



HAL
open science

De l'usage d'architectures Ethernet commutées embarquées dans les lanceurs spatiaux

Jérémy Robert

► **To cite this version:**

Jérémy Robert. De l'usage d'architectures Ethernet commutées embarquées dans les lanceurs spatiaux. Informatique [cs]. Université de Lorraine, 2012. Français. NNT : 2012LORR0128 . tel-01749322

HAL Id: tel-01749322

<https://hal.univ-lorraine.fr/tel-01749322>

Submitted on 2 Nov 2022

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

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



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Thèse
présentée pour l'obtention du titre de
Docteur de l'Université de Lorraine
en Sciences, spécialité Automatique,
Traitement du Signal et Génie Informatique
par **Jérémy ROBERT**

De l'usage d'architectures Ethernet commutées embarquées dans les lanceurs spatiaux

Soutenue publiquement le 23 octobre 2012

Membres du jury :

Rapporteurs : Anne BARROS-LEFEBVRE Professeur Université Technologique de Troyes, LM2S
Christian FRABOUL Professeur Institut National Polytechnique de Toulouse, IRIT

Examineurs : Jean-François PÉTIN Professeur Université de Lorraine, CRAN
Jean-Philippe GEORGES Mcf Université de Lorraine, CRAN
Philippe MIRAMONT Ingénieur CNES - Direction des Lanceurs, Paris
Thierry DIVOUX Professeur Université de Lorraine, CRAN (directeur de thèse)

Remerciements

J'adresse, en premier lieu, mes plus sincères remerciements à Thierry DIVOUX et Jean-Philippe GEORGES. Tous les deux, jamais avares de conseils, m'ont permis de réaliser et de réussir cette thèse. Il est vrai que s'ils ne m'avaient pas transmis le goût de la recherche (notamment pendant le master), je n'aurais jamais songé à faire une thèse... je les remercie chaleureusement.

J'associe également à ces premiers pas dans la recherche, Eric RONDEAU que je remercie également de m'avoir permis de réaliser un stage de recherche enrichissant et d'avoir contribué à mon à l'envie de poursuivre en thèse.

Je remercie également le directeur du laboratoire Alain RICHARD, par son accueil convivial mais également pour ces cours en Master, pendant lesquels, il a éveillé en moi l'envie d'aller plus loin dans les différentes réflexions scientifiques.

Je tiens également à remercier toutes les personnes du département Génie Electrique et Informatique Industrielle à l'IUT Nancy-Brabois qui m'ont aidées pour mes différents TP et TD, Frédéric CHAXEL, Christian SAUNAL, Cédric JOIN. Je remercie particulièrement Frédéric SARRY et Taha BOUKHOBZA de m'avoir recruté comme moniteur dans un département dynamique. Merci également à Caroline pour son appui administratif.

Mes remerciements s'adressent également aux différentes personnes avec qui j'ai partagé de bons moments à la BSR ou pendant les pauses. J'ai une pensée particulière pour : Leila, Cheick, Gilbert sans oublier tous les autres : Gaby, Pierre, David, Pascale, Esma, Idriss, Alex R., Thomas, Ludovic, Fabien, Damien, Mario, William, Nicolas S., Alex V., ...

Un grand merci à mon collègue de bureau, Sylvain KUBLER, avec qui je me suis bien entendu depuis les premières années post-bac. Toutes les discussions, qu'elles soient scientifiques ou personnelles ont été intéressantes et enrichissantes. Pour son aide, son soutien et nos apartés, je le remercie sincèrement.

Tout travail de recherche mérite d'être critiqué pour s'enrichir. Pour cela, je remercie les rapporteurs Anne BARROS et Christian FRABOUL pour leurs remarques et/ou questions constructives que ce soit dans leur rapport ou lors de la soutenance. Je remercie également Jean-François PETIN, président du jury lors de la soutenance, pour ses questions et suggestions pertinentes.

Je remercie également les personnes du CNES, Philippe MIRAMONT et Badr RMILI, avec qui les échanges ont été intéressants et fructueux.

Merci à ma famille, et plus particulièrement à mes parents : pour tout, mais vous le savez déjà!

Table des matières

Introduction	vii
1 Du bus MIL-STD-1553B au réseau Ethernet pour les lanceurs spatiaux	1
1.1 Introduction	1
1.2 Le bus actuel : MIL-STD-1553	2
1.2.1 Architecture du bus MIL-STD-1553 B	2
1.2.2 Notion de transfert	2
1.2.3 Format des messages MIL-STD-1553B pour le lanceur	3
1.2.4 Organisation temporelle des transferts sur les lanceurs	4
1.2.5 Commentaires	4
1.3 Des nouveaux besoins du spatial aux réseaux Ethernet	5
1.3.1 Des nouveaux besoins du spatial	5
1.3.2 . . . aux réseaux Ethernet	6
1.4 Analyse de solutions temps-réel basées sur Ethernet (Robert <i>et al.</i> , 2012)	7
1.4.1 Introduction	7
1.4.2 EtherCAT	8
1.4.3 Profinet IRT	12
1.4.4 Modbus/TCP	14
1.4.5 Ethernet/IP	16
1.4.6 Comparaisons des solutions	19
1.5 Discussions	22
1.6 Conclusion	24
2 Un premier « essai en vol » d’Ethernet	25
2.1 Introduction	25
2.2 Les exigences des réseaux embarqués dans les lanceurs spatiaux	25
2.3 Premier « essai en vol » d’Ethernet	30
2.3.1 Analyse des communications sur le bus MIL-STD-1553B	30
2.3.2 Propositions d’architectures Ethernet commutées	31
2.3.3 Evaluation temporelle des architectures proposées	38

2.4	Conclusion	43
3	Distribuer une avionique sur Ethernet ? (Robert <i>et al.</i>, 2011)	45
3.1	Introduction	45
3.2	Evaluation des majorants des délais	46
3.2.1	Outils d'évaluation des délais	46
3.2.2	Modélisation des trafics en fonction du scénario de communication	47
3.2.3	Modélisation du service élémentaire	49
3.2.4	Modélisation du système de communication	52
3.2.5	Calcul d'un majorant du délai de bout en bout	56
3.3	Evaluation des architectures tests	56
3.3.1	Architecture centralisée	57
3.3.2	Architecture distribuée	64
3.4	Conclusion	69
4	Assurer la disponibilité du système de communication	71
4.1	Introduction	71
4.1.1	« Redondance dans le réseau »	73
4.1.2	« Redondance dans les nœuds »	74
4.1.3	Commentaires	74
4.2	Proposition centralisée pour la reconfiguration d'une architecture Ethernet	75
4.2.1	Détection de la défaillance	76
4.2.2	Reconfiguration	77
4.2.3	Formalisation	77
4.2.4	Expérimentations	79
4.3	Proposition d'une solution multi-chemins (sans reconfiguration topologique)	87
4.3.1	Introduction	87
4.3.2	Détermination de la fonction objectif	88
4.3.3	Algorithmes d'optimisation	90
4.3.4	Expérimentations sur un Système Controlé en Réseau (SCR)	95
4.3.5	Synthèse	97
4.4	Conclusion	98
5	Observer pour « jouer au sol » et isoler les défaillances	101
5.1	Introduction	101
5.2	Observabilité d'un réseau Ethernet commuté	103
5.2.1	Collecte des échanges	103
5.2.2	Fusion des différentes collectes	104
5.3	Isolation des défauts	118

5.3.1	Expérimentation 1	119
5.3.2	Expérimentation 2	121
5.4	Conclusion	123
Conclusions & Perspectives		127
A Evaluation de candidats concurrents en vue d'un démonstrateur		131
A.1	Introduction	131
A.2	Listes des critères de sélection	132
A.3	Présentation des candidats	135
A.3.1	Candidats 1 et 2 : <i>Fibre Channel</i> avec FC-AE-1553	135
A.3.2	Candidat 3 : Ethernet standard commuté	137
A.3.3	Candidat 4 : AFDX	137
A.3.4	Candidat 5 : <i>Time-Triggered Ethernet (TT-Ethernet)</i>	139
A.3.5	Candidats 6 et 7 : <i>Time triggered protocol (TTP/C)</i> et Flexray	140
A.3.6	Candidats 8 et 9 : Arion-100 et Mémoire réfléchive	140
A.4	Evaluation des candidats	141
A.5	Discussions	148
A.6	Conclusion	149
Liste des publications		151
Notice bibliographique		153

Introduction

Les besoins des utilisateurs des réseaux informatiques sont en perpétuelle augmentation à tous les niveaux d'application : du réseau de terrain (capteurs, actionneurs . . .) aux réseaux de gestion (*Enterprise Resource Planning, Manufacturing Execution System*), en passant notamment par les niveaux « cellule » et « atelier ». Bien que le standard Ethernet (développé depuis les années 80) se soit rapidement imposé dans les réseaux informatiques, son utilisation en milieu industriel a été plus lente et progressive. En effet, les industriels étaient réticents à adopter ce standard à cause de son indéterminisme intrinsèque (Decotignie, 2005). L'incapacité à maîtriser les délais de manière déterministe accentuait ce refus dogmatique. L'avènement des commutateurs (IEEE Computer Society, 1998) a permis d'éliminer cet indéterminisme : chaque équipement peut être relié à un commutateur rendant impossible toute collision (*full-segmentation*). La qualification des performances d'une architecture Ethernet peut ensuite être réalisée de manière déterministe en évaluant des majorants des délais (Georges *et al.*, 2002) (Jasperneite *et al.*, 2002). Les contraintes de réduction des coûts, de modularité, d'évolutivité et de performances ont alors contribué cette dernière décennie à favoriser l'emploi d'Ethernet dans l'industrie pour interconnecter des applications de contrôle / commande à fortes contraintes temporelles (ateliers de fabrication, machines outils (Alliance et Lüder, 2005)). Jusqu'alors, ces applications industrielles reposaient sur des solutions de bus de terrain (comme FIP et Profibus) définies dans le standard IEC 61158 (IEC, 2003a). Ce standard est maintenant complété par le standard IEC 61784 (IEC, 2003b) dans lequel une multitude de solutions basées sur Ethernet sont intégrées .

Forts de cette expérience en milieu industriel, les systèmes embarqués et en particulier l'aéronautique se sont également orientés vers de l'Ethernet. Comme la plupart des applications (quelque soit le contexte), la quantité de données échangées augmente considérablement. En effet, les fonctions embarquées sont toujours de plus en plus nombreuses. Outre cette augmentation, les architectures centralisées évoluent vers des architectures distribuées entraînant une modification du mode de communication (et des échanges potentiellement plus importants). Les performances des bus avioniques embarqués (tels que ARINC 429, ARINC 629, STANAG 3910, etc) sont alors limitées par la bande passante intrinsèque (de 100 Kb/s à 20 Mb/s) et par leur caractère trop propriétaire. Des travaux ont alors été menés de manière à adapter la technologie Ethernet aux contraintes de l'aéronautique tout en profitant de ses avantages (en terme notam-

ment de coût grâce à l'utilisation de COTS - *Components Off-The-Shelf* et de bande de passante). Ainsi, il est possible de citer l'AFDX (*Avionics Full Duplex switched Ethernet* normalisé dans la norme ARINC 664) ou encore TT-Ethernet (*Time-Triggered Ethernet* qui devrait être normalisé prochainement sous le nom SAE AS6802). Le milieu aéronautique requiert que l'ensemble des systèmes embarqués soit qualifié. Par exemple, l'AFDX a été certifié par la théorie du *network calculus* (Grieu, 2004) pour l'A380. Ces évolutions dans le milieu aéronautique ont naturellement conduit à une réflexion quant à l'utilisation d'Ethernet commuté dans les systèmes spatiaux.

Actuellement, la plupart des systèmes spatiaux (lanceurs, satellites) possèdent plusieurs réseaux (un réseau de commande et un réseau de service) destinés à transporter chacun un trafic spécifique. La problématique de réduction des coûts et de la masse du système global (et donc des systèmes de communications) contribue à la volonté de s'appuyer sur une (et une seule) technologie permettant une utilisation de composants matériels et d'outils de développement existants à moindre coût (COTS). Dans ce contexte, les différents réseaux seront unifiés pour ne former qu'un seul système de communication qui devra alors supporter des trafics de plus en plus nombreux voire volumineux (comme la transmission d'images, de vidéos ou encore des données de télémétrie avancée). Par conséquent, le bus avionique *MIL-STD-1553B* (Department of Defense, 1978) embarqué dans les lanceurs occidentaux actuels constitue une limitation notable en terme de bande passante (1 Mb/s).

La particularité des systèmes spatiaux est d'évoluer dans un environnement fortement contraint, agressif et dynamique. D'une part de nombreuses vibrations et rayonnements hors atmosphère sont engendrés lors de la poussée, ce qui peut provoquer des défaillances (rupture des connecteurs, instabilité binaire ...). Et d'autre part, les étages du lanceur se séparent au fur et à mesure de l'avancement du vol, entraînant une modification de la topologie du réseau à chaque séparation.

A partir de ce constat, le Centre de Recherche en Automatique de Nancy (CRAN) a réalisé, en 2007, pour le Centre National d'Etudes Spatiales (CNES) une veille technologique sur les nouveaux systèmes de communication. A travers trois critères énoncés par le CNES (déterminisme, modélisation/simulation et tolérance à la séparation d'étage), plusieurs tendances ont été isolées : l'Ethernet commuté, les réseaux sur Fibre Optique et les réseaux sans fil. Après analyse de la veille, le CNES a choisi de confier au CRAN l'étude de l'Ethernet commuté comme technologie de remplacement du bus de communication actuel. Fort de cette première contribution du laboratoire, une collaboration entre le CNES et le CRAN s'est engagée à travers deux contrats de Recherche & Technologies (R&T) qui ont fait l'objet de mes travaux de thèse.

L'utilisation d'architectures Ethernet commutées dans le contexte spatial implique la prise en compte de contraintes du milieu aéronautique mais en ajoute également de nouvelles. On retrouve donc des contraintes temporelles et de disponibilité du système de communication communes à plusieurs domaines (spatial, industrie, aéronautique) et des contraintes plus spécifiques au do-

maine spatial telles que l'observabilité pour l'isolation de défauts du système de communication. Ainsi, il est nécessaire de proposer une architecture (redondante) satisfaisant aux contraintes temps-réel et de disponibilité. En outre, une contrainte originale dans le spatial est l'observabilité de tout (ou partie) des échanges en tout temps et en tout lieu. Bien que cette observabilité peut aisément se comprendre en phase de test et de certification (étape ultime à la progression d'une nouvelle technologie), celle-ci s'avère également (et surtout) nécessaire en phase opérationnelle afin de déceler, comprendre, analyser voire « jouer au sol » la moindre défaillance (ou le vol même sans défaillance).

Dans ce contexte, nous avons proposé et étudié, sur des architectures Ethernet commutées, comment :

- distribuer l'avionique (et d'en évaluer le gain),
- assurer la disponibilité du système de communication et
- observer pour « jouer au sol » et isoler les défaillances.

A la fin de ces travaux de R&T, le CNES a initié un projet de démonstrateur nommé Avionique-X pour la future génération de lanceurs (Monchaux *et al.*, 2012). Ce projet est financé par un Plan d'Investissement pour l'Avenir (PIA) avec le CNES comme maître d'ouvrage et ASTRIUM Space Transportation comme maître d'œuvre. De manière à définir la (ou les) solution(s) à mettre en œuvre sur le démonstrateur, une évaluation de plusieurs candidats a été réalisée sur 22 critères dans laquelle on retrouve notamment les critères étudiés dans la R&T. Le CRAN y a participé comme expert.

Organisation du document

Le premier chapitre vise à identifier le contexte des travaux. Le système de communication actuel y est présenté. Les nouveaux besoins du spatial montrent la limitation du bus actuellement embarqué dans Ariane 5. A travers un bref historique de l'évolution des réseaux locaux industriels, il est montré l'emploi d'Ethernet comme solution aux nouveaux besoins. L'étude porte à ce stade sur la vérification temporelle de solutions Ethernet pour les lanceurs.

Le chapitre 2 présente les différentes exigences propres au système de communication du domaine spatial. Le passage d'un bus à une architecture commutée implique la proposition de nouvelles architectures de communication. Un premier « essai en vol » d'Ethernet a été mené. Celui-ci consiste à « jouer » et évaluer les performances du scénario applicatif actuel sur les architectures Ethernet commutées proposées.

Le chapitre 3 propose une quantification des gains de la distribution sur les performances temporelles d'architectures Ethernet commutées pour les lanceurs spatiaux. Cette quantification mettra notamment en avant les majorants des délais de bout en bout.

Le chapitre 4 propose des solutions permettant d'assurer la disponibilité du réseau même en cas de défaillance d'un élément réseau. Une première proposition consiste à réaliser *a posteriori* la reconfiguration d'une topologie active, tout en recouvrant des performances similaires en termes de temps d'interruption de service aux solutions s'appuyant sur de la « redondance dans le réseau ». La seconde solution s'appuie sur l'utilisation simultanée de plusieurs chemins permettant d'annuler ce temps d'interruption de service.

Le chapitre 5 se concentre sur l'observabilité d'un réseau Ethernet commuté et isolation de défauts. A partir de la mise en place d'observateurs locaux en plusieurs points du réseau (*a contrario* à l'unique point d'observation nécessaire sur un bus), il a été proposé une solution pour recouvrer une observation globale du comportement du réseau.

Le chapitre 6 va permettre de reprendre les résultats présentés et d'en tirer des limites (notamment d'interprétation) afin de faire émerger des perspectives pour les lanceurs spatiaux. Cela est réalisé à travers l'évaluation de candidats concurrents en lice pour être implémentés sur la plateforme de démonstration Avionique-X.

Chapitre 1

Du bus MIL-STD-1553B au réseau Ethernet pour les lanceurs spatiaux

1.1 Introduction

Lors du choix initial du système de communication pour les lanceurs spatiaux (et plus particulièrement Ariane 5), des besoins spécifiques ont d'abord été énoncés. Le système de communication devait alors reposer sur un standard connu (notamment pour réduire les coûts de conception, d'exploitation,...) répondant à un besoin d'intégrité, d'observation du trafic, de débit supérieur à 300 *Kb/s*, etc. Le choix s'est alors porté sur le bus avionique multiplexé MIL-STD-1553B (Department of Defense, 1978) développé et standardisé dans les années 70. L'étude du protocole a mené à la définition de spécificités d'utilisation du système de communication pour le lanceur telles que l'architecture (centralisée et redondante), l'utilisation de répéteurs de bus ou encore la définition du trafic sur le bus.

Cependant, la stratégie actuelle consiste à proposer un nouveau lanceur spatial moins cher et moins lourd. Ainsi, l'utilisation d'une technologie reposant sur des composants matériels et d'outils de développement existants à moindre coût (COTS) pourra permettre de réduire le critère économique. D'autre part, l'unification du réseau de commande et du réseau de métrologie sur un seul système de communication aura *a priori* un impact sur la masse du réseau. Néanmoins, cette unification implique que le système de communication soit capable de transporter des trafics plus nombreux voire plus volumineux (comme de la transmission d'images et de vidéos ou encore des données de télémétrie avancée). Le bus MIL-STD-1553B actuellement embarqué dans les lanceurs spatiaux est alors limité en terme de bande passante (1 *Mb/s*). Par conséquent, et à l'instar du monde industriel, beaucoup d'intérêt a été porté au standard Ethernet.

Ce chapitre a donc pour objectif de présenter, d'une part le bus avionique actuel et des solutions industrielles basées sur Ethernet d'autre part. La présentation des différentes solutions

industrielles se fera sous la forme d'une analyse de performances temporelles de manière à montrer les capacités de solutions s'appuyant sur Ethernet en vue d'une étude approfondie au cas des lanceurs.

1.2 Le bus actuel : MIL-STD-1553

Dans cette section, l'objectif est de présenter les grandes lignes du fonctionnement du bus actuellement utilisé sur les lanceurs Ariane 5.

1.2.1 Architecture du bus MIL-STD-1553 B

Le standard MIL-STD-1553B définit trois types d'équipements : un gérant de bus (*Bus Controller* ou BC), des abonnés au bus (*Remote Terminal* ou RT) et un (ou des) espion(s) (*Monitor*). Le gérant de bus initie l'ensemble des échanges à travers un mécanisme de commande/réponse. Il n'y a alors aucune collision possible entre les paquets puisque l'abonné ne peut émettre des données que lorsqu'il est sollicité par le gérant de bus. L'abonné est caractérisé par son adresse (de 0 à 30 soit 31 équipements maximum sur le bus) et des sous-adresses (de 1 à 30, 0 et 31 étant réservées à des commandes de services) relatives aux sous-systèmes (capteurs, actionneurs, ...). Le rôle de l'espion est de réaliser l'observation (et l'archivage) de tout ou partie (notion de filtrage) des transferts sur le bus. Notons que des répéteurs de bus sont utilisés dans l'architecture de manière à gérer les désadaptations d'impédance lors des séparations d'étages. Le bus est redondé comme illustrée à la FIGURE 1.1.

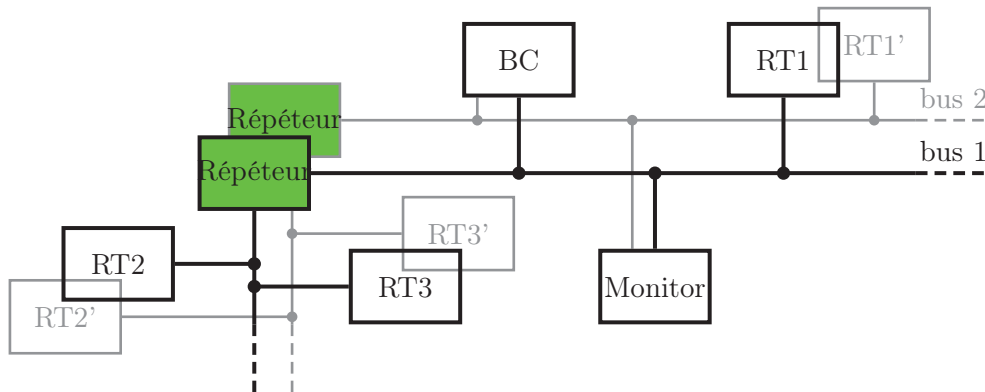


FIGURE 1.1: Illustration d'architecture du bus MIL-STD-1553 sur les lanceurs

1.2.2 Notion de transfert

Un transfert est un échange d'informations entre le gérant et les abonnés. Deux types de transferts sont définis en fonction du producteur de la donnée à transmettre (le gérant ou l'abonné).

On parle de transfert Gérant - Abonné (GA) lorsque le producteur de la donnée est le gérant de bus et de transfert Abonné - Gérant (AG) dans le cas inverse.

Pour gérer ces transferts, le protocole met en jeu trois types de messages de 20 *bits* (dont 3 *bits* de synchronisation et 1 de parité) : le mot de commande, le(s) mot(s) de données et le mot de statut. Pour chaque transfert (hors commande de service sans données), il y aura au minimum ces trois mots échangés (au minimum car il se peut qu'il y ait plus d'un mot de données). L'ordonnement de ces messages est précisé sur la FIGURE 1.2 selon les deux types de transferts existants.

Échanges Abonné - Gérant :



Échanges Gérant - Abonné :



Comm. : Mot de commande, D : Mot de données et Statut : mot de statut

FIGURE 1.2: Relation entre format de transfert et type de message

Comme la communication est à la seule initiative du gérant de bus, tout transfert débute alors par un mot de commande. C'est le contenu de celui-ci qui détermine l'ordre des mots statut et de données échangés. Le format de ces messages est détaillé ci-après.

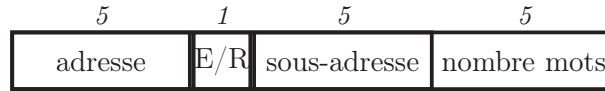
1.2.3 Format des messages MIL-STD-1553B pour le lanceur

Le format des messages MIL-STD-1553B utilisés dans le lanceur Ariane 5 est représenté à la FIGURE 1.3 (hors bits de synchronisation et de parité). Le mot de commande contient quatre informations différentes : l'adresse de l'abonné (5 *bits*), un bit pour spécifier si l'abonné doit émettre ou recevoir des données (bit E/R), la sous-adresse de l'abonné (5 *bits*) identifiant un bloc de données et le nombre de mots de données à échanger (5 *bits*).

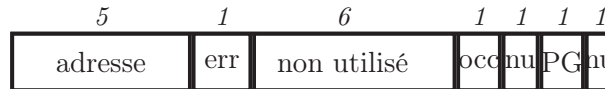
Le mot de statut est composé de l'adresse physique (sur 5 *bits*) et de 3 bits de fonctions diverses (erreur de message, occupé et prise de gestion). Il permet de connaître l'état des données reçues (bit « erreur »), l'état de l'abonné (bit « occupé » à "1" lorsqu'il n'est pas prêt à recevoir ou à émettre) et le changement de mode d'abonné à gérant de bus (quand un gérant de bus est défaillant par exemple). Notons que les 8 *bits* restants ne sont pas utilisés sur les lanceurs.

Le(s) mot(s) de données ne font pas l'objet d'encapsulation protocolaire, ce qui permet le transport de 16 *bits* de données par mot.

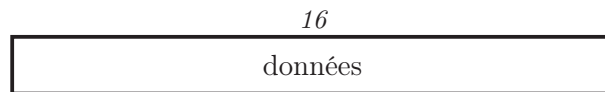
Mot de commande :



Mot de statut :



Mot de données :



err : erreur de message, nu : non utilisé, occ : occupé, PG : Prise de gestion

FIGURE 1.3: Mots MIL-STD-1553 B (*la longueur des champs est donnée en bits*)

1.2.4 Organisation temporelle des transferts sur les lanceurs

Les différents transferts sont organisés en série. Une série est un enchaînement successif de transferts dans lequel est contenu au moins un transfert, et un transfert peut se retrouver plusieurs fois dans la série. La notion de trame s'entend pour l'enchaînement automatique de séries.

