Controller Area Network. [30](#)
CBS Credit Based Shaper. [4](#)
CMI Class Measurement Interval. [109](#)
CPA Compositional Performance Analysis. [51](#)
CPM Core Processing Module. [15](#)
CRC Cyclic Redundancy Checking. [33](#)
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. [30](#)

ECU Electronic Control Unit. [17](#)
ES End System. [34](#)

FA forward end-to-end Delay Analysis. [3](#), [52](#), [135](#)
FCS Frame Check Sequence. [33](#)
FIFO First In First Out. [3](#)
Fmax Maximum frame size. [34](#)
Fmin Minimum frame size. [34](#)
FP/FIFO Fixed Priority/First In First Out. [3](#)

GWM GateWay Module. [15](#)

IMA Integrated Modular Architecture. [15](#)
IOM Input/Output Module. [15](#)

LIN Local Interconnect Network. [31](#)
LPT Longest Processing Time. [123](#)

LRM Line Replaceable Module. [15](#)
LRU Line Replaceable Unit. [14](#)
MAC Medium Access Control. [28](#)
MC Model Checking. [47](#)
MFS Maximum Frame Size. [109](#)
MIF Maximum Interval Frames. [109](#)
MOST Media Oriented Systems Transport. [31](#)
NC Network Calculus. [3](#), [47](#)
OSI Open Systems Interconnection. [27](#)
RTC real Time Calculus. [51](#)
SPT Shortest Processing Time. [124](#)
SR stream Reservation. [40](#)
SRP Stream Reservation Protocol. [40](#)
TA Trajectory Approach. [3](#), [49](#)
TAS Time Aware Shaper. [41](#)
TDMA Time Division Multiple Access. [30](#), [136](#)
TSN Time Sensitive Networking. [33](#)
TTE time Triggered Ethernet. [33](#)
VL Virtual Link. [34](#)

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 .. 88

$Bklg^h$ Arriéré de travail maximum du nœud h .. 61

$Bord_Y^h(t)$ Charge de travail maximum générée par la classe Y dans le nœud h selon le scénario de Bordoloi.. 103

$\mathcal{BP}_Y^h$ La plus longue période d'activité des classes plus prioritaires que la classe Y dans le nœud h .. 102

C_i^h Temps de transmission maximum du v_i dans le nœud h .. 60

$Crd_{Y,max}^h(t)$ Crédit maximum de la classe Y dans le nœud h étant donné un intervalle de durée t .. 103

$dep_i^h(t)$ Instant de départ pire cas d'une trame de v_i arrivée au nœud h à l'instant t .. 83

$first_i$ Nœud source du flux v_i .. 60

$Fmax_i$ Taille de trame maximum du flux v_i .. 60

$HP_{X_i}^h$ Ensemble des classes plus prioritaires que la classe X_i au niveau du nœud h .. 101

hp_i^h Ensemble des flux plus prioritaires que le flux v_i au niveau du nœud h .. 81

IP_x^h Port d'entrée x du nœud h .. 67

h_{x-1} Nœud précédant du nœud h à travers le lien d'entrée IP_x^h .. 67

J_i^h Gigue du v_i au niveau du nœud h .. 66

r^h Taux de service du nœud h .. 60

k_i^h Nombre de trames maximum générée par le flux v_i arrivant simultanément dans le nœud h .. 64

$last_i$ Dernier nœud traversé par v_i dans $\mathcal{P}_i$.. 60

$LinkRate_{i,x}^h(t)$ Charge de travail maximum sérialisée en provenance de Ip_x^h générée durant t dans le nœud h interférant avec le flux v_i .. 71

$LP_{X_i}^h$ Ensemble des classes moins s que la classe X_i au niveau du nœud h .. 101

lp_i^h Ensemble des flux moins prioritaires que le flux v_i au niveau du nœud h .. 81

NBF^h Nombre de trames maximum présentes simultanément dans le nœud h .. 121

- $\mathcal{P}_i$ Liste ordonnée des nœuds traversés par v_i .. 60
- P_i Niveau de priorité du flux v_i .. 81
- $rbf_i^h(t)$ RBF du flux v_i dans le nœud h pour un intervalle de durée t .. 66
- $rbf_x^h(t)$ Somme des RBFs des flux appartenant Γ_x^h pour un intervalle de durée t .. 71
- R_i^h Délai de traversée pire cas de v_i de $first_i$ au nœud h .. 61
- $HP_{X_i}^h$ Ensemble des classes $SP_{X_i}^h$ et $HP_{X_i}^h$.. 101
- shp_i^h Ensemble des flux sp_i^h et hp_i^h .. 82
- $\sigma_{X_i}^{h,-}$ SendSlope de la classe X_i .. 100
- $\sigma_{X_i}^{h,+}$ IdleSlope de la classe X_i .. 100
- $Smax_i^h$ Délai de traversée pire cas de v_i de $first_i$ pour atteindre le nœud h .. 61
- $Smin_i^h$ Délai de traversée minimum de v_i de $first_i$ pour atteindre le nœud h .. 62
- $SP_{X_i}^h$ Ensemble des classes de même priorité que la classe X_i au niveau du nœud h .. 101
- sp_i^h Ensemble des flux de même priorité que le flux v_i au niveau du nœud h .. 81
- T_i Période du flux v_i .. 60
- $WHP_i^h(t)$ Charge de travail maximum générée par hp_i^h durant un intervalle de durée t dans le nœud h .. 89
- $W_i^h(t)$ Charge de travail maximum générée durant un intervalle de durée t dans le nœud h interférant avec le flux v_i .. 83
- $WLP_i^h(t)$ Charge de travail maximum générée par lp_i^h durant un intervalle de durée t dans le nœud h .. 84
- $WNIL_i^h(t)$ Temps de réapprovisionnement du crédit des $WSP_i^h(t)$ en comptant la charge minimum des hpX .. 101
- $WSP_i^h(t)$ Charge de travail maximum générée par sp_i^h durant un intervalle de durée t dans le nœud h .. 85
- $W^h(t)$ Charge de travail maximum générée durant un intervalle de durée t dans le nœud h .. 62
- $W_x^h(t)$ Charge de travail maximum en provenance de IP_x^h générée durant un intervalle de durée t dans le nœud h .. 67
- X_i Classe SR à laquelle appartient le flux v_i .. 101