Cette trame sera alors réalisée périodiquement par le gérant de bus (cycle de communication). La définition des différentes séries permet d'obtenir les périodes désirées pour chacun des flux (inférieures à la période du cycle) selon sa répétition dans la série (macro-cycle ou cycle mineur) et de la répétition de la série dans la trame. Ces différentes séquences temporelles sont choisies scrupuleusement de manière à respecter les contraintes temporelles des différents transferts.

1.2.5 Commentaires

Le mode de communication maître/esclaves et la définition d'un cycle de communication périodique confère au bus MIL-STD-1553B un fonctionnement entièrement déterministe. De plus, il se caractérise par un comportement synchrone entre l'application et les couches basses de communication. Dans ce mode de communication, le contrôleur de bus (qui est également l'ordinateur de bord) prend seul l'ensemble des décisions : l'architecture de commande est donc dans l'application lanceur, centralisée.

Il est important de noter ici que la qualification est facilitée par le comportement déterministe du système de communication. La philosophie actuelle requiert que tout comportement (temporel, reconfiguration) soit entièrement connu *a priori*. A tout moment, il est possible de connaître

l'état du réseau et de l'application (grâce à la synchronisation communication/application et de l'organisation temporelle définie).

Pourtant, nous allons voir dans la section suivante que le bus MIL-STD-1553B n'est plus adapté.

1.3 Des nouveaux besoins du spatial aux réseaux Ethernet

1.3.1 Des nouveaux besoins du spatial ...

Les systèmes spatiaux européens (lanceurs, satellites, ...) se sont longtemps appuyés sur le standard MIL-STD-1553B comme technologie pour leur système de communication. Les différents équipements constituant ces systèmes doivent être capables de s'échanger deux types de trafics : du trafic de commande et du trafic de métrologie (scientifique dans le cas des satellites notamment). Ces deux types de trafics sont transportés par deux réseaux séparés : alors que le trafic de commande est supporté par le bus MIL-STD-1553B, le trafic de métrologie est supporté par un autre réseau (soit des liaisons point-à-point à haut débit pour les satellites ou un réseau propriétaire pour les lanceurs).

Dans une problématique de réduction des coûts et de la masse du système global, les deux réseaux pourraient être unifiés sur un même système de communication. Dans ce cas de figure, le système de communication doit être capable de supporter des trafics plus nombreux voire plus volumineux. Par conséquent, le bus MIL-STD-1553B constitue une limitation notable en terme de bande passante (1 Mb/s).

Ainsi, pour pallier à cette limitation, l'ESA (*European Space Agency*) a conçu le standard SpaceWire pour les futurs satellites offrant une bande passante jusqu'à 200 Mb/s. Ce standard s'appuie sur le routage *whormole* qui consiste à retransmettre un paquet à la connaissance de son seul premier caractère (au contraire d'un mode *Store&Forward* où il serait nécessaire de mémoriser l'ensemble du paquet). Lorsqu'un routeur émet un paquet, il ne peut transmettre un autre paquet ce qui a pour conséquence la congestion des routeurs amonts (on parle alors de blocage). Bien qu'une version temps-réel de SpaceWire est à l'étude (s'appuyant sur la notion de classe de trafic), (Ferrandiz, 2012) s'est attaché à proposer deux méthodes d'analyse de calcul de délais de la version initiale du standard : l'une s'appuie sur le calcul récursif du délai pire-cas et la seconde utilise la théorie du *network calculus*. Ces méthodes permettent la qualification d'un réseau SpaceWire.

En revanche, dans le cas des lanceurs, la volonté de s'appuyer sur une technologie permettant une utilisation de composants matériels et d'outils de développements existants à moindre coût (COTS), a suscité un intérêt certain pour le standard Ethernet.

1.3.2 ... aux réseaux Ethernet

Lors de la dernière décennie, le monde industriel a été soumis à la même problématique lors de l'évolution des applications de contrôle / commande. Les Réseaux Locaux Industriels (RLI), jusqu'alors utilisés, n'étaient plus adaptés aux nouveaux besoins. Alors que le standard Ethernet (développé depuis les années 80) est bien implanté dans les réseaux informatiques, il devient un réseau attractif de plus en plus utilisé pour les solutions d'interconnexion des équipements de terrain dans l'automatique industrielle (Decotignie, 2001). Bien que selon Decotignie, aucune réponse ne peut être de prime abord apportée de façon franche et définitive, il présente sept bonnes raisons d'utiliser Ethernet dans le monde industriel : la disponibilité d'un grand nombre de circuits électroniques de bon marché, la simplicité d'intégration avec Internet qui s'appuie sur la même technologie, le bénéfice des protocoles supportés par TCP (FTP, HTTP, ...), une perpétuelle évolution de la bande passante, l'universalité et la compatibilité (les multiples réseaux de terrain seront alors remplacés par une solution unique), l'incapacité des réseaux traditionnels (même dans des versions plus évoluées) à répondre aux nouveaux besoins (notamment pour les lanceurs comme évoqué précédemment) et l'évolution croissante du marché de la technologie Ethernet.

Cependant, la méthode d'accès au médium s'appuie sur un algorithme CSMA/CD non déterministe (IEEE Computer Society, 2002b), qui utilise une méthode stochastique pour résoudre les collisions et qui ne garantit pas la transmission du message. Par conséquent, Ethernet ne peut pas être *a priori* implémenté dans un milieu industriel à forte contrainte temporelle. Afin de faire face à ce problème, différentes solutions ont été proposées comme mentionnées dans (Alves *et al.*, 2000; Decotignie, 2001) et (Decotignie, 2005). L'avènement des commutateurs (IEEE Computer Society, 1998) associés à des liens *full-duplex* permet alors d'éliminer l'indéterminisme : chaque équipement est relié à un commutateur rendant impossible toute collision (*full-segmentation*). Il viendra ensuite des évolutions telles que les VLANs et la classification de service (IEEE Computer Society, 2003) permettant de différencier les trafics en fonction de leurs exigences temporelles. Fort de ces évolutions, chaque fournisseur fait la promotion de son produit avec pour but la standardisation de celui-ci (Felser et Sauter, 2004). A présent, le standard IEC 61784 (IEC, 2003b) présente une multitude de solutions non-interopérables basées sur Ethernet (EtherCAT, Profinet, ...), et est intégré au standard IEC 61158 (IEC, 2003a).

Dans le milieu industriel, pour pouvoir valider une solution, il est nécessaire de s'assurer en premier lieu que les contraintes temporelles soient respectées. Ainsi, depuis une dizaine d'années, des travaux portent sur l'analyse de performances de l'utilisation d'un réseau Ethernet commuté. On peut notamment citer, dans le cas de réseaux dédiés, des travaux sur l'évaluation de solutions basées sur le standard IEC 61784 (IEC, 2003b). (Jasperneite et Elsayed, 2004; Jasperneite *et al.*, 2007; Prytz, 2008; Seno et Zunino, 2008) conduisent des recherches sur les solutions Profinet IRT

et EtherCAT, alors que (Pereira *et al.*, 2004a; Alessandria *et al.*, 2007) travaillent sur Ethernet/IP de manière analytique et expérimentale.

Ces travaux s'appuient sur des scénarios particuliers de communication (souvent propres à la technologie) et restrictifs puisque le réseau est dédié. Or, le réseau est de plus en plus partagé avec des trafics non fonctionnels (la télémétrie par exemple). Dans l'optique de l'utilisation d'Ethernet commuté dans les milieux nécessitant des garanties temporelles (industrie, aéronautique et spatial), d'autres méthodes d'évaluation de performances sont alors requises. Bien qu'il existe des méthodes stochastiques (Song, 2001; Ridouard *et al.*, 2007; Brahimi, 2007), des analyses déterministes sont souvent préférées. Plus particulièrement, la conception des systèmes embarqués nécessitent le respect de contraintes temporelles. Dans ce cadre, de nombreux travaux s'appuient sur la théorie du calcul réseau (ou *network calculus*) qui va permettre de calculer des bornes sur les délais de bout en bout. Cette théorie initiée par (Cruz, 1991), (Chang, 2000) et (Le Boudec et Thiran, 2001) et issue de la théorie Min-Plus est utilisable pour les systèmes temps réel. (Jasperneite *et al.*, 2002) utilisent cette théorie pour analyser l'efficacité de topologies linéaires et en étoile sur différents scénarios. Des simulations sont utilisées pour confirmer l'applicabilité des résultats analytiques sur ces scénarios. Plusieurs modélisations de la commutation ont été proposées et appliquées aux Systèmes Contrôlés en Réseau (SCR) (Georges, 2005)(Diouri, 2010). Au delà des systèmes industriels, la théorie du calcul réseau a été employée dans l'aéronautique, avec la certification du réseau AFDX embarqué dans l'A380 (Grieu, 2004)(Charara, 2007).

A l'instar des milieux industriel et aéronautique qui se sont orientés vers la technologie Ethernet pour leur système de communication, il se pose la question de savoir si cette même orientation technologique peut être cohérente dans le domaine spatial. Ainsi, une première étude d'analyse de performances de solutions industrielles basées sur Ethernet est présentée dans la section suivante de manière à mettre en avant les capacités temporelles de solutions Ethernet en vue d'une étude approfondie au domaine des lanceurs.

1.4 Analyse de solutions temps-réel basées sur Ethernet (Robert *et al.*, 2010) (Robert *et al.*, 2012)

1.4.1 Introduction

Aujourd'hui, différentes solutions basées sur Ethernet existent sur le marché (regroupées dans le standard IEC 61784 (IEC, 2003b)). Ces différents protocoles peuvent être classés dans deux catégories. La classification se base principalement sur les fonctionnalités ajoutées à Ethernet standard. Une première catégorie consiste à utiliser l'Ethernet standard comme couches basses (par exemple, Modbus/TCP ou Ethernet/IP) avec un protocole applicatif particulier. La seconde catégorie consiste à ajouter un intergiciel supplémentaire d'accès au médium (par exemple, Ether-

CAT ou Profinet IRT avec modification de l'électronique). L'objectif de cette partie est d'évaluer les performances de solutions basées sur Ethernet s'inscrivant dans l'une ou l'autre des catégories. Ainsi, les solutions utilisées dans les milieux industriels et retenues dans cette étude, de par leur représentativité dans l'industrie et de l'utilisation d'Ethernet, sont pour cette étude : EtherCAT, Profinet IRT, Modbus/TCP et Ethernet/IP.

Afin de comparer les performances obtenues, un scénario commun est défini : un unique calculateur agissant comme un maître collecte sur un réseau dédié (sans flux concurrents) les différentes mesures et met à jour périodiquement les données. Les capteurs et actionneurs sont quant à eux considérés comme des esclaves. Le temps de cycle peut être divisé en trois parties comme montré dans la FIGURE 1.4 : collecter les données (c'est à dire les données produites par les capteurs), calculer les nouvelles données et finalement les transmettre sur le réseau.

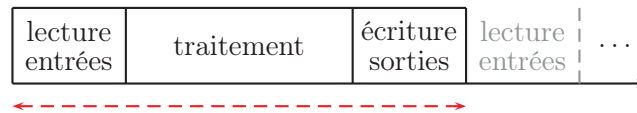


FIGURE 1.4: Temps de cycle automate

Le réseau doit être capable de supporter pendant un cycle automate les échanges relatifs aux entrées envoyées par les capteurs au contrôleur et les sorties envoyées par le contrôleur aux actionneurs. De ce point de vue, le temps de cycle doit être plus long que le temps nécessaire à tous ces échanges. Un temps de cycle minimum correspond alors au temps de communication nécessaire pour collecter et mettre à jour les mémoires de données de tous les capteurs et actionneurs.

Le temps de cycle minimum sera utilisé dans ce chapitre comme la métrique permettant de comparer la capacité des différentes solutions basées sur Ethernet à satisfaire les contraintes industrielles. Le temps de cycle minimal est calculé pour chaque protocole et exprimé en fonction de différents paramètres tels que le temps de transmission d'une trame, la latence d'un équipement réseau et le temps de propagation d'un lien. Le tableau 1.1 décrit les notations utilisées dans ce chapitre.

1.4.2 EtherCAT

EtherCAT est développé par Beckhoff (type 12 dans le standard IEC 61158, (IEC, 2006a; IEC, 2006b)). En théorie, EtherCAT utilise des cartes Ethernet standard. Cependant en pratique, un matériel spécifique (FPGA : *Field-Programmable Gate Array* ou ASIC : *Application-Specific Integrated Circuit*) est utilisé pour minimiser le temps de traversée de la trame. EtherCAT ajoute un protocole maître/esclaves au dessus d'Ethernet. La trame est envoyée par le maître et les esclaves peuvent lire et écrire à la volée (mécanisme *on the fly*). Le temps de lecture et d'écriture

TABLE 1.1: Notations

Termes	Notation	Unités
Temps de transmission	τ	s
Latence de l'équipement	ℓ	s
Temps de propagation	δ	s
Capacité du lien	C	b/s
Charge utile	x	<i>octets</i>
Nombre d'équipements (esclaves)	n	–

est indépendant de la taille de la trame et est identique pour tous les esclaves : ce temps correspond à la latence ℓ introduite par l'équipement. Un anneau logique est défini entre les esclaves tel que lorsque la trame atteint le dernier esclave (de l'anneau), elle est renvoyée au maître. Le diagramme spatio-temporel de la FIGURE 1.5 illustre le comportement des communications EtherCAT.

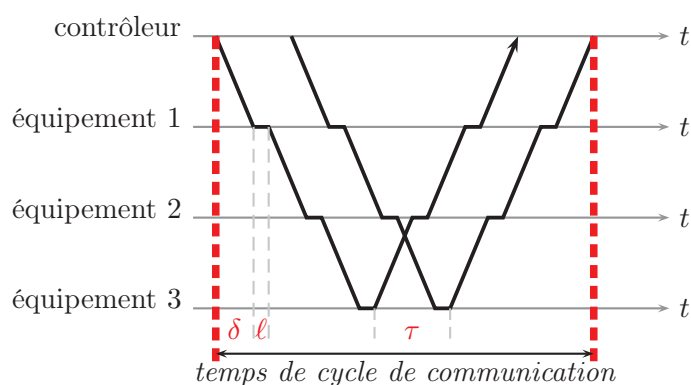


FIGURE 1.5: Diagramme spatio-temporel d'EtherCAT

Le protocole EtherCAT supporte des topologies linéaires et en anneau. Comme la topologie linéaire (FIGURE 1.6) est principalement utilisée en industrie, elle sera étudiée ici.

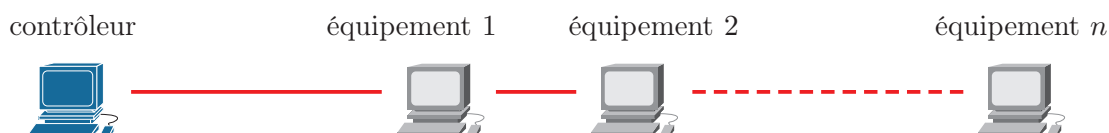
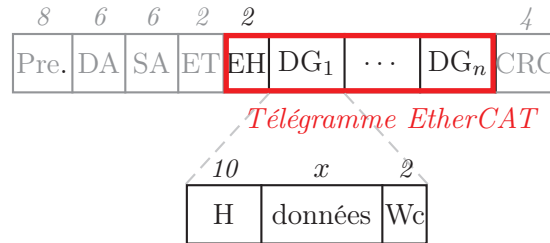


FIGURE 1.6: Topologie linéaire d'EtherCAT

Les datagrammes EtherCAT sont directement encapsulés dans la trame Ethernet comme

montré dans la FIGURE 1.7. Une trame EtherCAT est composée d'une entête spécifiant la longueur de la trame et de l'ensemble des datagrammes. Un datagramme est défini pour chaque esclave et contient le type de la commande ainsi que les données associées. D'ores et déjà, il est possible de dire que le nombre d'esclaves aura un impact important sur la taille de la trame.



Pre : Preamble, SA/DA : Source/Destination Address, ET : Ethershape
 EH : EtherCAT Header, DG : Datagram, H : Header, Wc : Working counter

FIGURE 1.7: Trame EtherCAT (la longueur des champs est donnée en octets)

L'analyse des performances d'EtherCAT considère ici les hypothèses suivantes :

- topologie linéaire,
- l'étape d'initialisation est ignorée et seules les communications cycliques sont prises en considération,
- le maître envoie une seule trame par cycle,
- la charge utile x est identique pour chaque esclave.

Sous ces hypothèses, il est à présent possible d'évaluer la durée minimale du temps de cycle (FIGURE 1.5). Le temps de transmission est le rapport entre la taille de la trame et la capacité du lien C . La taille totale de la trame peut être divisée en deux parties :

- une partie constante égale à la somme du protocole Ethernet (26 octets), le trou inter-trame (12 octets) et l'entête EtherCAT (2 octets),
- une partie variable dépendant du nombre d'esclaves n , de la quantité de données à transporter x et de l'entête (12 octets).

Le temps de transmission est alors :

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

Le terme 44 dans l'équation précédente est ajouté dans le but de prendre en compte la taille minimale des données définie par le protocole Ethernet : si la longueur d'un télégramme EtherCAT est inférieur à 46 octets, une quantité de données de remplissage sera insérée dans la trame Ethernet. Comme un télégramme EtherCAT inclut une entête de 2 octets, il n'y aura donc pas de remplissage si la taille de la séquence de datagrammes est supérieur à 44 octets.

D'après la FIGURE 1.5, le temps de cycle de communication est la somme de :

- la latence pour traverser tous les esclaves (identique pour tous les esclaves)

- le temps de propagation. Le temps de propagation est supposé identique sur tous les liens.
- le temps de transmission (calculé précédemment).

L'estimation du temps de cycle peut alors être obtenue par l'expression suivante :

$$\begin{aligned}\Gamma &= (2n - 1)\ell + 2n\delta + \tau \\ &= (2n - 1)\ell + 2n\delta + \frac{8(40 + \max(44, n(12 + x)))}{C}\end{aligned}\quad (1.1)$$

Notons que (1.1) prend en compte seulement une trame. Comme la charge utile dépend directement du nombre d'esclaves et comme la trame Ethernet ne peut excéder 1526 *octets* (et donc un champs de données de 1500 *octets*), l'équation ne peut être utilisée que si le nombre d'équipements interconnectés au réseau est inférieur à :

$$n \leq n_{max} = \left\lfloor \frac{1500 - EH}{12 + x} \right\rfloor$$

où EH est la taille de l'entête EtherCAT (2 *octets*) et n_{max} le nombre de datagrammes de longueur x qui peut être inclut dans une seule trame (il est donc supposé ici que $x \leq 1486$ *bytes*).

Si le nombre d'équipements sur le réseau n'est pas limité à la valeur donnée précédemment, il est nécessaire de considérer l'envoi de plusieurs trames par le contrôleur. En fait, le nombre de trames Ethernet nécessaires pour transporter les données (de charge utile x) des n équipements est donné par l'expression suivante :

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

Par conséquent, l'équation (1.1) doit maintenant intégrer les différents temps de traversée de chaque trame (le remplissage n'étant considéré si besoin que pour la dernière trame). L'expression du temps de cycle minimal devient :

$$\begin{aligned}\Gamma &= (2n - 1)\ell + 2n\delta + \frac{8}{C} \left(40k + (k - 1)n_{max}(12 + x) \right) \\ &\quad + \frac{8}{C} \max \left(44, (n - (k - 1)n_{max})(12 + x) \right)\end{aligned}\quad (1.2)$$

Comparé aux résultats de (Jasperneite *et al.*, 2007), le temps de cycle minimal défini en (1.2) prend en compte le mécanisme *on the fly* proposé par EtherCAT. L'avantage principal de ce mécanisme est qu'un équipement peut commencer à transmettre une trame avant la réception complète de celle-ci (contrairement à un mode *store and forward*), ce qui permet de réduire considérablement le délai de bout-en-bout global (FIGURE 1.5). De plus, comparé à (Jasperneite *et al.*, 2007; Prytz, 2008), l'équation (1.2) prend en compte la possibilité de remplissage définie par Ethernet et le temps pour transmettre l'information envoyée par les équipements au contrôleur (FIGURE 1.5). Finalement, il est important de noter que les trames Ethernet ne seront complètes (1526 *octets*) que pour une configuration particulière de la charge utile x des équipements EtherCAT.

1.4.3 Profinet IRT (*Isochronous Real-Time*)

Le protocole Profinet est développé par Siemens (type 10 dans le standard IEC 61158, (IEC, 2006a; IEC, 2006b)). Différentes versions du protocole existent telles que RT (*Real-Time*) et IRT. Profinet IRT est capable de supporter les communications temps-réel. Des cartes Ethernet standard ne peuvent être utilisées et donc Profinet requièrent un matériel spécifique sur les esclaves (ASIC de type commutateurs 2 ou 4 ports intégrés). Le protocole est basé sur la division temporelle du cycle de communication en deux modes (FIGURE 1.8).

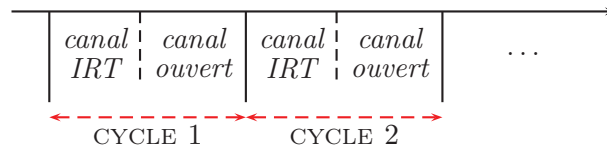


FIGURE 1.8: Découpage temporel de Profinet

Les modes asynchrones et isochrones sont utilisés respectivement pour des trafics non contraints (canal ouvert) et temps-réel (canal IRT). Seul le mode isochrone sera analysé ici. Le lecteur peut consulter (Jasperneite et Elsayed, 2004) pour une simulation de ces deux modes. Dans le mode isochrone, le maître envoie une trame de données à chaque équipement. Et chaque équipement fournit une réponse au maître. Profinet supporte des topologies linéaires, en étoile ou en anneau. La topologie linéaire (FIGURE 1.9) avec des liens *full-duplex* est sélectionnée dans cette étude pour les mêmes raisons que pour EtherCAT.

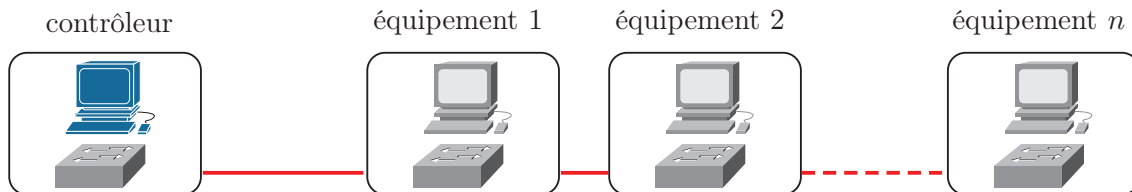
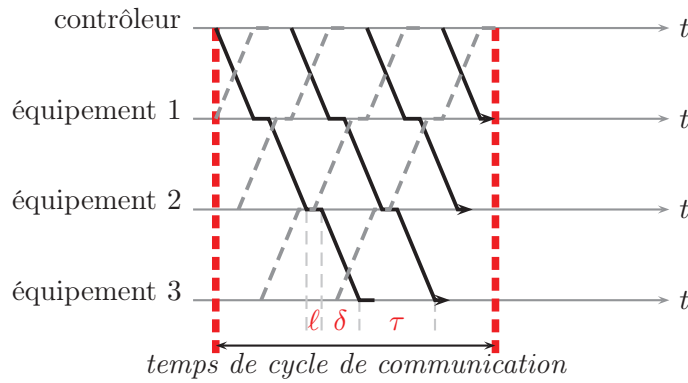


FIGURE 1.9: Topologie linéaire de Profinet

Ici, il est considéré que Profinet utilise le *slipstreaming effect* qui consiste en l'envoi par le contrôleur des trames dans un ordre particulier : de l'esclave le plus éloigné au plus proche sur la topologie linéaire. Un comportement similaire est utilisé pour le trafic généré par les esclaves au contrôleur puisqu'il est considéré des échanges simultanés et en mode *full-duplex*. Le schéma de communication est introduit dans la FIGURE 1.10.

Le diagramme spatio-temporel de la FIGURE 1.10 correspond au meilleur cas d'utilisation de Profinet IRT. Les équipements sont supposés synchronisés, et capables d'envoyer périodiquement leurs messages au contrôleur. De plus, le *slipstreaming effect* sera considéré seulement lorsqu'il est positif, et donc il est nécessaire que $\tau \geq \delta + \ell$ comme mentionné dans (Jasperneite *et al.*, 2007).

FIGURE 1.10: Diagramme spatio-temporel de Profinet IRT avec *slipstreaming effect*

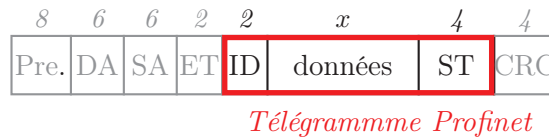
Dans ce cas, le temps de cycle est donné par (Jasperneite *et al.*, 2007; Prytz, 2008) comme étant la somme de :

- la latence l pour traverser tous les esclaves et le temps de propagation δ pour chaque lien
- et le temps de transmission τ de chaque trame envoyée par le contrôleur.

Le temps de cycle minimal s'exprime alors :

$$\Gamma = \delta + l + n\tau \quad (1.3)$$

Comparé au temps de cycle minimal pour EtherCAT (équation (1.2)), il est à noter que le temps de transmission est plus important dans Profinet puisqu'il est multiplié ici par le nombre d'esclave. L'équation (1.3) peut être développée en analysant ce temps de transmission τ . Les télégrammes Profinet sont encapsulés directement dans la trame Ethernet (FIGURE 1.11).



Pre : Preamble, SA/DA : Source/Destination Address, ET : Ethertype
ID : Identifier, ST : Status

FIGURE 1.11: Trame Profinet IRT (la longueur des champs est donnée en octets)

La FIGURE 1.11 montre que trois champs sont ajoutés à la trame Ethernet : un identifiant (2 octets), les données (sous l'hypothèse suivante $x \leq 1494$ octets) et un statut d'information (4 octets). Avec le respect de la taille minimale des données Ethernet, le temps de transmission d'une trame est donné par l'expression suivante :

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

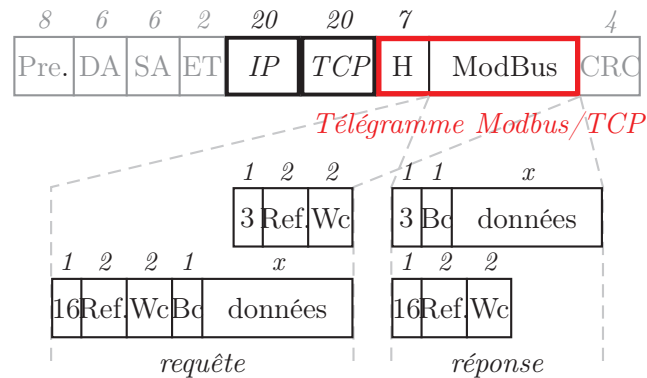
et l'équation (1.3) peut alors être complétée comme suit :

$$\Gamma = \delta + l + n \frac{8}{C} (38 + \max(46, 6 + x)) \quad (1.4)$$

L'équation (1.4) sera utilisée pour le calcul du temps de cycle minimum d'un réseau Profinet donné.

1.4.4 Modbus/TCP

Modbus est un protocole de communication série proposé par Modicon en 1979. Modbus/TCP (type 15 dans le standard IEC 61158, (IEC, 2006a; IEC, 2006b)) est une variante du protocole Modbus qui est adapté à l'utilisation d'Ethernet (couche physique et liaison de données (The Modbus Organization, n.d.; Modbus-IDA, 2006)). Modbus/TCP encapsule une trame Modbus dans une trame TCP (FIGURE 1.12). Le datagramme Modbus est composé d'une en-tête applicative du protocole Modbus (MBAP), de la fonction (lecture/écriture) et des données.



Pre : Preamble, SA/DA : Source/Destination Address, ET : Ethertype
 IP : Internet Protocol, TCP : Transport Control Protocol
 Ref. : Reference number, Wc : Word Counter, Bc : Byte Counter

FIGURE 1.12: Trame Modbus/TCP et trame des fonctions spécifiques à Modbus (la longueur des champs est donnée en octets)

Avant d'aller plus loin dans l'analyse, il est important de noter que les performances de Modbus/TCP dépendent de différents paramètres comme la topologie, les échanges applicatifs ou même les performances des processeurs. En effet, Modbus/TCP est une approche pragmatique dans laquelle toutes les configurations sont possibles. Bien que le protocole soit basé sur des transactions orientées connexion, Modbus/TCP peut utiliser un schéma de communication maître/esclaves, producteur/consommateur ou bien client/serveur. Le mode maître/esclaves est choisi dans cette étude afin d'introduire un autre principe d'accès au médium comparé à EtherCAT et Profinet. Une autre raison est que ce mode simplifie l'analyse car les équipements sont synchronisés par les événements réseau. La communication peut se décrire de la façon suivante : le maître envoie une trame par esclave, et à la réception de la trame, l'esclave envoie au maître une réponse applicative (couche 7). Le maître interroge un autre esclave dès la réception de la trame entière. Ce mécanisme de requête/réponse est montré sur la FIGURE 1.13.

Des échanges additionnels dus au protocole TCP ne sont pas considérés dans la FIGURE 1.13.

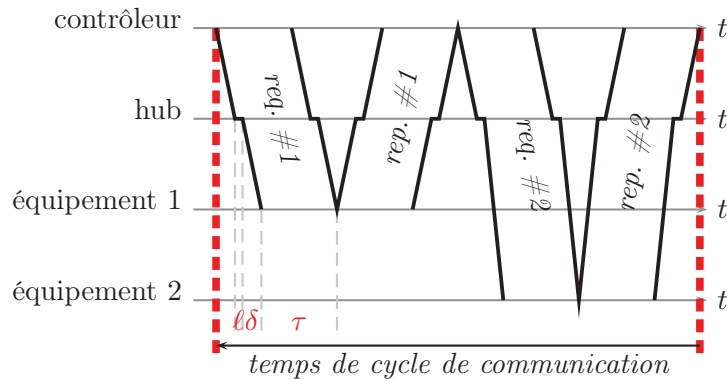


FIGURE 1.13: Diagramme spatio-temporel de Modbus/TCP

Comme Modbus/TCP repose sur le protocole TCP, son comportement doit être pris en compte. Premièrement, les sessions d'ouverture et de fermeture ne sont pas considérées, l'étude porte sur la phase périodique. Deuxièmement, TCP est responsable de l'acquittement des transmissions par segment. À la réception d'un segment, TCP décide d'envoyer immédiatement un acquittement, d'attendre la réception de plusieurs segments dans le but d'acquitter l'ensemble en une seule fois ou bien d'utiliser le "piggybacking" ce qui signifie ajouter un acquittement dans la prochaine transmission de données. En fait, le comportement de TCP diffère en pratique selon différents paramètres comme le système d'exploitation, les configurations TCP, ou l'activation de l'algorithme de Nagle.

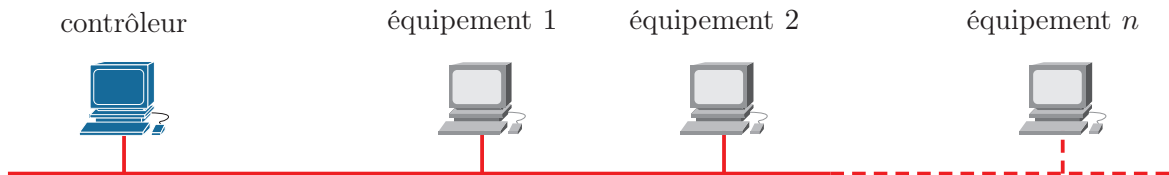


FIGURE 1.14: Topologie linéaire de Modbus/TCP

Modbus/TCP supporte différentes topologies (linéaire, étoile ou boucle). Des architectures complexes basées sur des commutateurs peuvent être utilisées spécifiquement quand le réseau est partagé avec d'autres applications. Cependant, comme le schéma de communication de type maître/esclaves est considéré et que l'utilisation de commutateurs implique des coûts additionnels, une infrastructure de type bus sera appliquée à notre étude (FIGURE 1.14).

Chaque équipement est supposé interconnecté à un concentrateur commun. Le délai d'un équipement à un concentrateur est égal à δ . Pour cette conception, la FIGURE 1.13 montre que le temps de cycle est égal au nombre d'équipements n multiplié par le temps nécessaire pour interroger un équipement. Ce temps d'interrogation dépend :

- du temps de transmission de la requête et de la réponse,

- et du temps de propagation de la requête et de la réponse.

Des précédentes hypothèses, le temps de cycle minimal peut s'écrire :

$$\Gamma = n(\tau_{req} + \tau_{rep} + 2(2\delta + \ell)) \quad (1.5)$$

Dans le pire cas, où les acquittements TCP sont envoyés pour chaque segment, le temps de cycle minimal sera égal à $\Gamma = n(\tau_{req} + \tau_{rep} + 3 \times (2\delta + \ell) + 2\tau_{ack})$ avec $\tau_{ack} = 672/C$ et 672 la taille minimale en bits d'une trame Ethernet (trou inter-trame inclus). Notons que l'équation (1.5) diffère quand plusieurs concentrateurs sont utilisés car les temps de propagation doivent être augmentés.

D'après la FIGURE 1.12, le temps de transmission des trames est alors composé de :

- une partie constante relative à la somme du protocole Ethernet (38 *octets* avec le trou inter-trame), l'en-tête IP (20 *octets* sans options), l'en-tête TCP (20 *octets* sans options) et l'en-tête Modbus/TCP (7 *octets*)
- une partie variable relative au type de message Modbus. En fait, le format du télégramme Modbus encapsulé dans la trame Modbus/TCP varie en fonction du type de données (code fonction) et du type de transaction (requête ou réponse). La FIGURE 1.12 montre le format correspondant à une lecture de registre (code fonction 03h) et à une écriture multiple de registre (code fonction 10h). Ici, les requêtes en écriture seront considérées. Chaque message Modbus/TCP encapsule un télégramme Modbus.
- et une partie variable proportionnelle à la charge utile (comme le champs "compteur d'octet" est sur un octet, le champs de donnée est limité à $x \leq 255$ *octets*).

Comme la somme des tailles des en-têtes IP, TCP et Modbus est largement supérieure à la taille minimale des trames Ethernet, il n'y a pas besoin d'ajouter du remplissage. Le délai est directement donné par :

$$\tau_{req_{10h}} + \tau_{rep_{10h}} = 8 \frac{91 + x}{C} + 8 \frac{90}{C}$$

tel que l'équation (1.5) correspond dans ce cas à :

$$\Gamma_{10h} = n \left(8 \frac{181 + x}{C} + 2(2\delta + \ell) \right) \quad (1.6)$$

Modbus/TCP s'appuie ici sur un mode de communication maître/esclaves où un seul paquet ne peut être présent à la fois sur le réseau (synchronisme de l'application et de la communication). Ainsi, bien qu'il s'appuie sur une pile TCP/IP standard, cette solution s'écarte de l'Ethernet standard (avec un possible asynchronisme des communications) vers lequel on tend. C'est pourquoi, il est présenté dans la section suivante la solution Ethernet/IP.

1.4.5 Ethernet/IP

Ethernet/IP (IP signifie *Industrial Protocol*) est un réseau développé par Rockwell Automation en 2001 et supporté par l'ODVA (*Open DeviceNet Vendor Association*) (Brooks, 2001; Schiff-

fer, 2006). Ethernet/IP (type 2 dans le standard IEC 61158,, (IEC, 2006a; IEC, 2006b)) utilise CIP (*Common Industrial Protocol*) ce qui permet d'utiliser des produits sur étagère compatibles avec la pile TCP-UDP/IP. Ethernet/IP est un réseau basé sur les connexions CIP qui définissent le paquet qui sera produit sur le réseau. Deux catégories de connexions sont définies : la messagerie explicite (ou *Explicit Messaging*) et la messagerie d'entrées/sorties (ou *I/O messaging* ou *Implicit messaging*). Tandis que la messagerie explicite est utilisée pour des communications génériques et multiples entre deux nœuds, la messagerie d'entrées/sorties est spécifique aux applications d'entrées/sorties. Quand une application est contrainte temporellement, la messagerie d'entrées/sorties est le mode préféré car il utilise UDP plutôt que TCP. De plus, CIP utilise le modèle de communications producteur/consommateur qui requiert des échanges en diffusion (ou *broadcast*) supportés par UDP. L'analyse suivante sera réalisée dans le cas particulier de connexions de type entrées/sorties.

Comme Ethernet/IP s'appuie sur des composants sur étagère (*COTS : Components On The Shelves*), aucune topologie particulière n'est requise : des architectures basées sur des commutateurs (ou *switches*) ou des concentrateurs (ou *hubs*) sont possibles. Les commutateurs sont principalement intéressants dans le sens où ils réduisent les domaines de collisions, ils sont également utiles pour la redondance et peuvent supporter des mécanismes de réseau virtuel (ou *VLAN*) et de classification de service (Modlovansky, 2002). Des architectures commutées sont recommandées pour des messages temps-critique (ou *implicit*). Cependant, différents types d'architectures Ethernet commutées sont possibles. Dans cette étude, une topologie linéaire commutée est choisie (FIGURE 1.15). Nous pouvons voir celle-ci comme une extension de la configuration expérimentale considérée dans (Alessandria *et al.*, 2007) où le nombre de commutateurs varie ici en fonction du nombre de ports par commutateurs.

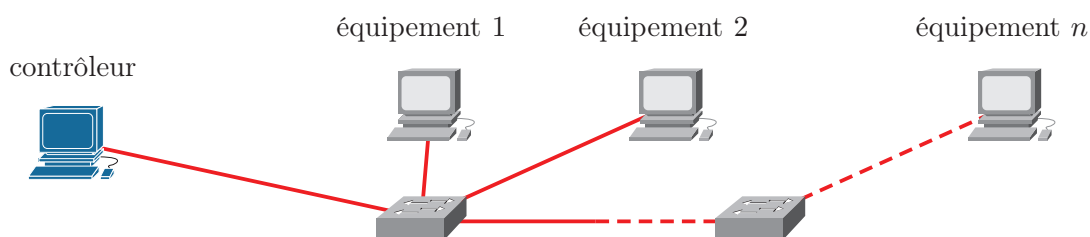


FIGURE 1.15: Topologie linéaire commutée d'EtherNet/IP

Dans le but de garantir la comparaison avec les protocoles précédemment étudiés, il est considéré que le contrôleur envoie une trame à chaque équipement et que respectivement chaque équipement produit une information pour le contrôleur. La FIGURE 1.16 représente ce comportement particulier.

Initialement, EtherNet/IP ne supporte pas de synchronisation d'accès au médium comme il y a dans une approche de type maître/esclaves utilisé dans Modbus/TCP. Cela signifie que tout

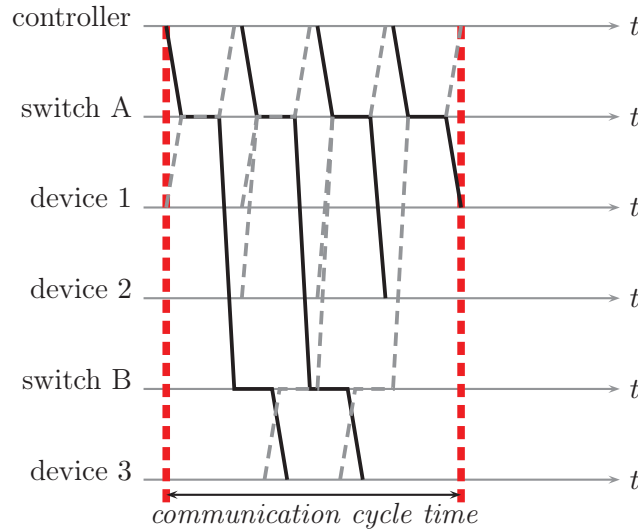


FIGURE 1.16: EtherNet/IP space-time diagram for 4-ports switches

équipement peut accéder au réseau en tout temps. Cependant, un mécanisme de synchronisation peut être utilisé : EtherNet/IP supporte le protocole IEEE 1588 implémenté dans le profil CIP-Sync. Par conséquent, il est tout à fait possible par exemple de demander aux esclaves d'envoyer leur message en même temps. Il est alors proposé ici d'utiliser comme pour Profinet le *slipstreaming effect* pour le contrôleur en plus du départ commun pour les équipements comme représenté sur la FIGURE 1.16. Bien entendu, ce profil correspond à un cas idéal relatif au temps de cycle minimum pour cette architecture. Premièrement, les erreurs de synchronisation mèneront à un autre profil et à un temps de cycle plus long. Deuxièmement, l'utilisation du *slipstreaming effect* requiert ici comme pour Profinet que $\tau \geq \delta + \ell$.

Considérons maintenant le profil donné par la FIGURE 1.16, le temps de cycle minimum correspond à la somme :

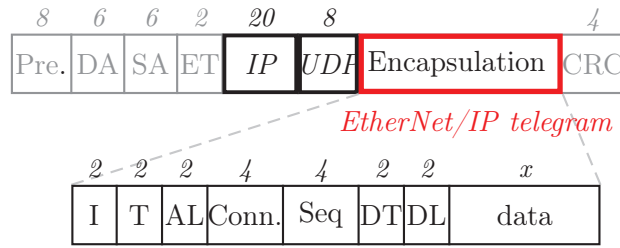
- des latences ℓ pour traverser seulement un commutateur plus deux fois le délai de propagation δ entre le contrôleur/équipement et un commutateur,
- et le temps de transmission τ de chaque trame envoyée par le contrôleur.

Le temps de cycle minimum peut alors être écrit de la façon suivante :

$$\Gamma = 2\delta + \ell + n\tau \quad (1.7)$$

Notons ici que le terme donnée dans l'équation 1.7 est seulement valide si $\tau \geq \delta + \ell$, ce qui signifie déjà que le commutateur ne travaille pas en mode *store and forward*. Le temps de transmission est alors calculé *via* le format d'encapsulation des messages *I/O* d'EtherNet/IP (FIGURE 1.17).

La FIGURE 1.17 montre que 18 *octets* sont ajoutés par le protocole CIP, 8 *octets* par UDP, 20 *octets* par IP et finalement 38 *octets* par Ethernet (pas de remplissage ici). Sous l'hypothèse



Pre : Preamble, SA/DA : Source/Destination Address, ET : Ethertype

IP : Internet Protocol, UDP : Uniform Data Protocol

I : Item count, T : Type ID, AL : Address Length, Conn. : Connection ID

Seq. : Sequence number, DT : Data Type ID, DL : Data Length

FIGURE 1.17: Trame des messages I/O d'EtherNet/IP (la longueur des champs est donnée en octets)

$x \leq 1454$ octets, le temps de transmission est donné par :

$$\tau = 8 \frac{38 + 20 + 8 + 18 + x}{C}$$

(1.7) peut être modifiée ainsi :

$$\Gamma = 2\delta + \ell + 8n \frac{84 + x}{C} \quad (1.8)$$

Comme le protocole CIP permet d'encapsuler plusieurs items dans une trame, il est à noter ici que le profil donné dans la FIGURE 1.16 peut différer avec l'application d'un scénario particulier. Dans la section suivante, les résultats obtenus pour EtherCAT, Profinet IRT, Modbus/TCP et EtherNet/IP seront comparés relativement au scénario commun et à la bande passante.

1.4.6 Comparaisons des solutions

L'objectif de cette partie est d'analyser le comportement des différentes solutions basées sur Ethernet dans le but des les comparer entre-elles.

Le tableau 1.2 résume les différentes expressions du temps de cycle minimum établies pour chacune des solutions présentées.

TABLE 1.2: Expression du temps de cycle minimum

Protocole	Temps de cycle minimum (Γ)
EtherCat	$(2n - 1)\ell + 2n\delta + \frac{8}{C} \left(40k + (k - 1)n_{max}(12 + x) \right) + \frac{8}{C} \max \left(44, (n - (k - 1)n_{max})(12 + x) \right)$
Profinet IRT	$\delta + \ell + n \frac{8}{C} \left(38 + \max(46, 6 + x) \right)$
Modbus/TCP	$n \left(8 \frac{181 + x}{C} + 2(2\delta + \ell) \right)$
EtherNet/IP	$2\delta + \ell + 8n \frac{84 + x}{C}$

Ces équations permettent d’ores-et-déjà d’observer que des protocoles ont la capacité de masquer les temps de propagation (c’est le cas de Profinet IRT et Ethernet/IP). Notons également que le bénéfice de l’augmentation de la bande passante sera moins sensible pour EtherCAT que pour les autres solutions car EtherCAT utilise k trames au lieu de n ($k \leq n$).

Une application numérique de ces expressions est réalisée pour deux bande-passantes (100 Mb/s et 1 Gb/s) mais également pour deux charges utiles introduites dans (Prytz, 2008) (16 *octets* dans le premier cas correspondant à une valeur acceptable dans les communications industrielles et 100 *octets*). La latence des équipements ℓ comme définie dans (Jasperneite *et al.*, 2007; Prytz, 2008) est résumée dans la TABLE 1.3. Les résultats des FIGURE 1.18 et FIGURE 1.19 expriment le temps de cycle en fonction du nombre d’esclaves connectés au réseau.

TABLE 1.3: Valeurs numériques

Protocole	<i>FastEthernet</i> (100 Mb/s)	<i>GigaEthernet</i> (1 Gb/s)
EtherCAT	1.35 μs	0.85 μs
Profinet IRT	3 μs	0.6 μs
Modbus/TCP	1 μs (<i>hub</i>)	
EtherNet/IP	3 μs	0.6 μs

Il est supposé que les performances des commutateurs utilisés dans EtherNet/IP sont les mêmes que Profinet IRT. Bien entendu, d’autres valeurs donneraient des autres temps de cycle (à noter également que ces résultats ne sont valides que si $\tau \geq \delta + \ell$). Le délai de propagation est de 50 ns (pour une distance de 10 m entre les équipements). La FIGURE 1.18 montre l’évolution du temps de cycle pour des petites charges utiles.

Premièrement, il peut être noté à partir de la FIGURE 1.18 que Modbus/TCP fournit de mauvais résultats : cela est dû à son mécanisme d’accès au médium. Le *polling* n’apparaît pas ici intéressant quand le temps de cycle minimum est considéré. Ce résultat est à mettre en relation avec les performances d’un réseau MIL-STD-1553B qui partage avec Modbus/TCP, un mécanisme d’accès au médium où une seule trame est sur le réseau à un instant donné. Deuxièmement, la figure met en avant le fait que si EtherCAT fournit le meilleur résultat quand *FastEthernet* est utilisé, il ne bénéficie pas de l’avantage de l’augmentation de la bande passante comme pour Profinet IRT ou EtherNet/IP. En effet, la bande passante a seulement un impact sur le temps de transmission τ . Comme une seule trame est envoyée en EtherCAT et comme Profinet et EtherNet/IP requièrent la transmission de n trames, le bénéfice de l’augmentation de la bande passante est plus sensible pour ces deux derniers protocoles. De plus, $2n$ latences seront considérées en EtherCAT alors que les temps de cycle en Profinet IRT et EtherNet/IP sont seulement composés d’une latence ℓ grâce au *slipstreaming effect*. C’est pourquoi Profinet et EtherNet/IP permettent des temps de cycle moins longs, et comme la différence d’un temps

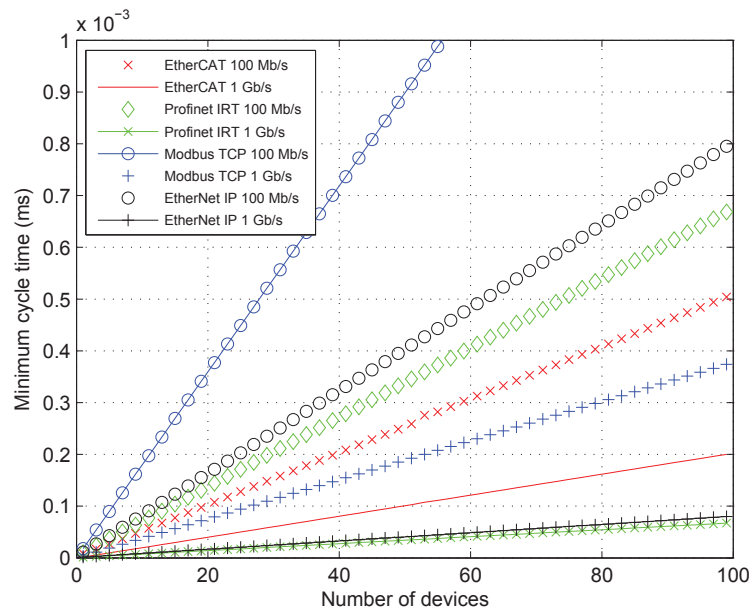


FIGURE 1.18: Temps de cycle minimum en fonction du nombre d'équipements pour une charge utile constante de 16 octets par équipement pour deux débits de 100 Mb/s et 1 Gb/s

de propagation existe entre les temps de cycle de Profinet IRT et EtherNet/IP, Profinet fournit de meilleurs résultats à 1 Gb/s.

La FIGURE 1.19 montre que les performances sont principalement les mêmes quand la charge utile augmente. La seule différence est relative à EtherCAT puisque son temps de cycle minimum sera plus long que Profinet quand le nombre d'équipements augmente. Cela est dû au fait que dans un tel cas, il y a de la fragmentation pour les télégrammes EtherCAT dans plusieurs trames Ethernet ce qui diminue les performances du protocole.

Cette première étude permet de mettre en avant qu'une solution Ethernet standard comme Ethernet/IP peut être compétitive face à une solution conçue pour le temps-réel (EtherCAT ou Profinet IRT). On note néanmoins que cette étude a montré qu'un réseau Ethernet standard comme Modbus/TCP fonctionnant en mode maître/esclaves (solution la plus proche du MIL-STD-1553B existant dans Ariane 5) présente de moins bonnes performances.

Cette étude est restreinte à un scénario commun et aux architectures définies. Toutefois, elle permet de dégager une tendance des performances obtenues d'une part par des solutions modifiant les couches basses ou utilisant des mécanismes de couches hautes d'autre part.

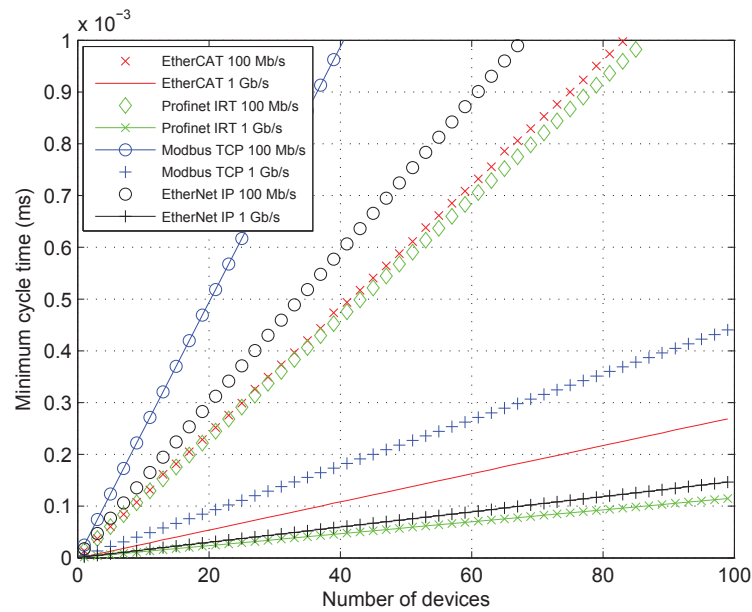


FIGURE 1.19: Temps de cycle minimum en fonction du nombre d'équipements pour une charge utile constante de 100 octets par équipement pour deux débits de 100 Mb/s et 1 Gb/s

1.5 Discussions

La volonté de s'appuyer sur une technologie permettant une utilisation de composants matériels et d'outils de développements existants à moindre coût (COTS), a suscité un intérêt certain pour le standard Ethernet. Les différents travaux de la littérature menés avec pour objectif de montrer les capacités du standard Ethernet à répondre à des contraintes temps-réel dans des applications variées (industrie et aéronautique) ont permis de conforter le choix d'Ethernet. Seulement, un nombre important de solutions temps-réel reposent sur ce standard Ethernet. Il est alors question de connaître quelle solution pourrait être adaptée à un scénario cyclique (relativement à ce qu'il peut être fait en industrie ou dans le domaine spatial) où un unique calculateur collecte les différentes mesures des différents capteurs et met à jour périodiquement les données des actionneurs. La métrique permettant d'évaluer le temps de communication nécessaire à la réalisation de ce scénario est alors défini comme le temps de cycle minimum. Pour réaliser la comparaison de leurs performances temps-réel, chacune des solutions est placée dans des conditions avantageuses relativement à la contrainte de temps de cycle minimum.

La première étape de cette analyse consiste à identifier les topologies qui minimisent la métrique du temps de cycle en considérant les mécanismes intrinsèques à chacune des solutions.

EtherCAT s'appuie sur la définition d'un anneau logique entre les esclaves. Cette définition permet l'utilisation de topologies linéaires, en anneau ou en étoile. Pour minimiser le temps de

traversée de la trame dans les équipements (lecture/écriture à la volée), ceux-ci s'appuient principalement sur un matériel spécifique à deux ports. Dans cette configuration, seule la topologie linéaire est alors possible. Néanmoins, des topologies en étoile sont envisageables grâce à l'utilisation d'équipements quatre ports ou de commutateurs standard mais leur temps de traversé est plus important (Prytz, 2008). Ainsi, la topologie linéaire (la plus couramment utilisé) permettant de minimiser la métrique est alors retenue dans cette analyse de performances.

Profinet IRT permet le support de communications temps-réel grâce à l'utilisation d'un matériel spécifique à 2 ou 4 ports. Cette caractéristique permet de supporter des topologies linéaires, en étoile ou en anneau. Pour minimiser le temps de cycle, un ordonnancement spécifique des trames est réalisé (*slipstreaming effect*). Ce mécanisme est d'autant plus bénéfique dans une topologie linéaire (Jasperneite *et al.*, 2007; Prytz, 2008). Ceci est principalement dû à la difficulté d'ordonner les messages sur une topologie plus complexe qui engendre un temps de cycle plus important. Par conséquent, nous retiendrons ici la topologie linéaire.

Dans le cas de Modbus/TCP, toutes les topologies (du bus à des architectures commutées) sont possibles. Le *pooling* alors considéré met alors en jeu qu'un message sur le réseau à un instant donné. Par conséquent, l'utilisation de commutateurs n'est pas nécessaire pour éviter les collisions et impliquerait une augmentation du temps de cycle dû au temps de traversée. De manière à minimiser la métrique, une infrastructure de bus sera utilisée dans notre étude.

Comme Ethernet/IP s'appuie sur des composants sur étagère compatibles avec la pile TCP-UDP/IP, aucune topologie particulière n'est requise. Néanmoins, Ethernet/IP s'appuie sur un modèle de communications producteurs/consommateur pouvant mettre en jeu à un instant donné plusieurs messages simultanés. Ainsi, une architecture commutée *full-duplex* permet d'éliminer les collisions. De manière à minimiser la métrique, nous proposons d'utiliser le *slipstreaming effect*. Comme il a été montré dans (Jasperneite *et al.*, 2007; Prytz, 2008), ce mécanisme est bénéfique pour des architectures linéaires. Par conséquent, la topologie commutée linéaire est choisie.

L'utilisation de deux débits différents permet de montrer les capacités des solutions à profiter (ou non) d'une augmentation de la bande passante. Ce n'est donc pas tant les valeurs numériques qui sont intéressantes ici (même si 100 Mb/s et 1Gb/s sont les débits les plus utilisés) mais bel et bien les conclusions associées à l'évolution du débit. Pour EtherCAT, le temps de transmission n'est qu'un paramètre additif dans l'expression du temps de cycle minimum. L'augmentation du débit ne permet pas une diminution significatif du temps de cycle minimum. En revanche, ce temps de transmission est multiplié par le nombre d'esclaves dans l'expression du temps de cycle minimum des solutions Profinet IRT, Modbus/TCP et Ethernet/IP. Ainsi, la diminution du temps de cycle sera d'autant plus importante que le nombre d'esclaves est conséquent lorsque le débit augmente.

Dans la mesure où il pourra être envisagé de transporter des trafics volumineux (comme de la vidéo par exemple), deux charges utiles ont été utilisées de façon à montrer l'aptitude des solutions à supporter (ou non) des données plus importantes. Ainsi, comme EtherCAT s'appuie sur un mécanisme d'agrégation de données, l'utilisation de charges utiles importantes pourrâ nécessiter la fragmentation des données de tous les équipements dans plusieurs trames. Aussi, l'agrégation de données n'est bénéfique (par rapport aux autres solutions) tant qu'une seule trame est utilisée.

1.6 Conclusion

Dans l'optique de répondre à la réduction des coûts et de la masse du système de communication, un intérêt certain s'est porté sur Ethernet à l'instar du monde industriel. Cependant, il se pose la question de savoir si la technologie Ethernet est envisageable pour les lanceurs spatiaux.

Dans la mesure où le réseau Ethernet s'appuie sur un mécanisme d'accès à la voie indéterministe et n'offre par conséquent aucune garantie (notamment temporelle), une étude approfondie de ce type de réseau est nécessaire pour justifier l'opportunité de son utilisation dans le cadre d'une application spatiale. C'est pourquoi une première étude de solutions industrielles proposant des scénarios de communication particuliers (et souvent propres à la technologie) visant à conférer du déterminisme aux communications s'effectuant sur un réseau dédié, a d'abord été réalisée. Cette étude a permis de mettre en évidence qu'une solution standard telle que Modbus/TCP où les communications sont centralisées est moins performante et que le support de la distribution avec Ethernet/IP s'avère être compétitif face aux solutions conçues pour le temps-réel (EtherCAT ou Profinet IRT).

Ces deux solutions standards utilisent un mode de communication différent qui les rend déterministes : alors que Modbus/TCP s'appuie sur un mode maître/esclaves, Ethernet/IP utilise dans cette étude le *slipstreaming effect*. L'utilisation de ces modes de communication est bien supportée dans le cas d'une commande centralisée sur un réseau dédié. A terme, l'objectif est d'abandonner le mode maître/esclaves, qui est sous-optimal en terme de bande passante. D'autre part, le mécanisme *slipstreaming effect* a été utilisé de manière à minimiser les temps de cycle de communication sur un réseau dédié. Or, le système de communication des futurs lanceurs ne sera non plus dédié mais partagé (avec des flux concurrents) et comme la métrique du temps de cycle minimum n'est plus adaptée relativement à la contrainte de réactivité, ce mécanisme ne sera pas retenu. L'évaluation des délais de bout-en-bout est alors plus adéquate pour qualifier le respect de cette contrainte de réactivité. Cependant, la seule garantie sur le déterminisme sur un réseau dédié ne suffit pas pour les lanceurs. Le chapitre suivant présente alors les exigences du système de communication des lanceurs spatiaux ainsi qu'une étude des délais de bout-en-bout.

Chapitre 2

Un premier « essai en vol » d’Ethernet

2.1 Introduction

Le chapitre précédent a permis de valider la technologie Ethernet standard en vue d’une étude plus approfondie au cas des lanceurs. Nous avons vu que le système de communication actuel repose sur une topologie à diffusion (le bus MIL-STD-1553B). En ce qui concerne la commande, sa conception s’était reposée sur un paradigme de commande centralisée. Pour cela, le mode de communication était de type maître/esclaves de sorte que les échanges sur le réseau soient synchrones avec l’évolution de la commande. La topologie de type bus support à ce type de communication conduisait alors à ce qu’un seul paquet soit présent sur le réseau.

Cependant, nous proposons de passer d’une topologie à diffusion à une topologie commutée. Cette évolution de la topologie va permettre d’avoir plusieurs trames sur le réseau. Ainsi, de nouveaux paradigmes tels que la distribution de la commande vont pouvoir être supportés. Cette distribution de la commande sera d’autant plus évolutive que ce soit en termes de mission ou de lancement. Elle pourra également remettre en cause la synchronisation des échanges avec l’évolution de la commande de sorte que les exigences de performances attendues du réseau demandent à être reconsidérées.

Dans ce contexte, nous présentons dans la section suivante les exigences des réseaux embarqués dans les lanceurs spatiaux.

2.2 Les exigences des réseaux embarqués dans les lanceurs spatiaux

Cette section présente les exigences spatiales auxquelles devra répondre le système de communication en précisant les différences que peut entraîner le changement de topologie (à diffusion/segmentée) et d’architecture de commande (centralisée/distribuée). Les différentes exigences que nous présentons selon notre point de vue, sont notées ici « E. i » (avec i le numéro de l’exi-

gence). A ce stade, aucune relation d'importance n'est introduite. La question de leur pondération sera traitée au chapitre 6.

E. 1 - Déterminisme :

La philosophie actuelle repose sur un déterminisme du comportement du système de communication. En effet, la technologie utilisée actuellement est intrinsèquement déterministe grâce à la méthode d'accès au médium centralisée (méthode maître/esclaves). D'autre part, les comportements temporels sont également déterministes puisque liés directement à l'application (par la définition des séries et des trames). Autrement dit, le déterminisme actuel relève de la connaissance de l'ordre des faits dans lequel ils apparaissent.

A contrario, dans le cas d'une topologie segmentée, le déterminisme dépend de la capacité à prévoir le résultat du comportement du système de communication. En effet, bien que l'utilisation des commutateurs élimine les collisions, celles-ci vont être remplacées par une potentielle congestion dans les commutateurs qui aura un impact sur le comportement (notamment temporel) du système. Par conséquent, c'est cet impact qui devra être évalué de manière déterministe.

Notons que l'étude précédente a couvert en partie cette exigence, selon la philosophie actuelle, dans le cas d'un scénario particulier non spatial. En effet, le comportement des solutions évaluées précédemment était intrinsèquement déterministe par leur méthode d'accès au médium. Néanmoins, lorsque nous allons relâcher la contrainte d'utilisation du mode maître/esclaves et qu'il y aura alors plusieurs trames sur le réseau, il sera nécessaire de répondre à cette exigence en prévoyant de manière déterministe le comportement du système (chapitre 3).

E. 2 - Temps-réel :

Le système de communication doit assurer le transport de données en un temps borné (fortes contraintes temporelles). Il s'agit de respecter les contraintes de réactivité (temps-réel) du système global.

Dans le paradigme partagé et centralisé de l'application lanceur, seul le trafic contraint est transmis sur le bus fonctionnel (les informations de télémétrie par exemple circulaient par ailleurs). Ainsi, la définition de l'application (donc de l'ordonnancement temporel) permet de respecter et de garantir l'émission en un temps connu.

En revanche, dans le cas d'une topologie segmentée où il pourra y avoir plusieurs trames transmises sur le réseau (de manière potentiellement asynchrone), il est nécessaire de faire appel à une méthode d'évaluation (basée sur une théorie déterministe) de manière à pouvoir garantir les bornes temporelles respectant les contraintes de l'application. Par ailleurs, des mécanismes de classification de service (priorité, ségrégation, ...) pourront être mis en place.

Notons que l'étude précédente a permis de vérifier en partie la réponse à cette exigence

à travers la métrique du temps de cycle minimum dans le cas d'un paradigme de commande centralisée. Néanmoins, ce critère n'est pas adapté aux lanceurs. Ainsi, ce chapitre présentera une étude des délais se basant de nouveau sur un comportement déterministe mais avec une application lanceur (le premier « essai en vol » d'Ethernet). Ensuite, nous allons relâcher la contrainte d'utilisation du mode maître/esclaves. Il sera alors nécessaire d'évaluer les délais de bout en bout de manière déterministe pour vérifier que cette exigence temps-réel est bel et bien respectée (chapitre 3).

E. 3 - Robustesse à la séparation d'étages :

Dans la vie du lanceur, les étages se séparent au fur et à mesure provoquant une modification de la topologie du système de communication. Cette modification entraîne notamment la perte d'équipements. Ainsi, le système de communication doit être robuste à cette évolution de son architecture : il doit être capable d'anticiper, de se préparer ou de se reconfigurer en temps-réel (autant d'un point de vue réseau que d'un point de vue applicatif) et de gérer l'adaptation des lignes (si nécessaire).

E. 4 - Reproductibilité :

La reproductibilité consiste à retrouver un même résultat lors de différentes simulations / expérimentations pour des conditions initiales identiques. Ainsi, là où celle-ci est facilitée dans le paradigme « centralisé » (qui permet un comportement intrinsèquement déterministe), elle peut être plus complexe dans le cas d'un paradigme « distribué » où plusieurs équipements pourront émettre simultanément sur le réseau. En effet, cela pourra se traduire par exemple par des délais aléatoires puisque ceux-ci sont fonction de la charge, c'est-à-dire entre autre de la liste des équipements en compétition à un instant donné. En conséquence, certains paramètres pourront présenter un comportement aléatoire (la gigue par exemple). Cela se traduit par une explosion combinatoire des cas de figure à prendre en considération de sorte qu'il ne soit plus possible de recouvrer une reproductibilité totale (et ce même pour des conditions initiales identiques).

En conséquence, il s'agira à terme d'assouplir cette propriété de reproductibilité totale de manière à pouvoir minimiser le coût des tests du système de communication.

E. 5 - Redondance (disponibilité) :

Les lanceurs doivent être conçus pour opérer dans un environnement fortement contraint, agressif et dynamique. En effet, lors de la poussée, de nombreuses vibrations et rayonnements hors atmosphère sont engendrés ce qui peut provoquer des défaillances. Ainsi, de manière à assurer la continuité de service du système de communication, un certain niveau de redondance doit être développé. Une stratégie d'utilisation de celle-ci devra alors être mise en place. Il faut

pouvoir détecter la défaillance puis la traiter (reconfiguration du réseau en un temps limité) de manière à rendre transparente cette défaillance pour le système global.

La performance de cette redondance peut être qualifiée par :

- le temps de reconfiguration qui correspond au temps pour commuter sur une solution de secours lors d'une défaillance,
- la robustesse : nous définissons la robustesse comme le nombre de solutions de secours (qui peut être vu comme la k -connectivité d'un graphe de communication). Plus le nombre de solutions de secours est grand, plus la capacité de communication est robuste.
- le coût de la redondance : ce paramètre est important mais il est difficile de l'évaluer avec précision. Ainsi, il peut être basé sur le nombre d'équipements ou de liens redondants, sur l'utilisation de composants standards (sans sur-couche propriétaire),...

E. 6 - Observabilité :

L'observabilité est la capacité à observer les échanges sur le réseau. Cette caractéristique est importante pour connaître l'état du système de communication lors des tests au sol préalables au vol mais également lors des analyses menées à l'issue de chacun des vols du lanceur. Cette exigence est extrêmement importante et l'incapacité d'une solution à réaliser cette fonctionnalité pourrait être rédhibitoire pour l'utilisation de celle-ci dans les lanceurs.

Dans le cas d'une topologie à diffusion, l'utilisation du bus permet facilement d'observer l'ensemble des échanges.

En revanche, le changement d'architecture d'un bus à une topologie commutée accroît la complexité d'observabilité. En effet, le trafic ne pourra plus être vu globalement, mais sera observé en plusieurs points du réseau qui devront être définis et étudiés de manière à avoir une vue reconstruite du comportement global du réseau.

E. 7 - Isolation de défaillances :

L'isolation de défaillances consiste en la recherche du (ou des) acteurs responsables d'un problème (une anomalie) qui aurait été auparavant détecté(e). Cette fonctionnalité permet d'écarter (ou au contraire d'incriminer) le réseau. Le système de communication est incriminé dès lors qu'il ne rend pas le service attendu, à savoir le transport de données selon des critères de Qualité de Service. Cette fonctionnalité est utile lors de l'analyse menée à l'issue de chacun des vols. La défaillance pourra alors être rejouée au sol pour réaliser des investigations supplémentaires (non sur le réseau défaillant mais bien sur « une copie » du réseau au sol).

E. 8 - Coût :

Les lanceurs (tout comme tout autre système industriel) sont soumis à la contrainte de réduction des coûts. Ainsi, le système de communication doit également faire l'objet d'une diminution des coûts de manière à réduire le coût global du lanceur. Un premier effort est alors d'étudier une technologie pouvant s'appuyer sur des COTS (*Components Off-The-Shelf*). C'est une des raisons de l'étude de la technologie Ethernet commuté (standard).

E. 9 - Modularité :

La modularité du système de communication correspond à la capacité à pouvoir être décomposé en sous-systèmes qui pourraient être conçus/gérés par différents prestataires. Dans le cas des lanceurs, chaque étage pourrait alors être conçu (et fonctionner) indépendamment du lanceur intégré. Bien entendu, dans le cas d'une gestion centralisée (comme actuellement), cette modularité est plus complexe puisqu'il est nécessaire de posséder le calculateur (donc le lanceur intégré) pour faire fonctionner un étage. En revanche, plus le fonctionnement sera distribué (notamment par étage), plus la modularité pourra être importante.

E. 10 - Evolutivité :

Une architecture réseau est dite évolutive dès lors qu'il est possible de modifier en ligne sa topologie physique et logique mais également le mode de communication (Maître/esclaves, Producteur/consommateurs ...). Autrement dit, l'architecture du système de communication d'un lanceur peut évoluer en fonction de la mission entraînant la nécessité de pouvoir déplacer, ajouter ou supprimer des nœuds. L'insertion/réinsertion à chaud d'équipements est également à considérer ici.

E. 11 - Masse :

Les lanceurs sont soumis à la contrainte de réduction de la masse. Ainsi, le système de communication doit également réduire sa masse de manière à diminuer la masse globale du lanceur. Une attention particulière devra donc être portée quant à la longueur totale de câbles utilisés

Autres exigences liées aux lanceurs :

Bien entendu, d'autres exigences sont spécifiées par ailleurs comme la robustesse des composants à l'environnement (radiations, vibrations,...), les problématiques liées aux embargos ITAR (*International Traffic in Arms Regulations*) des composants, ... mais ces exigences sont en dehors de nos compétences. Elles ne seront donc pas présentées (ni traitées) dans cette section.

Les chapitres suivants auront donc pour objectif de montrer comment la technologie Ethernet pourra répondre aux différentes exigences que nous avons énoncées.

L’étude précédente a permis de retenir Ethernet pour une étude approfondie au cas des lanceurs. Cette étude se basait sur un scénario industriel dans lequel la métrique de temps de cycle minimum était pertinente. Or, cette métrique n’est plus suffisante dès lors qu’un scénario spatial est considéré. Nous allons donc compléter cette étude en réalisant l’évaluation analytique et par simulation des délais d’une application spatiale sur Ethernet de manière à vérifier le respect de l’exigence E. 2 (temps-réel). Ainsi, les mêmes hypothèses d’étude seront conservées (réseau dédié, mode de communication maître/esclaves) ce qui permet de maintenir (dans un premier temps) le caractère intrinsèquement déterministe (E. 1). Pour réaliser l’évaluation du « premier vol » d’Ethernet, il est nécessaire de connaître le trafic applicatif d’un point de vue qualitatif et quantitatif. L’originalité de la méthode consiste à extraire de la trace d’un vol (issu de l’observabilité d’une station « monitor » sur le bus MIL-STD-1553B) les différentes informations de chacun des trafics (source, destination, inter-arrivée, taille des messages). Outre ces informations, aucune architecture n’étant définie a priori (dû au passage d’un bus à une architecture commutée), deux architectures ont alors été proposées (tout en essayant de satisfaire a priori aux exigences E. 3 et E. 5, c’est-à-dire robustesse à la séparation d’étages et redondance).

2.3 Premier « essai en vol » d’Ethernet

2.3.1 Analyse des communications sur le bus MIL-STD-1553B

Cette analyse requiert la compréhension du fonctionnement du protocole MIL-STD-1553B et plus particulièrement son mode de communication et le format des messages (présenté dans le premier chapitre). De plus, les caractéristiques des flux (source, destination, longueur des données utiles et inter-arrivée) sont extraites de la trace d’un vol (un extrait est donné dans la TABLE 2.1 : seules les colonnes Cpt, Date, Mess et Lg seront utilisées dans cette analyse correspondant respectivement au Compteur ou numéro de l’échange, la date, la référence de l’échange et la longueur des données utiles).

L’aspect temporel des échanges et plus particulièrement leur inter-arrivée est extrait du fichier de trace. La trace contient une date (d’acquisition) qui sera considérée comme la date d’émission (sur un bus, c’est relativement le cas à un temps de propagation près). Il s’agit ensuite d’effectuer la différence entre deux dates de chaque message d’un même flux pour obtenir l’inter-arrivée.

Pour illustrer cette analyse, prenons le premier transfert du fichier trace. Chacun des flux est identifié grâce à une référence (colonne « Mess », ici AG13025) constituée du sens du transfert (AG), du bus sur lequel se trouve l’abonné (1), de l’adresse de l’équipement (30) et de la sous-adresse de l’équipement (25). Ce transfert est émis une première fois à la date $t_2 = -3,0015 s$ (FIGURE 2.1 : transfert encadré rouge) et une seconde fois à la date $t_2 = -2,983520 s$ (trans-

TABLE 2.1: Extrait de la trace d'un vol

Cpt	Date	Trf	Stf	Id Bl	Mess	MC	MS	T'TAG	Lg	Données
1	-3.001500	0	1	C024	AG13025	F724	F000	315D	4	811D 1B72 6977 4972
2	-3.001310	0	1	C01C	AG23025	F724	F000	3160	4	811D 1D72 5107 DD72
3	-3.001060	0	1	C024	AG13022	F6C3	F000	3164	3	811D 07B6 0710
4	-3.000860	0	1	C01C	AG23022	F6C3	F000	3167	3	811D 09B2 0710
5	-3.000670	0	1	C024	AG12710	DD47	D800	316A	7	0000 FFFF 0000 0000 0003 0400 0400
6	-3.000350	0	1	C01C	AG22710	DD47	D800	316F	7	0000 0000 0000 0000 0000 0400 0400
7	-3.000100	0	1	C024	AG11310	6D41	6800	3173	1	0103
8	-2.999900	0	1	C01C	AG21310	6D41	6800	3176	1	0103
9	-2.996960	0	1	C024	AG11529	7FA5	7800	31A4	5	58FF FFCF FFB8 0294 2D48
10	-2.996700	0	1	C048	AG21529	7FA5	7800	31A8	5	42F8 000B 007E 0294 847A
11	-2.996450	0	1	C045	AG10122	0EC6	0800	31AC	6	8015 C015 8005 8017 C017 8001
12	-2.996190	0	1	C048	AG20622	36C6	3000	31B0	6	8017 C017 83F5 8015 C015 83FE
13	-2.995870	0	1	C045	AG10622	36C6	3000	31B5	6	8015 C015 83FF 8018 C018 8003
14	-2.995620	0	1	C048	AG20122	0EC6	0800	31B9	6	8018 C018 83F9 8014 C014 83F7
15	-2.995360	0	1	C045	AG12922	EEC6	E800	31BD	6	8002 C001 83FA 8004 C004 83E4
16	-2.995100	0	1	C048	AG22922	EEC6	E800	31C1	6	8003 C003 83F1 8002 C002 83EE
17	-2.994850	0	1	C045	AG10122	0EC6	0800	31C5	6	8015 C015 8005 8017 C017 8002
18	-2.994590	0	1	C048	AG20622	36C6	3000	31C9	6	8017 C017 83F5 8015 C015 83FE
19	-2.994340	0	1	C045	AG10622	36C6	3000	31CD	6	8015 C015 83FF 8018 C018 8003
20	-2.994080	1	0	C048	AG20122	0EC6	0800	31D1	6	8017 C018 83F8 8014 C014 83F7
21	-2.993820	0	1	C045	AG12922	EEC6	E800	31D5	6	8002 C001 83FA 8004 C004 83E3
22	-2.993570	0	1	C048	AG22922	EEC6	E800	31D9	6	8003 C003 83F1 8002 C002 83EF
23	-2.992990	0	0	C045	GA11509	7921	7800	31E2	1	000C
24	-2.992610	0	0	C048	GA21509	7921	7800	31E8	1	000C
25	-2.987040	0	0	C045	GA10121	0AA2	0800	323F	2	004D 004D
26	-2.986850	0	0	C048	GA20621	32A2	3000	3242	2	004D 004D
27	-2.986660	0	0	C045	GA10621	32A2	3000	3245	2	004D 004D
28	-2.986460	0	0	C04F	GA20121	0AA2	0800	3248	2	004D 004D
29	-2.986270	0	0	C04A	GA12921	EAA2	E800	324B	2	0000 0000
30	-2.986080	0	0	C04F	GA22921	EAA2	E800	324E	2	0000 0000
31	-2.983520	0	1	C04A	AG13025	F724	F000	3276	4	811D 1B72 6977 4972
32	-2.983260	0	1	C04F	AG23025	F724	F000	327A	4	811D 1D72 5107 DD72
33	-2.983070	0	1	C04A	AG13022	F6C3	F000	327D	3	811D 07B6 0710

fert encadré vert). La différence donne une première valeur d'inter-arrivée à savoir $t_2 - t_1 = -2,983520 - (-3,0015) = 0,01798 s$. Cette opération est alors réalisée sur l'ensemble du fichier de trace de manière à trouver une périodicité (suivant les spécifications, les périodes possibles sont : 18 ms, 36 ms, 72 ms, 144 ms, 288 ms, 576 ms et 1152 ms) et connaître les entités aperiodiques.

2.3.2 Propositions d'architectures Ethernet commutées

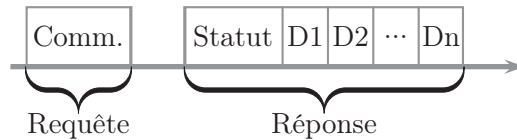
En se basant sur la description précédente du fonctionnement et des caractéristiques des messages, des propositions sont faites de manière à vérifier qu'il est possible de jouer le scénario applicatif sur Ethernet conformément à ce qui se fait sur le réseau MIL-STD-1553B. Il s'agit de faire en quelque sorte de faire un « *MIL-STD-1553B over Ethernet* ».

2.3.2.1 Mode de communication

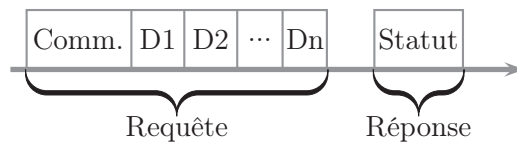
Dans cette étude, le mode de communication maître/esclaves est conservé (rappelons que dans la suite des travaux, nous voulons tendre à se défaire de ce mode de communication qui est non optimal en termes de bande passante notamment). La notion de transfert est alors remplacée

par la notion de requêtes/réponses. Le lien entre ces deux notions est décrit à la FIGURE 2.1.

Échanges Abonné - Gérant :



Échanges Gérant - Abonné :



Comm. : Mot de commande, D : Mot de données et Statut : mot de statut

FIGURE 2.1: Relation entre format de transfert et requête/réponse

Lors d'un échange Abonné - Gérant, la requête émise par le maître correspond au mot de commande du MIL-STD-1553B alors que la réponse est constituée du mot statut et des données. Dans le cas où le maître est producteur des données (échange Gérant - Abonné), les mots de commande et de données sont intégrées dans la requête. La réponse correspond alors à un acquittement (mot de statut). Le détail de la requête et de la réponse est présenté dans la section suivante.

2.3.2.2 Données applicatives

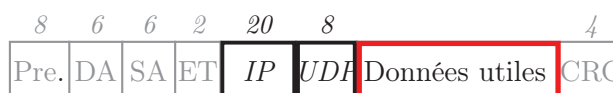
Dans le premier chapitre, il a été défini les formats de messages utilisées sur le réseau MIL-STD-1553B. Comme nous voulons réaliser en quelque sorte un « *MIL-STD-1553B over Ethernet* », il s'agit ici de déterminer les informations des différents mots à conserver pour la réalisation du protocole applicatif sur Ethernet (autrement dit, les informations à mettre dans le champ données).

Commençons par le mot de commande. Sur Ethernet, l'adresse physique de l'équipement est intégrée dans l'encapsulation protocolaire. Le bit « E/R » ne sera donc pas repris sur Ethernet car chaque station est capable de savoir si elle est émetteur ou destinataire du message (*via* ses informations d'adresses source et destination). Les deux autres champs sont à conserver sur Ethernet, à savoir la sous-adresse et le nombre de mots. Pour résumer, une requête est alors composée au minimum de 10 *bits* de données utiles. Au minimum puisque dans le cas d'un transfert Gérant - Abonné, il convient d'ajouter les données à la requête. Par conséquent, il y a $10 + 16.n$ *bits* de données utiles où n est le nombre de mots de données.

Prenons à présent le mot de statut : seuls les deux bits « occupé » et « prise de gestion »

seront conservés sur Ethernet. Le bit « Erreur » ne sera pas repris puisque'une erreur de message sera vue par la couche 2 du modèle OSI grâce au CRC (*Cyclic Redundancy Check*). Donc une réponse est composée au minimum de 2 *bits* de données utiles. Au minimum puisque dans le cas d'un transfert Abonné - Gérant, il convient d'ajouter les données à la réponse. Par conséquent, il y a $2 + 16.n$ *bits* de données utiles où n est le nombre de mots de données.

Ces différentes valeurs de données utiles seront utilisées dans l'évaluation analytique et la simulation d'un réseau Ethernet. La trame utilisée sur Ethernet est récapitulée à la FIGURE 2.2.



Pre : *Preamble*, SA/DA : *Source/Destination Address*, ET : *Ethertype*
 IP : *Internet Protocol*, UDP : *User Datagram Protocol*

FIGURE 2.2: Trames utilisées sur Ethernet (*la longueur des champs sont données en octets*)

2.3.2.3 Inter-arrivée des messages

Afin de reproduire le comportement du réseau MIL-STD-1553B sur le réseau Ethernet en termes d'ordonnancement, périodicité, ... il est nécessaire de connaître l'inter-arrivée des différents messages. L'inter-arrivée des messages sur le réseau MIL-STD-1553B a été estimée dans la section précédente.

Bien que l'ensemble protocolaire soit défini, aucune architecture n'a encore été choisie (puisqu'on n'évolue plus sur un bus où le choix topologique était transparent). La section suivante présente les architectures Ethernet proposées.

2.3.2.4 Présentation des architectures

Dans cette partie, deux architectures sont proposées. Ces deux architectures correspondent à deux extrêmes : une architecture minimaliste d'une part, et une architecture plus évoluée en termes de redondance d'autre part. Bien entendu, elles ne sont pas exhaustives : de nombreuses topologies auraient pu être réalisées entre ces deux architectures Ethernet. Notons, dès à présent, que l'utilisation de commutateurs dans les architectures (bien que non nécessaire avec ce mode de communication où l'on a une seule trame sur le réseau comme dans Modbus/TCP au chapitre 1) permet de répondre implicitement à l'exigence de séparation d'étages (E. 3) ; en effet, il n'y a plus de désadaptation d'impédance lors de la séparation d'étage.

Comme dans le réseau MIL-STD-1553B, l'équipement maître (le *Bus Controller* dans MIL-STD-1553B) est relié à deux commutateurs (avec comme vue d'esprit la représentation des deux bus MIL-STD-1553B) ; les messages destinés au groupe 1 (aux équipements nominaux) seront

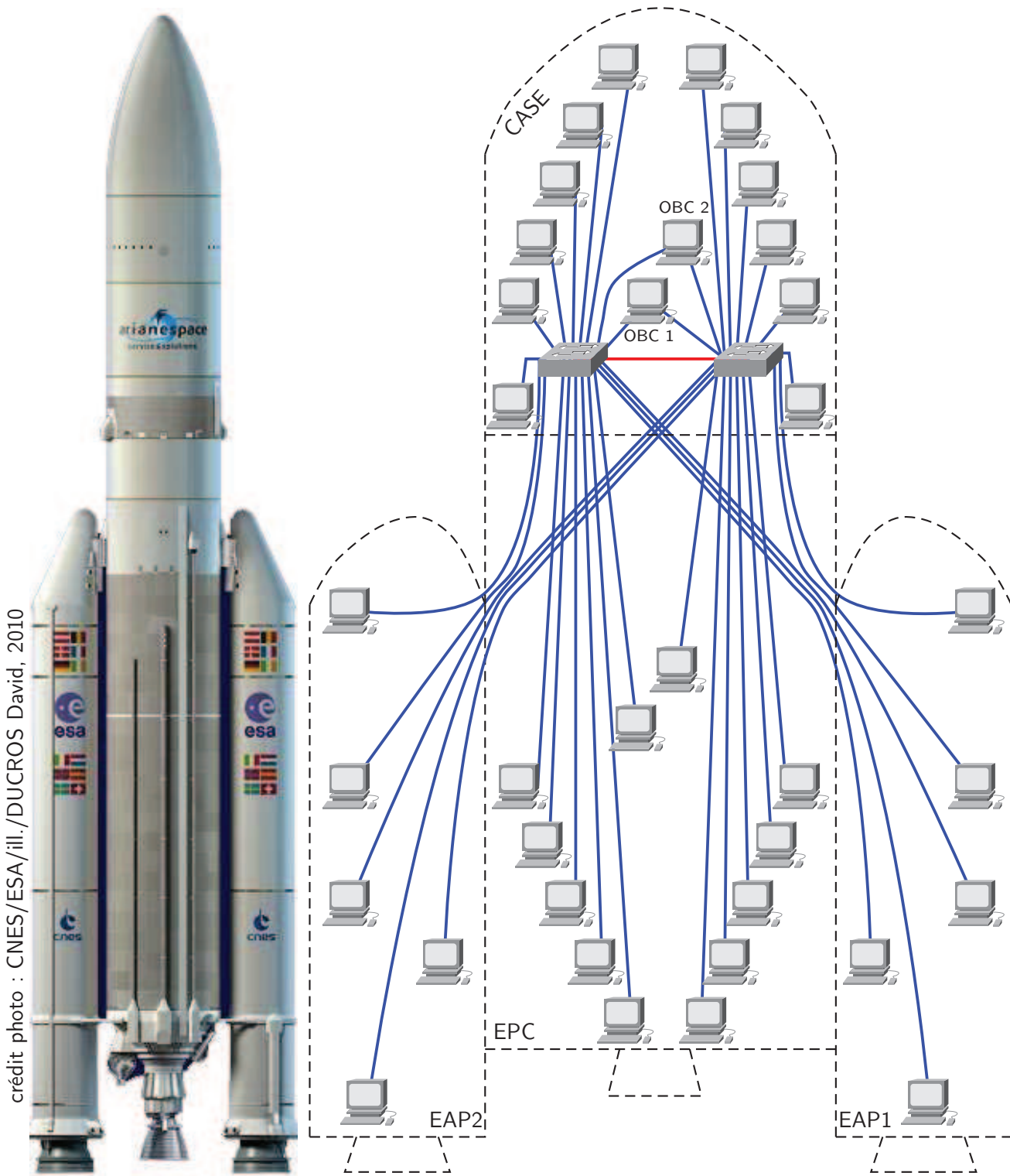


FIGURE 2.3: Architecture 1

dirigés sur un commutateur et les messages destinés au groupe 2 (aux équipements redondés) sur l’autre.

Dans le cadre de travaux de conception d’architectures Ethernet commutées, différentes structures hiérarchiques, fédératives, linéaires ont été proposées (Rüping *et al.*, 1999; Krommenacker, 2002; Jasperneite *et al.*, 2002). Dans le cadre de cette étude, nous nous sommes appuyés sur une structure hiérarchique complétée par des liens supplémentaires (type structure linéaire) permettant d’augmenter la redondance. La répartition hiérarchique s’appuie sur la conception en étage du lanceur, ce qui permettra d’intégrer implicitement la séparation d’étages. Enfin, de par la nature centralisée de la commande d’Ariane 5, il a été considéré de placer le contrôleur OBC (*On-Board Computer*) au niveau de l’étage fédérateur de la topologie commutée.

La première architecture, composée de deux commutateurs et de l’ensemble des équipements (FIGURE 2.3) est minimaliste tout en conservant un niveau de redondance opérationnel relatif aux deux groupes définis précédemment. Ainsi, chaque commutateur est relié à un groupe d’équipements. Le maître (OBC) est quant à lui relié aux deux commutateurs (deux cartes Ethernet) pour envoyer les messages à l’un ou l’autre des groupes.

La deuxième architecture est plus évoluée (FIGURE 2.4). Elle possède deux commutateurs par étage (un par groupe d’équipements). De la même manière que dans la première architecture, chaque commutateur et les équipements qui y sont raccordés représente un modèle de bus MIL-STD-1553B dans chaque étage. Bien entendu, il n’y a bien ici qu’un seul réseau physique Ethernet !

2.3.2.5 Estimation de la longueur des câbles Ethernet

Nous avons vu que la stratégie actuelle est de proposer un lanceur moins cher (exigence E. 9) et moins lourd (E. 11). Ainsi, une première solution pour réduire la masse consisterait à réduire la quantité de câbles. Une estimation de la longueur des câbles des architectures permettra de mettre en évidence si l’ajout de redondance (dans la seconde architecture) ne fait pas exploser la longueur de câbles utilisés.

Prenons un lanceur de hauteur de 50m et de diamètre de 5 *m* ayant un étage CASE (case à équipements) de 20 *m*, EPC (Etage Principal Cryotechnique) de 30 *m* et deux EAP (Etage d’Accélération à Poudre) de 30 *m*. Bien entendu, ce ne sont pas les vraies dimensions des étages (qui permettraient évidemment une estimation bien plus précise) mais elles permettent d’avoir un ordre de grandeur de la longueur de câbles globale (il s’agit en fait d’un majorant de la longueur réelle, car nous considérons en plus tous les équipements aux extrémités des étages). Ainsi, en connaissant la longueur d’un lien (considéré identique pour tous les équipements d’un même étage), du nombre de lien par étage, il est possible d’estimer la longueur de câbles Ethernet globale. La TABLE 2.2 résume les différents paramètres pris en compte pour l’estimation ainsi que les résultats obtenus.

crédit photo : CNES/ESA/ill./DUCROS David, 2010

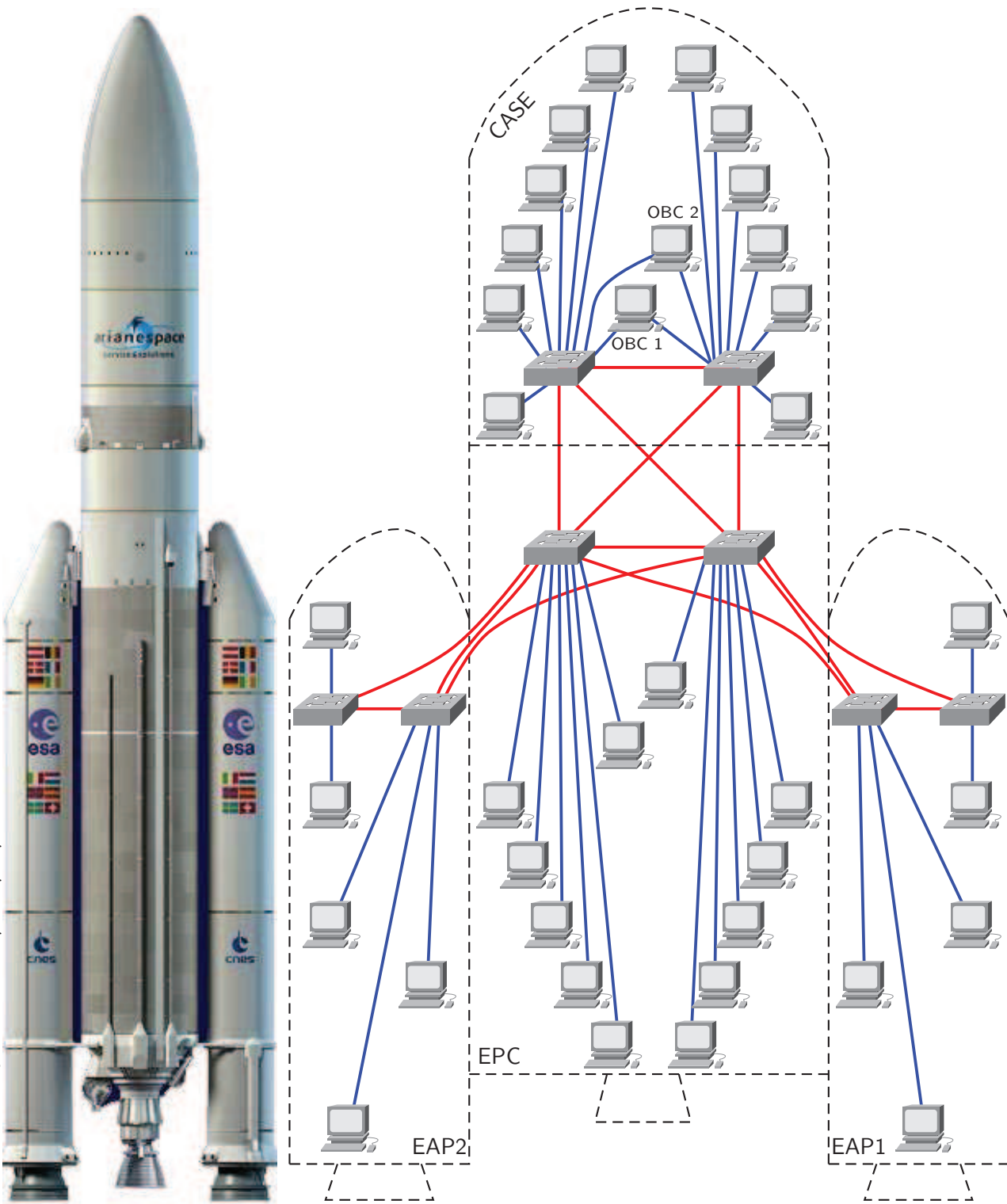


FIGURE 2.4: Architecture 2

TABLE 2.2: Estimation de la longueur de câbles

Architecture	Etage	Longueur d’un lien	Nombre	Taille totale
1	CASE	20m	17	340m
	EPC	30m	12	360m
	EAP 1	35m	5	175m
	EAP 2	35m	5	175m
	TOTAL :			1050m
2	CASE	20m	21	420m
	EPC	30m	18	540m
	EAP 1	35m	6	210m
	EAP 2	35m	6	210m
	TOTAL :			1380m

2.3.2.6 Commentaires

La première architecture correspond à la traduction directe d’une architecture de commande centralisée. Elle présente l’avantage d’être de faible complexité puisqu’elle possède un faible nombre de commutateurs. Inversement, la seconde architecture, plus complexe, correspond à la traduction de contraintes spatiales et plus particulièrement en termes d’étages. Elle possède un nombre plus important de commutateurs permettant d’ajouter de la redondance de liens (qui se traduit directement par une redondance de chemins : un combinatoire d’arbres logiques importants). Malgré l’ajout de matériel et de liens pour la redondance dans l’architecture 2, on peut remarquer que la longueur totale de câbles n’explose pas (ce qui aurait pu dans le cas contraire être rédhibitoire vis-à-vis de la contrainte de masse relative à la longueur de câbles utilisés).

En termes de séparation d’étages (E. 3), l’utilisation de commutateurs permet l’adaptation d’impédance dans les deux architectures. La pyrotechnie est actuellement utilisée pour la séparation d’étages. Selon la technique d’explosions utilisée (par lien ou par groupe de liens), l’architecture 1 peut potentiellement demander plus d’explosions que l’architecture 2.

D’un point de vue modularité (E. 9), l’architecture 2 présente une segmentation fine par étage ce qui la rend plus modulaire que l’architecture 1. Cependant, dans l’architecture 1, un équipementier pourra quand même fournir son équipement, et le connecter au commutateur sans problème du fait de l’interopérabilité offerte par l’utilisation d’un standard Ethernet.

D’un point de vue évolutivité (E. 10), les deux architectures permettent l’ajout, la suppression ou le déplacement d’équipements. En revanche, pour des questions de passage de câbles, l’architecture 2 permettra plus aisément ces modifications (car présente une segmentation plus

fine par étage). De plus, dans les deux architectures, le mode de communication peut être modifié mais apportera un gain d'efficacité beaucoup plus faible dans l'architecture 1.

Bien que ces deux architectures répondent a priori à quelques exigences (E. 3, E. 9 et E.10), une évaluation temporelle permettra de vérifier si ces architectures répondent à l'exigence temps réel (E. 2). Cette évaluation est présentée dans la section suivante.

2.3.3 Evaluation temporelle des architectures proposées

Cette section a pour objectif de présenter les résultats temporels (délai de bout-en-bout) obtenus sur les deux architectures proposées. Pour cette analyse, il a été considéré qu'il n'y a aucune défaillance (de liens et/ou d'équipements). La topologie logique est définie logiquement de manière à éliminer les boucles (pouvant entraîner la saturation du réseau, par d'infinies retransmissions). De plus, les commutateurs utilisent (tous) le mode *store & forward*.

L'objectif de cette première étude menée sur une trace d'un vol est de montrer qu'Ethernet est capable d'obtenir (au minimum) des délais du même ordre de grandeur que ceux calculés sur le bus actuel. Les différents paramètres pour jouer le scénario sur Ethernet sont ceux extraits de la trace (Table 2.1) et analysés dans la section 2.3.1.

2.3.3.1 Analyse temporelle théorique

Comme le mode de communication est de type maître/esclaves, l'évaluation (théorique) des délais de bout-en-bout peut être faite analytiquement et de manière déterministe sans avoir recours à des outils spécifiques. Le comportement temporel des échanges est représenté sur la FIGURE 2.5. En considérant ce profil et en notant n le nombre de commutateurs traversés, le délai de bout-en-bout d'une requête ou d'une réponse est la somme de :

- n fois les latences ℓ pour traverser un commutateur plus deux fois le délai de propagation δ entre le contrôleur/équipement et un commutateur,
- et $n + 1$ fois le temps de transmission τ d'une trame dû au mode de configuration des commutateurs (ici, en mode *store & forward* par exemple).

Le délai de bout-en-bout peut alors être écrit de la façon suivante :

$$\Gamma = 2\delta + n\ell + (n + 1)\tau$$

Le temps de transmission est calculé en considérant le format des requêtes/réponses. 8 *octets* sont ajoutés par UDP, 20 *octets* par IP et finalement 26 *octets* par Ethernet (trou inter-trame exclu). En considérant la taille de données utiles x et avec le respect de la taille minimale des données Ethernet, le temps de transmission est donné par :

$$\tau = 8 \frac{26 + \max(46, 20 + 8 + x)}{C}$$

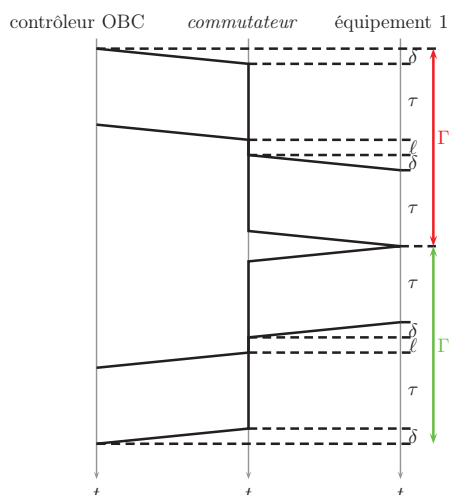


FIGURE 2.5: Comportement sur Ethernet des communications maître/esclaves

Connaissant le débit (100 Mb/s) et la taille des données (variant selon chaque flux), il ne reste plus qu'à déterminer le nombre de sauts n en fonction de l'arbre logique initial (pour l'architecture 2 notamment, l'architecture n'étant composée que d'un seul saut $n = 1$). La FIGURE 2.6 présente l'arbre logique initial de l'architecture 2 (les ports des liens qui ne sont plus représentés ont été inhibés de manière logicielle). Ainsi, selon la localisation des équipements dans les étages du lanceur, le nombre de sauts diffère. La TABLE 2.3 représente les différentes valeurs de n en fonction des équipements et du fait que le maître envoie sur l'un ou l'autre des commutateurs en fonction de la destination du message (groupe nominal ou redondant).

TABLE 2.3: Nombre de sauts en fonction des équipements destinataires sur l'architecture 2

<i>Equipements</i>	Nombre de sauts
CASE (bus 1)	1
CASE (bus 2)	1
EPC (bus 1)	2
EPC (bus 2)	3
EAP1 (bus 1)	3
EAP1 (bus 2)	4
EAP2 (bus 1)	3
EAP2 (bus 2)	4

Pour le calcul, il est considéré que les temps de propagation sont négligeables. La valeur de la latence des commutateurs retenue est $\ell = 0.35 \mu\text{s}$. La taille des données utiles est comprise entre 10 bits (qui correspondra alors à une trame Ethernet de taille minimale avec remplissage)

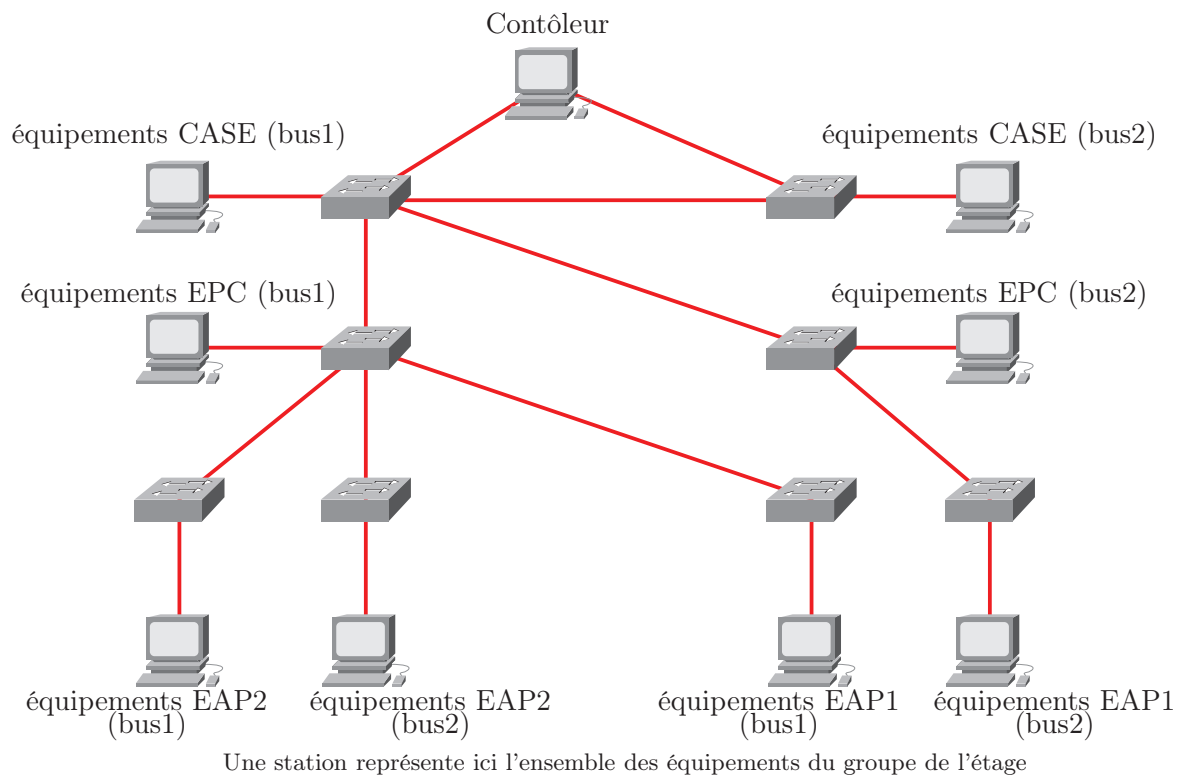


FIGURE 2.6: Arbre logique initial

et 522 bits.

Les délais de bout-en-bout dans l'architecture 1 sont alors compris entre $11,8 \mu s$ et $19,4 \mu s$ ($11,8 \mu s < \Gamma < 19,4 \mu s$). Sur la seconde architecture, ils sont compris entre $11,8 \mu s$ et $30,2 \mu s$ ($11,8 \mu s < \Gamma < 30,2 \mu s$). Cette variation s'explique par la différence du nombre de sauts pour les différents équipements.

Une analyse similaire a été menée sur les délais obtenus sur le bus 1553 : le réseau des lanceurs actuels introduit des délais de bout-en-bout compris entre $20 \mu s$ et $660 \mu s$ ($20 \mu s < \Gamma < 660 \mu s$).

2.3.3.2 Analyse temporelle par simulation

La simulation du scénario présenté précédemment a été réalisée sous le simulateur réseau Opnet Modeler. Celui-ci permet de représenter des architectures de communication, de modéliser des trafics et d'observer le comportement du réseau à travers un large panel de statistiques.

Un des intérêts de l'utilisation de cet outil est la capacité à construire un réseau grâce à des bibliothèques de modèles constructeurs (Cisco, 3Com, ...). D'autre part, il faut savoir qu'il est possible :

- de développer et simuler ces propres protocoles de communication (voire même modifier ceux existants),
- d'utiliser OPNET comme émulateur avec un interfaçage avec une partie du réseau réel,

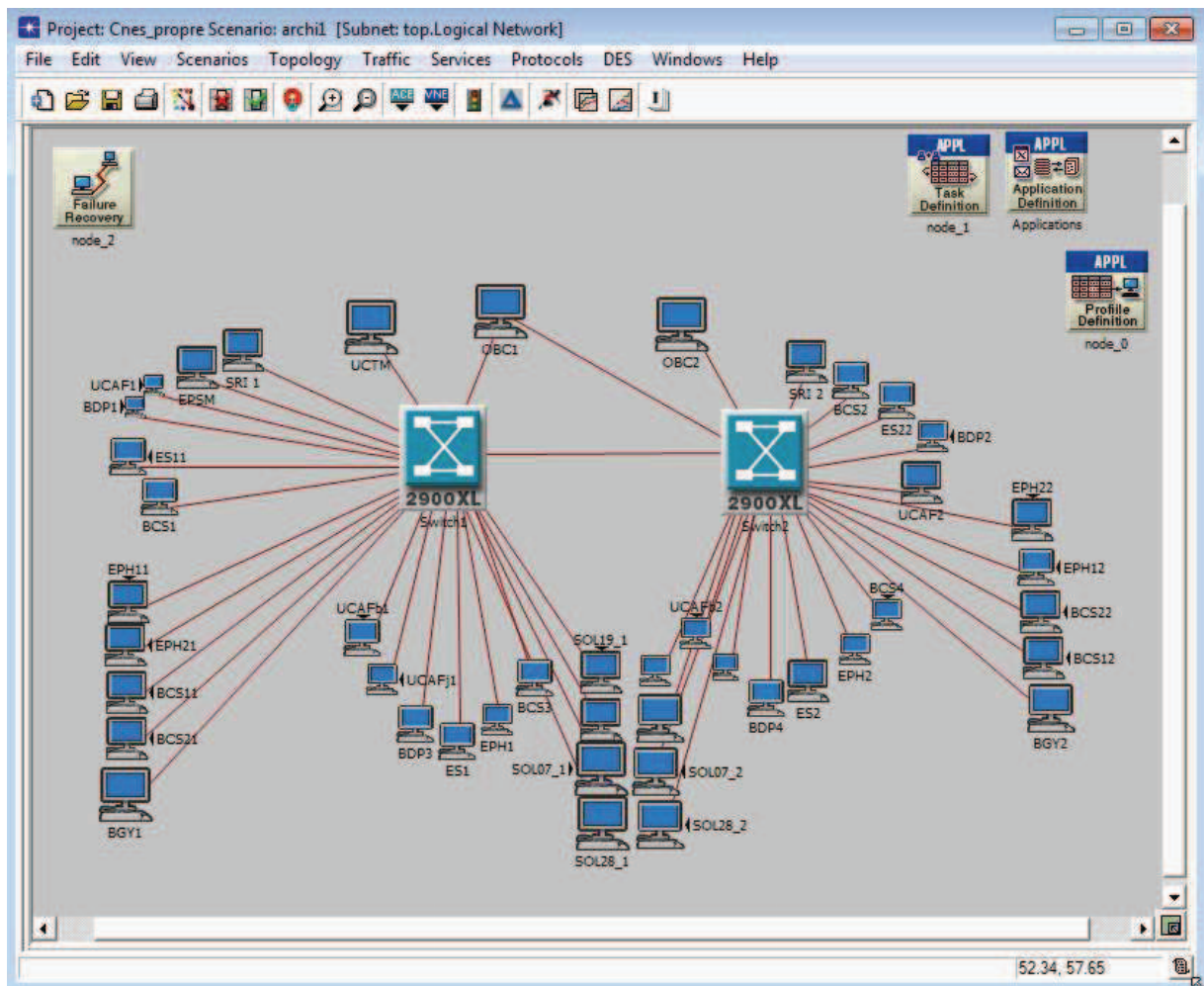


FIGURE 2.7: Simulation de l'architecture 1 dans le simulateur Opnet

– et de prendre en compte la nature du terrain sur l'élaboration de réseaux sans fil.

Ainsi, pour réaliser la simulation du scénario présenté précédemment, les architectures ont été construites avec le modèle de commutateur Cisco Catalyst 2912 XL avec des liens 100 Mb/s : une illustration de la première architecture dans le simulateur est fournie à la FIGURE 2.7. Les trafics ont été modélisés en fonction des caractéristiques des flux (extraites des traces comme vu précédemment) et du comportement maître/esclaves (à une requête est associée une réponse avant de réémettre une autre requête).

La simulation nous permet alors d'obtenir les délais de bout-en-bout pour chacun des flux : par exemple, la FIGURE 2.8 illustre le délai d'un flux émis par l'équipement UCAF1 sur l'architecture 1. Comme nous sommes dans la situation où un seul paquet est sur le réseau à un instant donné, nous obtenons des délais constants pour chaque flux.

En résumé, les délais de bout-en-bout dans l'architecture 1 sont alors compris entre $11,8 \mu s$ et $19,5 \mu s$ ($11,8 \mu s < \Gamma < 19,5 \mu s$) et ceux dans la seconde architecture entre $11,8 \mu s$ et $30,1 \mu s$

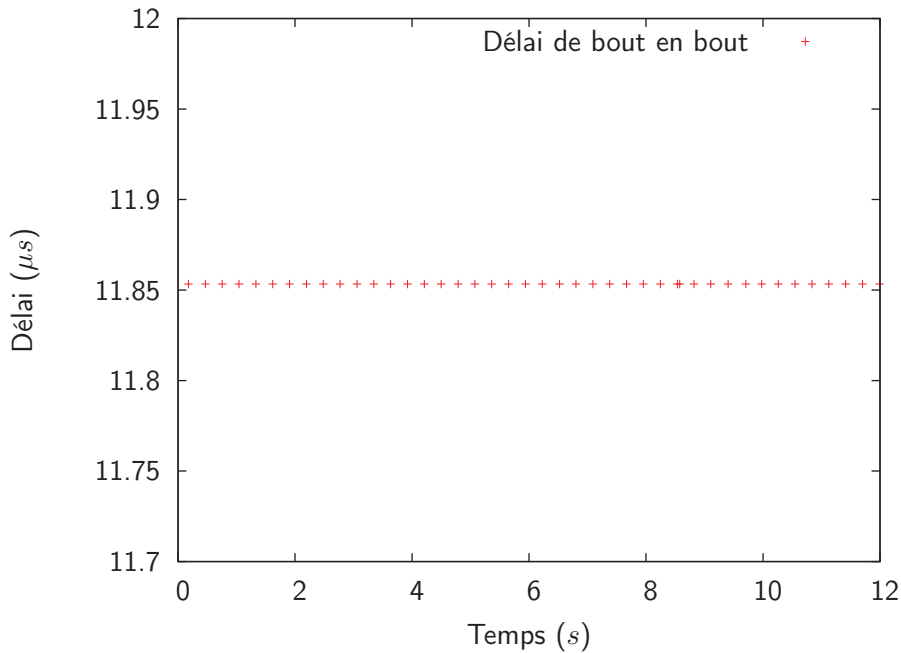


FIGURE 2.8: Délai d'un flux émis par l'équipement UCAF1 obtenu par simulation

($11,8 \mu s < \Gamma < 30,1 \mu s$) selon la taille des trames des différents flux (de 10 *bits* à 522 *bits* de données utiles).

2.3.3.3 Commentaires

Dans cette section, les échanges issus de la trace d'un vol ont été mis en œuvre sur une architecture Ethernet à travers une évaluation analytique des délais de bout-en-bout et par la simulation. On notera que d'un point de vue des délais, des variations (somme toute négligeables, de l'ordre de $0,1 - 0,2 \mu s$) apparaissent entre les valeurs analytiques et de simulation. Cette variation est observée sur les paquets qui ont plusieurs de commutateurs à traverser. La valeur de la latence considérée est principalement responsable de cette variation (celle-ci n'étant pas présentée dans le simulateur).

Rappelons que l'objectif était de vérifier que les performances temporelles du scénario applicatif actuel sur Ethernet était au moins du même ordre grandeur que sur le réseau MIL-STD-1553B. Ainsi, les résultats mettent en avant la similarité des valeurs de délais de bout-en-bout pour de petites tailles de données utiles sur les deux technologies. En effet, lorsque peu de données (une dizaine de bits) sont à transmettre, Ethernet ne profite pas de son débit plus élevé (100 *Mb/s* contre 1 *Mb/s*) puisqu'il est nécessaire d'utiliser une trame de taille minimum (72 *octets*). En revanche, l'utilisation d'une quantité de données utiles plus importante permet de bénéficier de l'augmentation de débit (exemple pour 522 *bits* de données utiles : délai de bout-en-bout égal à 19,5 μs sur Ethernet contre 660 μs sur le réseau 1553).

2.4 Conclusion

Ce chapitre a permis de mettre en avant qu’Ethernet permet de respecter (au minimum) les mêmes contraintes temporelles que celles actuelles. D’autre part, il est clair que ces valeurs ne peuvent pas refléter directement les capacités d’Ethernet. En effet, cette étude de cas est réalisée dans des conditions largement sous-optimales (et ce afin de se placer dans les mêmes conditions que le scénario actuel) : mode de communication maître/esclaves, pas d’optimisation de l’arbre logique (par exemple, les équipements redondants pourraient franchir au minimum un saut de moins), etc. Ces considérations laissent donc entrevoir différentes évolutions (transmettre plus de données utiles par trame, ajout de trafic non fonctionnel, etc) de manière à profiter de la bande passante disponible.

Du fait de sa non-optimalité vis-à-vis de la bande passante, le mode de communication maître/esclaves est abandonné. Il correspondait jusqu’à maintenant au paradigme centralisé. Le passage d’un bus à une architecture commutée permet d’envisager un mode de communication différent tel que le mode producteur/consommateur ou client/serveur. Ce changement permettra un asynchronisme entre l’applicatif et la communication. Ainsi, chaque équipement pourra produire une donnée librement à destination d’un calculateur. Par conséquent, les flux pourront alors être en concurrence pour l’accès aux ports de sortie des commutateurs. Cette concurrence peut alors se traduire par de la congestion et donc des temps de traversée des commutateurs plus élevés, variables et inconnus. L’ordre des échanges devient alors potentiellement non garanti et une analyse comme celle présentée dans ce chapitre n’est plus envisageable. Comme précisé dans l’exigence de déterminisme (E. 1), l’analyse ne s’appuie plus sur le comportement déterministe de la solution mais doit s’appuyer sur une méthode déterministe pour prévoir le résultat de ce comportement. Ainsi, nous évaluerons dans le chapitre suivant les performances temporelles d’une architecture Ethernet grâce à la théorie du calcul réseau.

Ce chapitre a permis de « rejouer un vol » Ariane 5 sur Ethernet avec le même succès que sur le MIL-STD-1553B. Cet « essai » n’a cependant pas été réalisé dans des conditions optimales (en termes de bande passante, d’évolutivité, etc). Donc, nous quittons à présent le monde Ariane 5 pour nous tourner vers un lanceur « idéal » (Ariane 6) bénéficiant de futures évolutions. Nos contributions vont permettre :

- de distribuer l’avionique sur Ethernet,
- d’assurer la disponibilité du système de communication et
- d’observer pour « jouer au sol » et isoler les défaillances.

Chapitre 3

Distribuer une avionique sur Ethernet ? (Robert *et al.*, 2011)

3.1 Introduction

L'étude précédente a permis de jouer un vol Ariane 5 sur Ethernet. Jusqu'ici, le modèle de communication s'appuyait sur une architecture centralisée, c'est-à-dire non distribuée (où la commande est concentrée sur un seul équipement) et sur un mode de communication maître/esclave type *pooling*. Du fait de sa non-optimalité vis-à-vis de la bande passante, il est envisagé de ne pas reconduire le mode de communication maître/esclave sur Ariane 6. En effet, ce mode n'est pas le seul mode possible pour une architecture de commande (qu'elle soit centralisée ou distribuée) : un mode de communication producteurs/consommateur pourrait être également utilisé. Pour le réseau, cela signifie que la communication n'est plus à la seule initiative d'un maître unique. De manière macroscopique, on passe d'un mode où l'on ne retrouve à un instant donné qu'un seul paquet sur le réseau à un nouveau mode où plusieurs paquets peuvent être en concurrence sur le réseau (cf. FIGURE 3.1). De plus, rappelons que dans le cadre de l'unification future des réseaux de commande et de télémétrie, le réseau pourra alors supporter des trafics additionnels à ceux de la commande.

Ce nouveau mode disqualifie alors l'étude précédente. En effet, la concurrence des flux peut entraîner de la congestion au niveau des commutateurs. Il convient cependant d'être capable de déterminer formellement et de manière déterministe (E. 1) si les exigences temps réel (E. 2) sont néanmoins satisfaites. Nous proposons donc de réaliser l'évaluation des architectures Ethernet grâce à une théorie déterministe d'analyse des files d'attente. Cette évaluation aura pour objectif de calculer les majorants des délais (de chacun des flux transportés par le système de communication).

Notons que le choix de centralisation de la commande implique, d'un point de vue réseau,

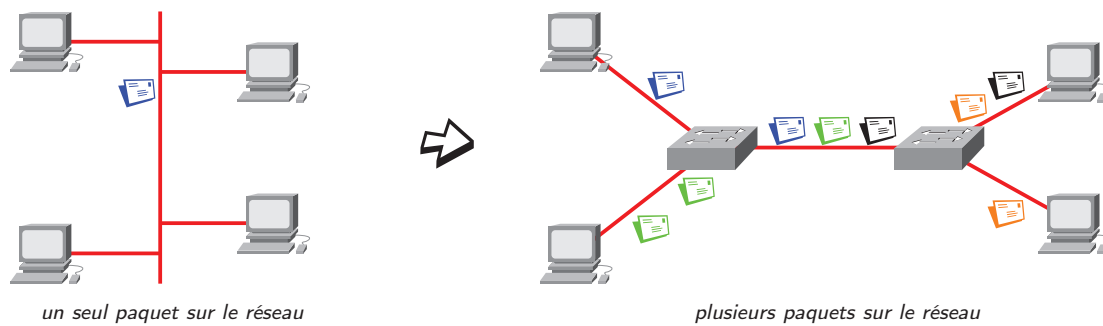


FIGURE 3.1: Evolution de la topologie et du mode de communication (De Ariane 5 vers Ariane 6)

un modèle de communication particulier. Ainsi, il est nécessaire que la distribution (ou non) de la commande soit une réflexion préalable à l'évaluation du système de communication. En effet, la question de la distribution de tout ou partie de la commande nécessite lors du passage de l'architecture fonctionnelle à l'architecture organique d'isoler les fonctions pouvant être implantées sur des composants spécifiques et de définir les interfaces de communications nécessaires à l'exécution de ces fonctions. Cela doit également conduire à la définition d'exigences de qualité de service concernant ces communications. La spécification de la distribution de la commande permet d'établir un modèle de communication accompagné d'exigences temps critiques et de fiabilité qui seront alors les références de l'évaluation / qualification du réseau. Ainsi, le mode de communication approprié sera donc sélectionné en fonction de sa satisfaction aux exigences de Qualité de Service.

Comme ce choix n'est encore pas réalisé (notamment parce que la commande n'est pas encore établie), une illustration de l'évaluation d'architecture sera menée d'une part sur un mode centralisé et sur un mode distribué d'autre part avec l'objectif sous-jacent de montrer si la distribution peut être envisageable pour les lanceurs spatiaux. Nous montrerons qu'un gain peut être observé.

3.2 Evaluation des majorants des délais

3.2.1 Outils d'évaluation des délais

Il existe différentes méthodes d'évaluation des délais dans le cadre de la communauté des systèmes à événements discrets candidates à l'étude des réseaux Ethernet commutés. D'une part, on peut citer un certain nombre d'outils comme ceux de la simulation. On retrouve ainsi des travaux sur les réseaux de Pétri colorés temporisés (Brahimi, 2007; Zaitsev et Shmeleva, 2006; Marsal, 2006) ou encore des travaux basés sur les simulateurs Modelica/Dymola/True time/Opnet/Omnet++ (Liu et Frey, 2007; Cervin *et al.*, 2003; Pereira *et al.*, 2004b). Dans le cadre

des systèmes embarqués, l'évaluation des performances temporelles, lorsqu'elle est utilisée dans un contexte de certification comme pour l'A380, fait néanmoins appel à des théories différentes. On peut ainsi citer les méthodes de vérification formelle par *model-checking* (Ruel, 2009) ou les méthodes analytiques tels le calcul réseau ou l'approche par trajectoire. Une des principales limites du *model-checking* est souvent l'explosion combinatoire de l'ensemble des états. La théorie du calcul réseau permet le calcul de majorants des délais de bout en bout (Georges *et al.*, 2002; Grieu, 2004). La méthode par trajectoire (Bauer, 2011) consiste à étudier l'ordonnement produit par chacun des nœuds traversés par les flux de données.

Nous avons retenu la théorie du calcul réseau pour évaluer les majorants des délais de nos architectures, notamment parce que cette théorie a permis la certification de l'AFDX pour l'A380 (Grieu, 2004).

La théorie du calcul réseau (ou *network calculus*) a été initiée par (Cruz, 1991), (Chang, 2000) et (Le Boudec et Thiran, 2001). Basée sur la théorie Min-Plus, cette théorie est utilisable pour les systèmes Ethernet temps réel comme le montre de nombreux travaux (Grieu, 2004) (Georges, 2005) (Diouri, 2010) (Charara, 2007) dans des domaines différents (industriel, systèmes contrôlés en réseau, avionique, ...). D'une manière générale, nous nous appuyerons sur les notations introduites par (Le Boudec et Thiran, 2001).

3.2.2 Modélisation des trafics en fonction du scénario de communication

La première étape de ce calcul consiste à établir un modèle de trafic pour chaque flux défini par le scénario de communication. Plus exactement, il s'agit de définir les courbes d'arrivées de chacun de ses flux. La courbe d'arrivée représente la contrainte appliquée à l'arrivée des données d'un flux en entrée d'un système. Elle est définie comme suit.

Soit α , une fonction non décroissante définie pour $t \geq 0$. Un flux R est contraint (régulé) par α si et seulement si :

$$\forall s \leq t, R(t) - R(s) \leq \alpha(t - s) \quad (3.1)$$

La définition précédente (établie par (Cruz, 1991)) peut être exprimée suivant l'algèbre min +. Nous donnons ici les expressions présentées par (Le Boudec et Thiran, 2001) à partir des notations suivantes :

- $\inf S$, l'infimum dans \mathbb{R} , \wedge le minimum dans \mathbb{N} .
- \sup le supremum et $a \vee b = \max\{a, b\}$ le maximum.
- \otimes , la convolution min-plus

Soit f et g deux fonctions ou séquences de F . La min-plus convolution de f et g est la fonction :

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

– et \oslash , la déconvolution min-plus

Soit f et g deux fonctions ou séquences de F . La min-plus déconvolution de f par g est la fonction :

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

(Le Boudec et Thiran, 2001) redéfinissent alors la courbe d'arrivée ainsi :

Soit une fonction α non décroissante définie par $t \geq 0$ ($\alpha \in F$), un flux R est contraint par α si et seulement si

$$\forall s \leq t, R \leq R \otimes \alpha$$

Les différents trafics rencontrés dans les lanceurs peuvent être de nature :

- périodiques (avec ou sans gigue),
- apériodiques ou
- périodiques avec des paquets apériodiques supplémentaires.

Il convient alors de modéliser ces différents types de trafics.

✓ **Modélisation des flux périodiques :** Les flux périodiques sont représentés par une fonction en escalier. Ces flux peuvent présenter une éventuelle gigue qu'il convient de prendre en compte. Cette tolérance permet de mettre en évidence une avance (ou retard) de l'instance suivante. Par conséquent, la gigue (ou tolérance) doit être évaluée. Deux cas de figure sont possibles :

- soit une information de tolérance du type « précision $[0 - 10 \text{ ms}]$ » est donnée dans les spécifications,
- soit aucune information n'est fournie, et elle devra être identifiée à partir d'observations comme une ou plusieurs traces.

Dans les deux cas, il convient de connaître la pire gigue (c'est-à-dire la gigue maximale). Lorsque cette gigue maximale τ est définie, la fonction en escalier est totalement connue et définie par $\alpha(t) = v_{T,\tau}(t) = \lceil (t + \tau) / T \rceil$.

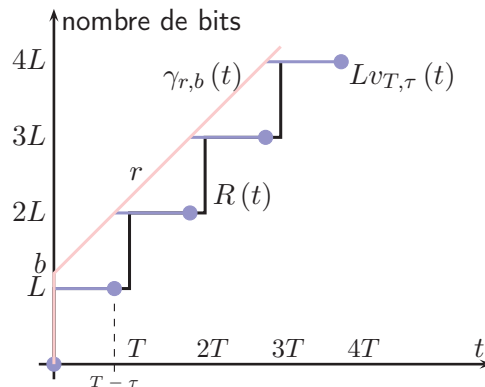


FIGURE 3.2: Courbes d'arrivée possibles pour un flux périodique

Ainsi, si un flux R est contraint par une telle courbe d'arrivée en escalier alors R est également contraint par la courbe d'arrivée affine $\gamma_{r,b}(t)$ (FIGURE 3.2) telle que :

$$\alpha(t) = \gamma_{r,b}(t) = \frac{L}{T}t + L\left(\frac{T + \tau}{T}\right)$$

Dans cette étude, nous nous sommes limité à une courbe d'arrivée affine. Néanmoins, il aurait pu être considéré d'autres courbes d'arrivée qui prennent en compte par exemple l'incapacité physique d'arrivée instantanée des rafales de données. Ces modèles s'intégreraient aisément dans la démarche suivie.

- ✓ **Modélisation des flux événementiels :** Les flux événementiels peuvent être de la même manière que précédemment représentés par une fonction affine. Dans le cas où n paquets sont émis, nous avons fait le choix de modéliser cette fonction par un échelon qui a pour équation $\alpha(t) = nL$ où L correspond à la taille du paquet comme le montre la FIGURE 3.3.

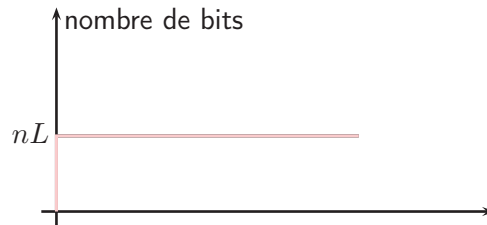


FIGURE 3.3: Représentation de type échelon

- ✓ **Modélisation des flux périodiques avec des paquets événementiels supplémentaires :**

Dans l'application « lanceur », il y a des flux principalement périodiques (avec ou sans gigue) où certains paquets supplémentaires sont émis aperiodiquement à des instants indépendants de la période. Ces paquets peuvent être pris en compte dans la modélisation de ce type de trafic.

La FIGURE 3.4 montre qu'il convient de combiner la représentation des flux périodiques et celle des flux typiquement aperiodiques. Ainsi la fonction affine représentant un flux périodique ayant n paquets aperiodiques s'écrit :

$$\alpha = \gamma_{r,b}(t) = \frac{L}{T}t + \left(\frac{T + \tau}{T} + n\right)L$$

3.2.3 Modélisation du service élémentaire

Le point précédent a permis d'identifier un modèle de trafic pour chaque flux issu du scénario de communication. Il s'agit à présent de s'intéresser à la définition de la topologie réseau adaptée au scénario de communication retenu. Outre les aspects de performances intrinsèques de la topologie, des éléments de redondance doivent être apportés à la conception afin de prendre en

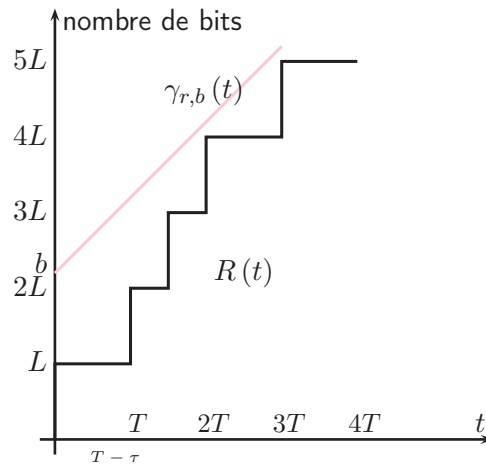


FIGURE 3.4: Courbes d'arrivée possibles pour un flux périodique avec 1 paquet apériodique ($n = 1$)

compte les exigences de fiabilité et de tolérance à la séparation des étages comme identifié par la commande.

Il est proposé ici de s'appuyer sur les deux architectures proposées dans le chapitre 2. Ces topologies sont spécifiques aux réseaux Ethernet où les commutateurs sont utilisés pour éliminer le non déterminisme du protocole. Comme précédemment, le principe méthodologique ci-présenté reste valable quelque soit la topologie retenue. Le choix précis de la topologie se fera suivant un ensemble de critères et plus particulièrement selon les exigences de délais et de fiabilité.

Une fois la topologie définie, il s'agit de déterminer un modèle de cette topologie et dans un premier temps de définir un modèle pour chaque composant de cette topologie. On parle de courbe de service. Elle est définie comme suit.

Soit β , une fonction non négative non décroissante. La quantité de données d'un flux servie par une file est définie par une courbe de service de file (ou *queue service curve* en anglais) β si et seulement si

$$\forall t, \exists t_0 \leq t \text{ tel que } R^*(t) - R(t_0) \geq \beta(t - t_0) \quad (3.2)$$

Une courbe de service implique donc que $\beta(0) = 0$. Cette appellation de courbe de service de file introduite dans (Le Boudec, 1996) renvoie au fait que la définition précédente est dédiée au système à file d'attente. Comme le montre la définition de la courbe de service, il s'agit en fait de la courbe de service minimal puisqu'elle représente le service le plus faible offert par le système. L'intérêt de ce type de courbe est qu'elle nous permettra de prendre en compte au travers du pire service, le pire délai. (Le Boudec et Thiran, 2001) présentent la courbe de service maximale qui permet de travailler sur le délai de traversée minimal.

Selon l'algèbre $\min +$ (et présenté (Le Boudec et Thiran, 2001)), elle peut être définie par :

Soit un système S et un flux traversant S avec comme fonctions d'arrivée et de départ R et

R^* . S offre au flux une courbe de service β si et seulement si

$$\beta \in F, R^* \geq R \otimes \beta$$

Chaque composant est alors modélisé de la façon suivante.

- ✓ **Modélisation d'un lien** : Un lien introduit une latence équivalente au temps de propagation. Cette latence est alors représentée par une fonction retard pur définie de la façon suivante :

$$\delta_i(t) = \begin{cases} 0 & \text{si } 0 \leq t \leq \delta \\ \infty & \text{sinon} \end{cases}$$

Une représentation de ce type de courbe de service est donnée à la FIGURE 3.5.

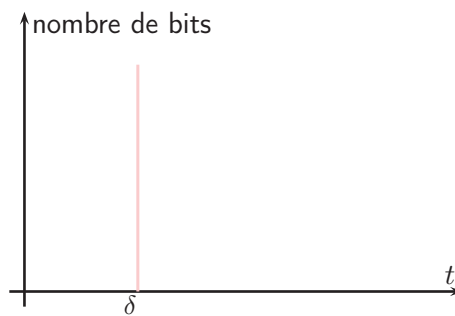


FIGURE 3.5: Courbe de service d'un lien

- ✓ **Modélisation d'un commutateur** : Dans cette étude d'évaluation de performances, nous nous limiterons à l'utilisation d'un modèle simplifié de commutateur puisque la nature exacte des composants qui seraient à terme utilisés n'est pas encore spécifiée. Dans la littérature, des modélisations fonctionnelles plus élaborées ont été proposées dans (Georges, 2005) et (Grieu, 2004) (en particulier, dans le contexte d'architectures AFDX). Ces modèles pourront s'intégrer aisément dans la démarche suivie ici.

Un commutateur introduit une latence électronique (ℓ) et peut transmettre conformément à la capacité de son port de sortie (R). Sa courbe de service est alors représentée par une fonction *rate latency* définie de la façon suivante : $\beta(t) = \beta_{R,\ell}(t) = R(t - \ell)^+$ où $(x)^+ = x$ si $x \geq 0$, $x = 0$ sinon.

Seulement ici, avec cette représentation, le mode du commutateur (*store and forward*, *cut through*, ...) n'est pas pris en compte. Seul le mode *store and forward* est alors considéré. La courbe de service d'un commutateur est ainsi toujours de type *rate latency* (FIGURE 3.6) mais avec une latence plus importante afin de prendre en compte le mode du commutateur. Ainsi, dans ce mode, la trame est totalement mise en mémoire avant d'être retransmise (ce qui implique un temps de transmission t_t). La courbe de service du commutateur (FIGURE 3.6) est alors définie par :

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

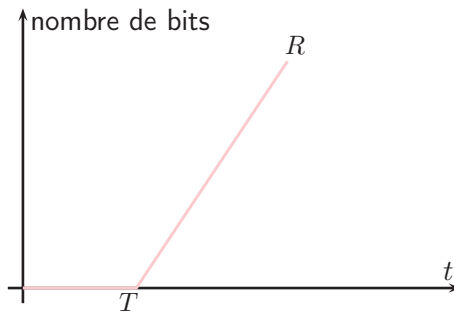


FIGURE 3.6: Courbe de service d'un commutateur

où $T = \ell + t_t$.

3.2.4 Modélisation du système de communication

Les deux premières étapes ont en fait permis d'établir un modèle du trafic (courbe d'arrivée) et un modèle des éléments de la topologie (courbe de service). A partir de là, l'objectif est de modéliser le service offert de bout-en-bout par une architecture réseau. Il s'agit alors de déterminer une courbe de service de bout en bout en utilisant les modèles des composants élémentaires précédemment définis.

Si l'on considère un flux traversant successivement deux commutateurs et trois liens comme représenté à la FIGURE 3.7, le principe de composition des serveurs va nous permettre de déterminer une courbe de service du chemin.

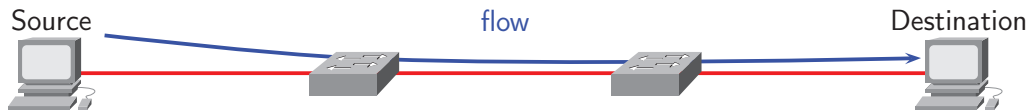


FIGURE 3.7: Concaténation de nœuds

Cette courbe est obtenue en appliquant le théorème 1.5.6 (concaténation de nœuds) défini par Jean-Yves Le Boudec ((Le Boudec et Thiran, 2001)) pour lequel il est défini que le service minimal offert par l'ensemble des n nœuds et des $n + 1$ liens est équivalent à :

$$\bigotimes_{i=1}^n \beta \bigotimes_{i=1}^{n+1} \delta_i$$

où :

$$\bigotimes_{i=1}^n \beta = \beta \otimes \beta \otimes \dots \otimes \beta$$

Ce résultat correspond au principe du *Pay Burst Only Once* qui permet de réduire le pessimisme du majorant du délai de bout en bout. Le calcul mathématique nous permet alors d'obtenir

que le service de bout en bout est contraint par :

$$\bigotimes_{i=1}^n \beta \bigotimes_{i=1}^{n+1} \delta_i = R(t - T)^+$$

avec $T = n * (\ell + t_t) + (n + 1)\delta$.

Ce résultat présente la particularité d'être spécifique à des modes de communication où un seul paquet est présent à un instant sur le réseau. En effet, pour des scénarios de communication où plusieurs stations vont pouvoir transmettre simultanément un paquet, il sera nécessaire de prendre en compte la problématique de l'agrégation de flux en entrée. Ce problème est illustré à la FIGURE 3.8.

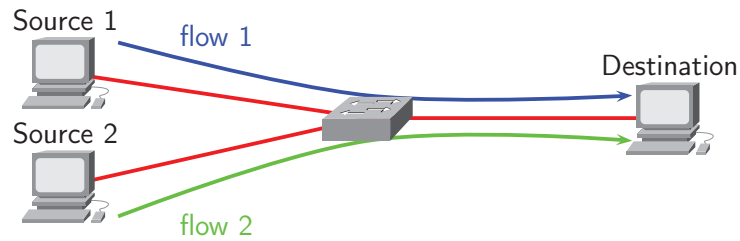


FIGURE 3.8: Illustration du phénomène de multiplexage

La figure ci-dessus montre que lorsque plusieurs flux entrent sur un commutateur simultanément, il devient nécessaire de déterminer une courbe de service spécifique à chaque flux. Cela signifie que l'on ne peut plus se contenter d'une courbe de service par commutateur. Pour déterminer une courbe de service par flux, il est alors nécessaire de s'intéresser au multiplexage des flux réalisé par le commutateur.

Il est alors proposé d'utiliser le théorème du service résiduel (Schmitt *et al.*, 2008) qui permet de déterminer une courbe de service pour un flux donné sous l'hypothèse que le nœud (commutateur) soit FIFO. Ce théorème est défini pour deux flux de la façon suivante :

THEOREM 1 : *Consider a node blindly multiplexing two flows 1 and 2. Assume that the node guarantees a service curve to the aggregate of the two flows. Assume that flow 2 has α_2 as an arrival curve. Then, a service curve for the flow 1 is :*

$$\beta_1 = (\beta - \alpha_2)^+$$

Compte tenu des courbes d'arrivée de type affine (de paramètres b_j et r_j) et des courbes de service des commutateurs de type *rate latency* (de paramètres R et T), le calcul réseau nous permet de déterminer que le service offert pour un commutateur pour un flux donné correspond

à la courbe de type *rate latency* suivante :

$$\beta_i = R'(t - T')^+$$

avec $T' = \frac{\sum_{j=1, i \neq j}^n b_j + \sum_{j=1, i \neq j}^n r_j T}{R'}$ et $R' = R - \sum_{j=1, i \neq j}^n r_j$.

Une illustration de ce service offert à un flux donné en fonction de la courbe de service du commutateur et de la courbe d'arrivée de l'agrégation des flux (de paramètres B et C) est donnée à la FIGURE 3.9.

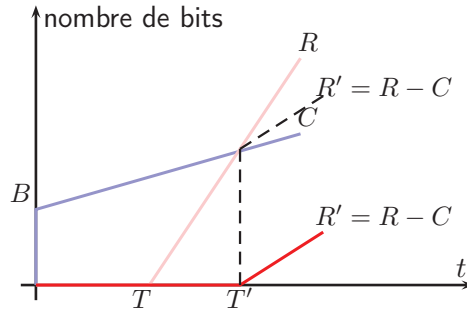


FIGURE 3.9: Courbe de service, courbe d'arrivée de l'agrégation des flux et courbe de service offert au flux considéré

Dans la cas d'une topologie où plusieurs flux vont pouvoir être en concurrence, il n'est alors plus possible d'appliquer directement le principe du *Pay Burst Only Once (PBOO)*. Considérons ainsi la FIGURE 3.10. Dans cette figure, on va chercher à identifier le service offert par le chemin au flux 2.

La FIGURE 3.10(a) démontre bien qu'il n'est plus possible ici d'appliquer le principe de composition des serveurs car, dans ce cas, cela reviendrait à travailler avec β_1 , β_2 et β_3 , c'est à dire les services offerts par les commutateurs à l'ensemble des flux entrants. Ce principe va seulement pouvoir s'appliquer de prime abord au commutateur 2 et 3.

Il est alors suggéré d'appliquer le principe du *Pay Multiplexing Only Once (PMOO)* (Schmitt *et al.*, 2006) qui revient à utiliser tour à tour le PBOO et le service résiduel. Sur cet exemple, on voit que cela implique de n'utiliser uniquement le PBOO lorsque des commutateurs partagent exactement le même trafic comme illustré aux FIGURE 3.10(a) et 3.10(c). Le service résiduel est quant à lui appliqué lorsqu'il n'est plus possible d'utiliser le PBOO comme pour les étapes FIGURE 3.10(a), 3.10(b) et 3.10(d). Au final, on obtient la courbe de service du chemin correspond à la courbe de service d'un commutateur virtuel (FIGURE 3.10(d)) symbolisant la traversée des trois commutateurs.

En conclusion, la modélisation du service de bout en bout doit s'adapter au mode de communication étudié. Dans le cas où un seul flux est présent sur le réseau, on pourra utiliser la formule donnée ci-dessus. Dans le cas inverse, la méthodologie proposée consiste à utiliser le principe du PMOO résumé à la FIGURE 3.10.

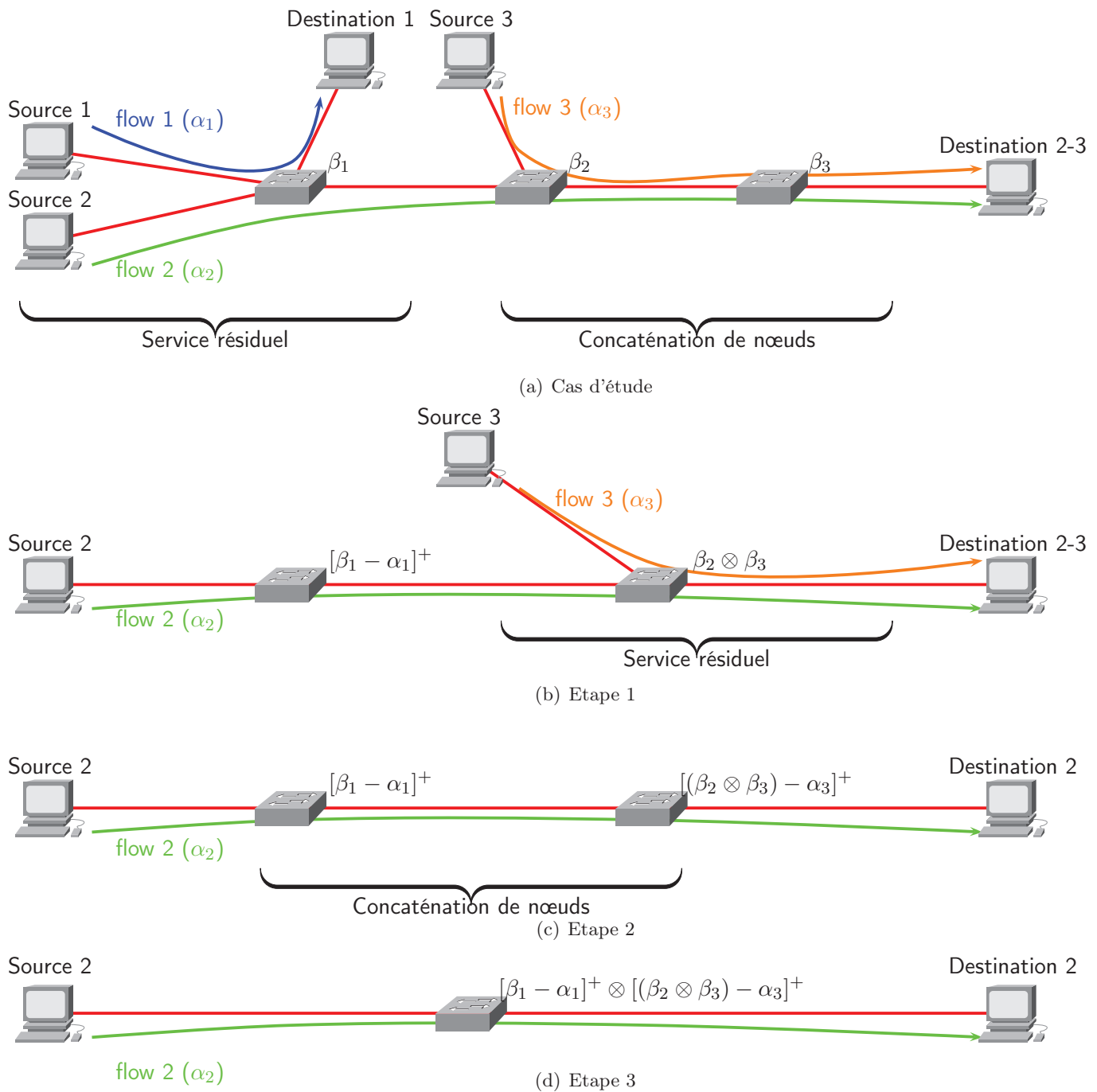


FIGURE 3.10: Principe de détermination d'une courbe de service offert à un flux

3.2.5 Calcul d'un majorant du délai de bout en bout

La section précédente a permis d'identifier une courbe de service pour le chemin suivi pour un flux donné. Dans cette étude, on s'aperçoit que les courbes d'arrivées sont de la forme $\gamma_{b,r}(t) = b + rt$ et que la courbe de service est quant à elle de la forme $\beta = R(t - T)^+$. Dans ce cas, le délai de bout en bout est la distance horizontale entre la courbe d'arrivée et de service comme représenté à la FIGURE 3.11.

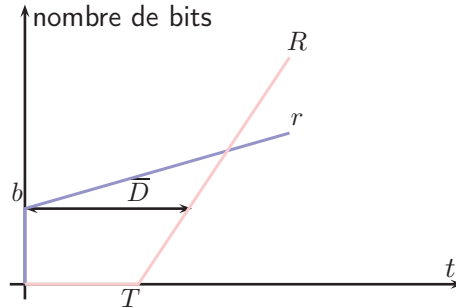


FIGURE 3.11: Courbe de service de bout en bout, courbe d'arrivée et délai de bout en bout

Pour ces hypothèses, il a été montré que le délai de bout en bout est alors majoré par :

$$d(t) \leq T + \frac{b}{R}$$

L'application de cette méthode permet l'évaluation formelle des performances temporelles d'une architecture réseau. Il pourra alors être établi un majorant du délai pour chacun des flux de l'application.

La section suivante présente l'évaluation temporelle des architectures présentées au chapitre 2.

3.3 Évaluation des architectures tests

Dans cette nouvelle section, nous allons appliquer la théorie du *network calculus* (précédemment énoncée) sur la première architecture proposée (FIGURE 2.3) pour un scénario de communication producteurs / consommateurs sur une architecture de commande centralisée. La FIGURE 3.12 illustre un scénario de communication producteur/consommateurs : le producteur émet une donnée à destination d'un ou plusieurs consommateurs (*multicast*) qui peut alors l'utiliser. L'intérêt de la distribution de la commande est ensuite discuté. La seconde architecture (FIGURE 2.4) servira d'appui à cette analyse. En effet, bien que l'architecture de commande puisse être distribuée sur l'architecture 1, cela ne permet pas un gain important en terme d'efficacité.

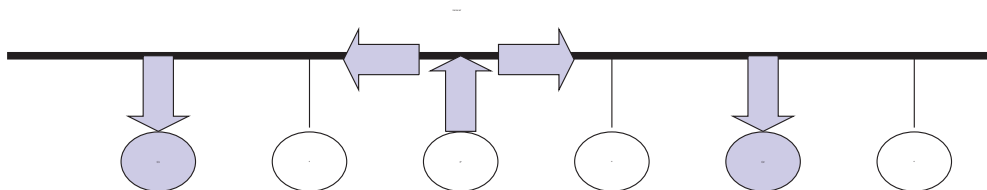


FIGURE 3.12: Mode de communication producteur / consommateurs

Notons que l'étude du système de communication voulue par le CNES est menée avant même la définition fonctionnelle du trafic, de la distribution, etc. C'est pourquoi nous nous appuyons de nouveau sur la définition du trafic de commande issue de la même trace que celle utilisée dans le chapitre 2.

3.3.1 Architecture de commande centralisée à communication type producteurs/consommateur

Dans cette première étude, l'architecture de commande est non distribuée (donc centralisée dans un seul équipement : le contrôleur). Le mode de communication considéré est de type producteurs/consommateur. Par conséquent (et contrairement au mode de communication maître/esclaves), les flux sont en concurrence. Les différentes courbes d'arrivées sont définies conformément à la modélisation proposée précédemment (en fonction de la nature du flux). Dans notre application, ces courbes d'arrivées sont modélisées à partir des informations extraites des traces (identiques à celles présentées dans le chapitre 2). Par exemple, prenons un flux dont on extrait les inter-arrivées et la tolérance (différence entre l'inter-arrivée et la période théorique) suivantes :

N° paquet	Inter-arrivée (s)
1	—
2	1,156680
3	1,151950
4	1,152060
5	1,151940
6	1,160010
7	1,144330
8	1,159630
9	1,145480
10	1,157130
11	1,151560
12	1,146570

N° paquet	Variance (s)
1	—
2	0,0047
3	0,0000
4	0,0001
5	0,0001
6	0,0080
7	0,0077
8	0,0076
9	0,0065
10	0,0051
11	0,0004
12	0,0054

Rappelons que ce sont les spécifications qui donnent comme périodes possibles : 18 *ms*, 36 *ms*, 72 *ms*, 144 *ms*, 288 *ms*, 576 *ms* et 1152 *ms* et permet la définition de la période théorique et le calcul de la gigue. La tolérance maximale retenue pour la modélisation de la courbe d'arrivée de ce flux est donc de 8 *ms*.

A présent, la période théorique (T), la tolérance maximale sur la date d'arrivée (τ) ainsi que la taille de la trame ($L = 576$ *bits*) sont connues. Il est alors possible de déterminer les paramètres de la courbe affine (r et b). Pour rappel, l'obtention des paramètres r et b est définie par :

$$\alpha(t) = \gamma_{r,b}(t) = \frac{L}{T}t + L\left(\frac{T + \tau}{T}\right)$$

Il résulte dans notre exemple que $r = 500$ *b/s* et $b = 580$ *bits*. La courbe d'arrivée de ce flux étant totalement définie, il est alors nécessaire de déterminer la courbe de service offert au flux considéré.

Prenons par exemple, un flux émis par la station EPH 11 (noté AG10103) représenté sur la FIGURE 3.13 et en concurrence avec les flux des autres stations (arrivant sur le même commutateur). C'est le cas critique considéré par le calcul réseau. Afin d'évaluer le délai de bout en bout de ce flux, il est nécessaire de déterminer la courbe de service offert à celui-ci.

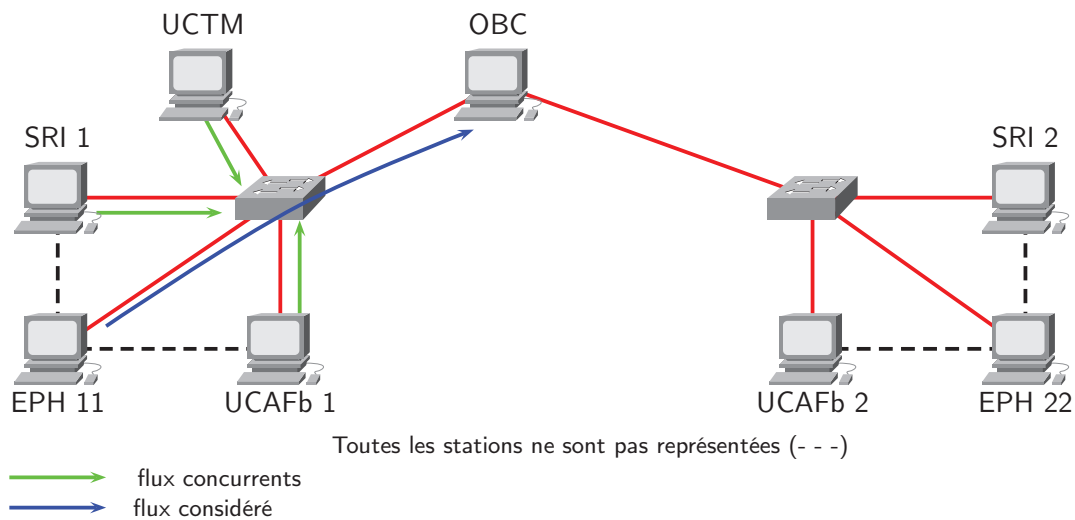


FIGURE 3.13: Exemple du flux AG10103

Puisque plusieurs flux peuvent être en concurrence pour le gain de l'accès au port sur lequel est connecté l'OBC, il conviendrait d'appliquer le principe de *Pay Multiplexing Only Once* défini précédemment. Cependant comme le chemin emprunté par les paquets du flux AG10103 ne comporte qu'un seul saut, il suffit d'appliquer le théorème du service résiduel. La courbe de service de bout en bout alors offert au flux est :

$$\beta_1(t) = R'(t - T')^+$$

avec $R' = R - \sum_{j=2}^n r_j$ et $T' = \frac{\sum_{j=2}^n b_j + \sum_{j=2}^n r_j(\ell + t_p)}{R'}$.

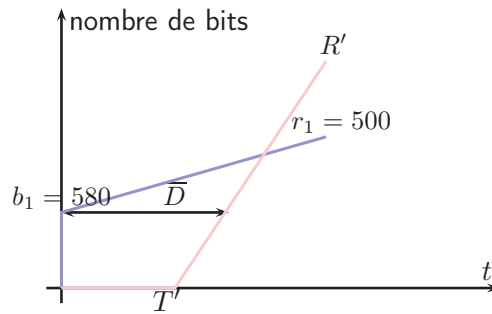


FIGURE 3.14: Courbe de service de bout en bout offert au flux considéré, courbe d'arrivée et délai de bout en bout

Comme dans la première partie, le majorant du délai correspond à la distance horizontale entre la courbe d'arrivée du flux considéré (de paramètres r_1 et b_1) et la courbe de service définie précédemment (FIGURE 3.14). Le délai peut alors s'exprimer selon la formule présentée précédemment et ainsi :

$$\bar{D} = T' + \frac{b_1}{R'} = \frac{\sum_{j=2}^n b_j + \sum_{j=2}^n r_j(\ell + t_p)}{R - \sum_{j=2}^n r_j} + \frac{b_1}{R - \sum_{j=2}^n r_j}$$

Cette équation sera appliquée à chacun des flux pour déterminer les délais bornés de chacun en fonction de ses caractéristiques. A titre d'exemple, le délai de bout en bout pour le flux AG10103 est de $977,23 \mu s$ (avec $\ell = 0,35 \mu s$, $b_1 = 580 \text{ bits}$, $R = 100 \text{ Mb/s}$, $\sum_{j=2}^n b_j = 97379 \text{ bits}$ et $\sum_{j=2}^n r_j = 352631 \text{ b/s}$).

Ces calculs sont établis lorsque chacun des flux est en concurrence avec l'ensemble des autres flux. Les délais bornés obtenus sont alors importants (de l'ordre de la milliseconde) au regard de l'architecture de commande centralisée à communication maître/esclaves (architecture pour laquelle les délais sont de l'ordre de $20 \mu s$). Cet écart vient notamment du fait que le mode de communication introduit une concurrence entre plusieurs flux ; en effet, l'avalanche de données (constituée de l'ensemble des flux) est très importante et implique un délai important. Il apparaît clairement ici que le choix de l'architecture (un seul commutateur par ensemble d'équipement) joue directement sur les valeurs de délais. En d'autres termes, les délais bornés auraient été moins importants si l'architecture avait été décentralisée.

Bien que largement supérieurs aux délais préalablement obtenus (dans le mode maître / esclaves), il est toutefois important de noter que ces délais restent acceptables eu égard aux exigences de la commande. On peut ainsi remarquer que les délais restent inférieurs à une milliseconde tandis que les périodes d'émission de message sont au minimum de 18 ms . On peut en conclure que même pour une architecture non optimale, on reste dans des performances acceptables même en considérant le pire cas, et ce pour un mode de communication non optimisé.

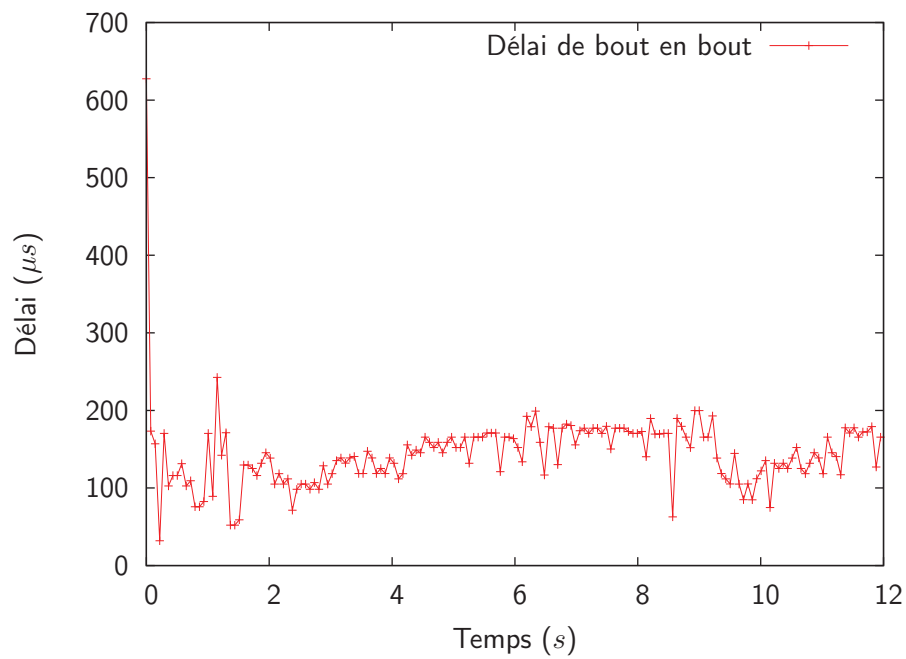
De plus, les bornes obtenues par le calcul réseau ont pu être confortées par des simulations. Bien que scientifiquement les simulations ne soient pas nécessaires (puisque les majorants sont

déterminés formellement et de manière déterministe), elles permettent néanmoins de rassurer les ingénieurs sur les majorants des délais relativement aux délais de simulation. Elles permettent également d'estimer le pessimisme moyen introduit. Par ailleurs, ce pessimisme peut être limité par une modélisation fine des commutateurs (Georges *et al.*, 2003). En outre, les simulations seront remplacées par des expérimentations (présentées dans les chapitres suivants) qui seront nécessaires pour répondre à l'exigence d'observabilité de manière à « (re)jouer au sol » .

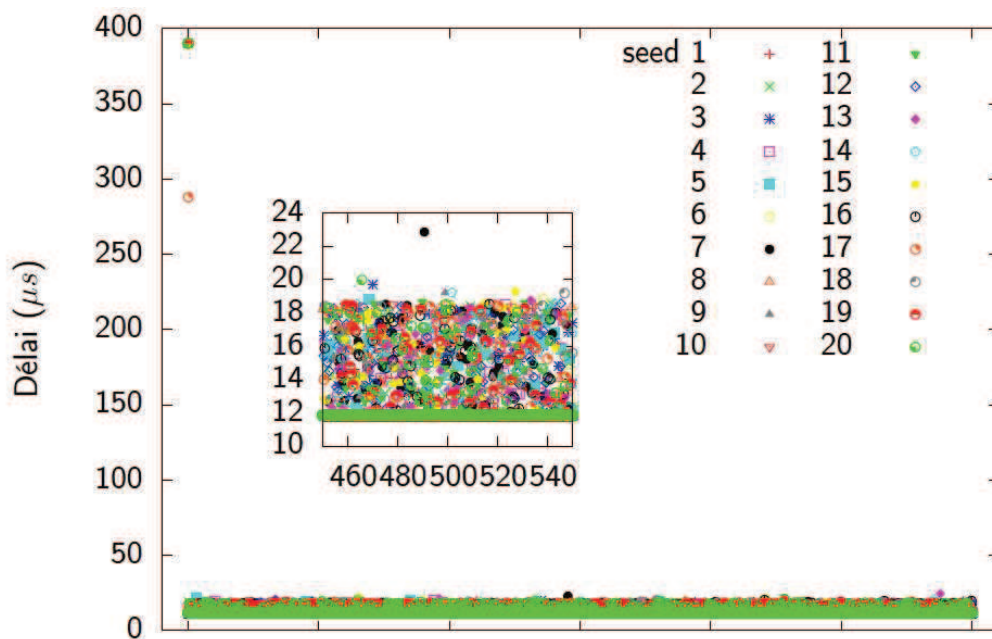
La FIGURE 3.15 correspond par exemple, à l'observation des délais de bout en bout de deux flux périodiques. Deux cas ont été considérés. Dans le premier, la génération du trafic est fondée sur la trace. La FIGURE 3.15(a) présente les résultats associés. L'objectif du second cas est de se rapprocher des hypothèses utilisées par le calcul réseau, à savoir la stricte connaissance de trafic pseudo-périodique où la gigue est simplement bornée. Les mesures présentées à la FIGURE 3.15(b) correspondent à une durée de simulation de 10 minutes. Le trafic généré présente les caractéristiques extraites dans les traces mais les dates d'inter-arrivées des paquets ne correspondent plus strictement aux dates établies par la trace. Elles sont alors tirées aléatoirement selon des caractéristiques extrapolées des spécifications et/ou de la trace. En ce qui concerne cette seconde série d'observation, 20 simulations (un *seed* correspond à un tirage aléatoire, donc à une simulation) correspondantes chacune à un tirage aléatoire spécifique ont été réalisées.

Si l'on compare les deux cas étudiés à la FIGURE 3.15, il apparaît que cette nouvelle architecture centralisée à communication type producteurs / consommateur apporte une variation des délais que l'on ne retrouvait pas dans l'architecture centralisée à communication maître/esclaves où un seul paquet était présent sur le réseau. Cette variation est d'autant plus visible sur le second graphe dans la mesure où celui-ci intègre le fait que la variation de la gigue est inconnue. Une première conclusion pour cette architecture consistera donc à affirmer qu'il n'est pas possible ici de reproduire deux simulations ou deux expérimentations identiques. Elle pourra alors amener à réfléchir à un nouveau cadre pour la vérification du bon fonctionnement du réseau.

L'intérêt du graphe 3.15(b) est de faire apparaître les cas critiques en termes de synchronisation des échanges sur le réseau, c'est à dire en termes de délais. Pour le délai, cela se concrétise par des valeurs approximativement 20 fois plus grande. Cette observation est à mettre en rapport avec la valeur précédemment obtenue *via* le calcul réseau. Par calcul réseau, nous avons obtenu des majorants allant de 963,71 μs à 977,24 μs . Ces valeurs correspondent en fait au pire cas "critique". Ce pire cas est atteint lorsque toutes les stations connectées au commutateur (soit approximativement la moitié des 42 stations présentes dans le lanceur) cherchent simultanément à émettre un message vers l'OBC. Une telle situation se retrouve de manière partielle au début de la simulation de la FIGURE 3.15(a). En effet, tous les flux sont ici calibrés pour débiter au même instant. Bien que la probabilité de recouvrer une synchronisation de toutes les stations soit minime, on note sur la FIGURE 3.15 que des cas intermédiaires ont pu être observés pour des délais allant jusqu'à 630 μs . Au final, on peut donc conclure que malgré les difficultés de



(a) A partir des traces



(b) Selon des tirages aléatoire

FIGURE 3.15: Observation des délais en simulation pour l'architecture de la FIGURE 3.13

reproduire deux expérimentations identiques, il est possible malgré tout à déterminer le pire cas avec une précision relativement acceptable. La qualité de cette précision dépendra par ailleurs de la qualité de l'identification de la commande.

Considérons à présent le cas d'une défaillance. La FIGURE 3.16 illustre la reconfiguration du

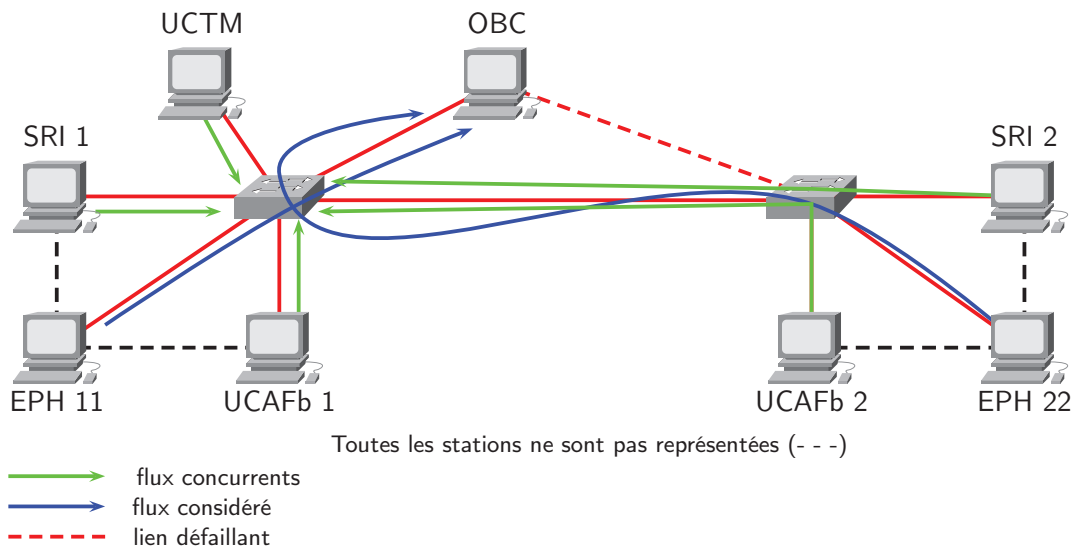
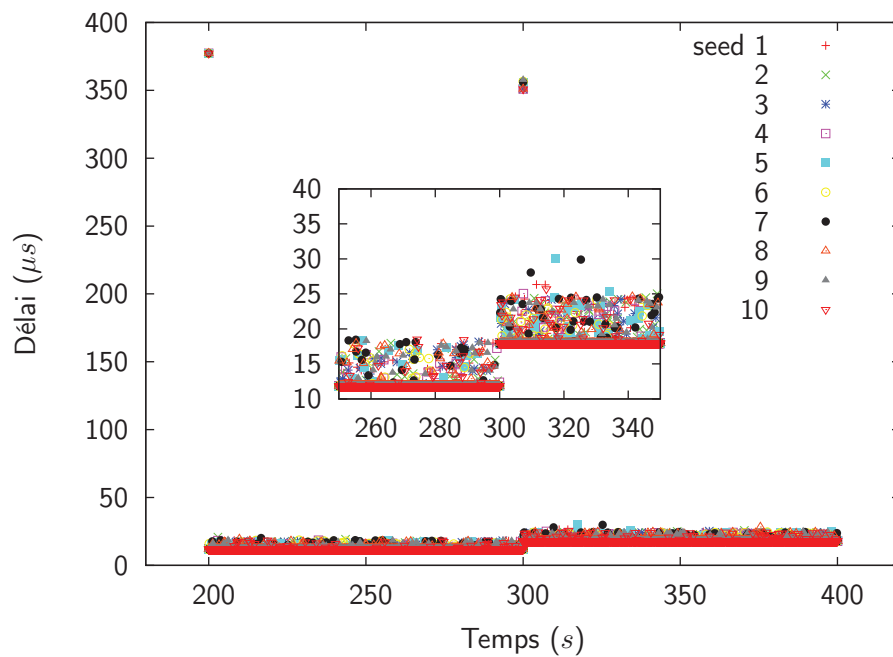


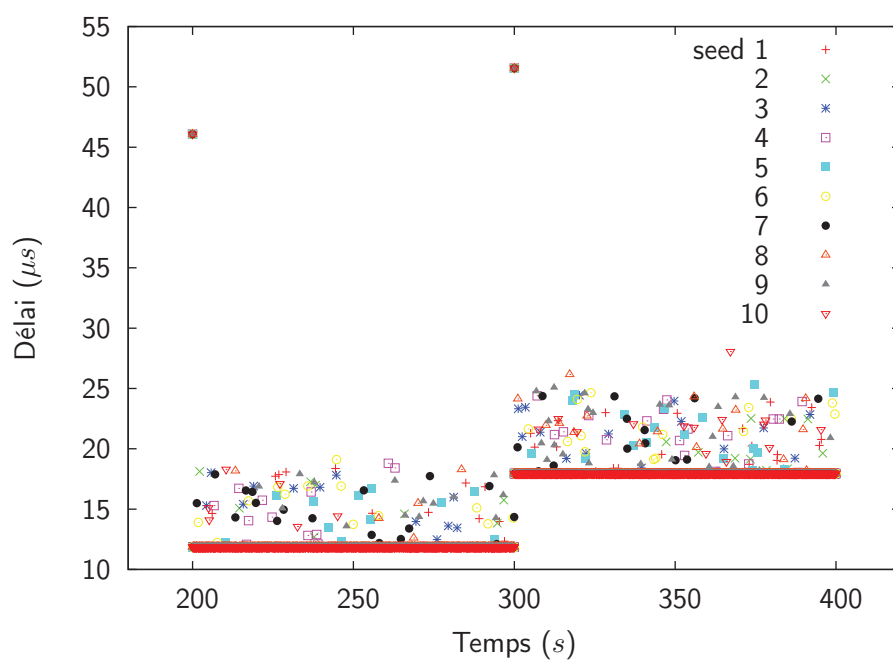
FIGURE 3.16: Exemple du flux AG10103 et du flux AG20103 lors de la défaillance d'un lien

chemin suite à une défaillance du lien redondant. Du point de vue calcul réseau, cela signifie que les calculs doivent être repris afin de prendre en compte ces nouveaux chemins (un commutateur de plus à traverser) ainsi que les nouvelles situations de multiplexage de plusieurs flux sur un même port de sortie. De nouvelles simulations ont ensuite été réalisées. Dans celles-ci, la défaillance intervient à 300 s. Les graphes 3.17 mettent en évidence une augmentation du délai suite à cette défaillance. Cette augmentation est due en grande partie à la traversée d'un commutateur supplémentaire (où il y a déjà concurrence pour l'accès au port de sortie reliant le second commutateur – de droite). Dans le premier exemple donné à la FIGURE 3.17(a), le délai au moment de la défaillance n'est pas supérieur à celui de début de simulation. Cela peut s'expliquer notamment par le non synchronisme des flux sur le commutateur 1 (de gauche), et ce malgré que l'ensemble des flux démarre *a priori* aux mêmes instants. Le temps de traversée du commutateur ainsi que le temps de propagation explique ce non synchronisme. En revanche, la situation où les flux sont en concurrence avec l'ensemble des autres flux n'est pas à exclure comme le montre la FIGURE 3.17(b). Cela s'explique notamment par le fait que la gigue est tirée aléatoirement et entraîne un re-synchronisme des flux sur le commutateur 1.

Au final, il est montré que la défaillance n'est pas sans conséquences mais que malgré tout, les délais observés restent acceptables. Notons également que les cas « critiques » sont rares même en cas de défaillance.



(a) Exemple 1



(b) Exemple 2

FIGURE 3.17: Observation des délais en simulation pour l'architecture de la FIGURE 3.13 lors de la défaillance du lien entre le commutateur 2 et l'OBC

3.3.2 Architecture distribuée à communication type producteurs/ consommateurs

Dans cette dernière étude, l'objectif est de mettre en évidence les bénéfices d'une architecture distribuée sur les délais de bout en bout. Ainsi, il est proposé de montrer cet intérêt sur une partie de l'architecture 2 sans redondance comme représenté à la FIGURE 3.18. Le mode de communication utilisé ici est de type producteurs/consommateurs.

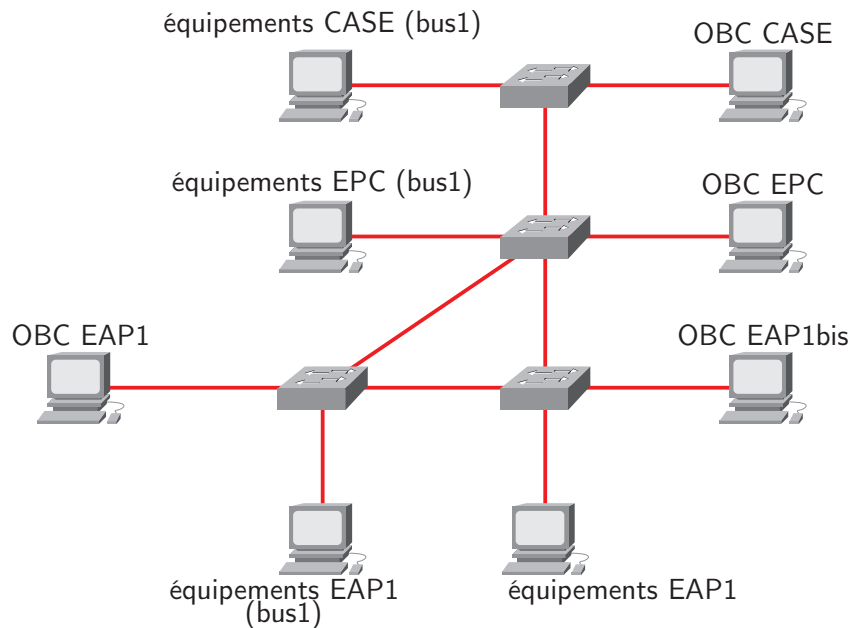


FIGURE 3.18: Architecture distribuée

A ce jour, la commande n'a pas encore été définie par le CNES. La stratégie de distribution de celle-ci est alors encore loin d'être pensée. Nous souhaitons ici sensibiliser au gain potentiel de la distribution. Ainsi pour cette étude, nous avons considéré une distribution par étage. Même si nous sommes conscients que d'autres choix (de distribution) peuvent donner des résultats différents, l'étude de la sensibilité de la répartition des stations est une perspective à ces travaux. Lorsqu'une stratégie sera effectivement définie par les experts de la commande du CNES, la méthodologie ne s'en trouvera pas pour autant disqualifiée.

Pour cette architecture, la commande est alors distribuée dans plusieurs OBC : une station type OBC par étage. Ainsi, l'ensemble des stations d'un étage communique avec « son » OBC (c'est-à-dire avec celui de l'étage) et non plus directement avec une station unique. Il s'agit ici donc d'une étude purement théorique, l'identification de la distribution devra être menée en amont.

Lorsque cette distribution est parfaitement connue, il est convenant de déterminer le trafic entre les différents OBC. Dans notre exemple, il est considéré que chacun des OBC des étages inférieurs

(à savoir EPC et les deux EAP) émet un trafic périodique de 1000 *octets/s* à destination de l'OBC (CASE) principal. Les courbes d'arrivée de ces flux particuliers relatifs aux trafics entre OBC peuvent être facilement modélisés de la même façon que les flux des autres équipements. De plus, les trafics des stations (productrices) sont identiques au cas précédent ce qui nous permet de réexploiter les mêmes courbes d'arrivée. La différence réside dans la destination. En effet, chaque producteur émet à son propre consommateur (OBC de l'étage).

Les flux de commande des différentes stations vers les différents OBC sont à différencier des flux entre OBC lors de la détermination de la courbe de service de bout en bout offert aux différents flux.

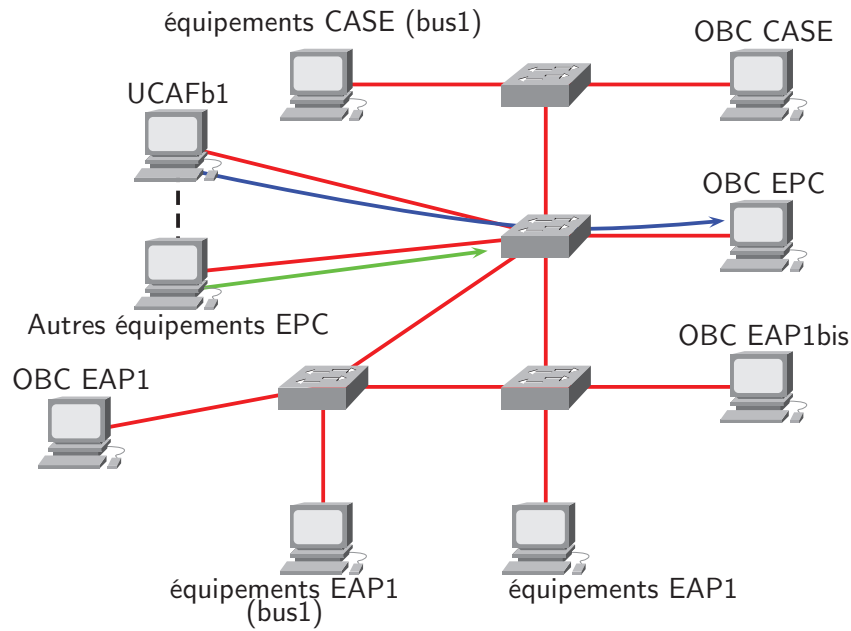


FIGURE 3.19: Cas d'un flux de commande à destination de l'OBC EPC impliquant l'application du service résiduel

Intéressons nous, tout d'abord, aux différents flux de commande à destination des différents OBC. Le cas critique considéré par le calcul réseau suggère que l'ensemble des stations d'un étage transmet simultanément un message vers l'OBC de leur étage comme indiqué à la FIGURE 3.19. Puisque plusieurs flux peuvent être en concurrence pour l'accès au port sur lequel est connecté l'OBC de l'étage considéré, et comme le nombre de saut est égal à 1, il convient d'appliquer le théorème du service résiduel. La courbe de service alors offert aux flux est :

$$\beta_1(t) = R'(t - T')^+$$

avec $R' = R - \sum_{j=2}^n r_j$ et $T' = \frac{\sum_{j=2}^n b_j + \sum_{j=2}^n r_j(\ell + t_p)}{R'}$.

Il est noter cependant que pour l'étage supérieur (CASE), les flux des différents OBC des étages inférieurs sont à prendre en considération dans les flux concurrents.

Les flux entre OBC seront alors en concurrence entre eux et avec les flux de l'étage supérieur (CASE) pour l'accès au port sur lequel est connecté l'OBC principal comme indiqué à la FIGURE 3.20. Ainsi, il conviendrait de nouveau d'appliquer le principe du *Pay Multiplexing Only Once* présenté précédemment pour définir la courbe de service offert à chacun de ces flux.

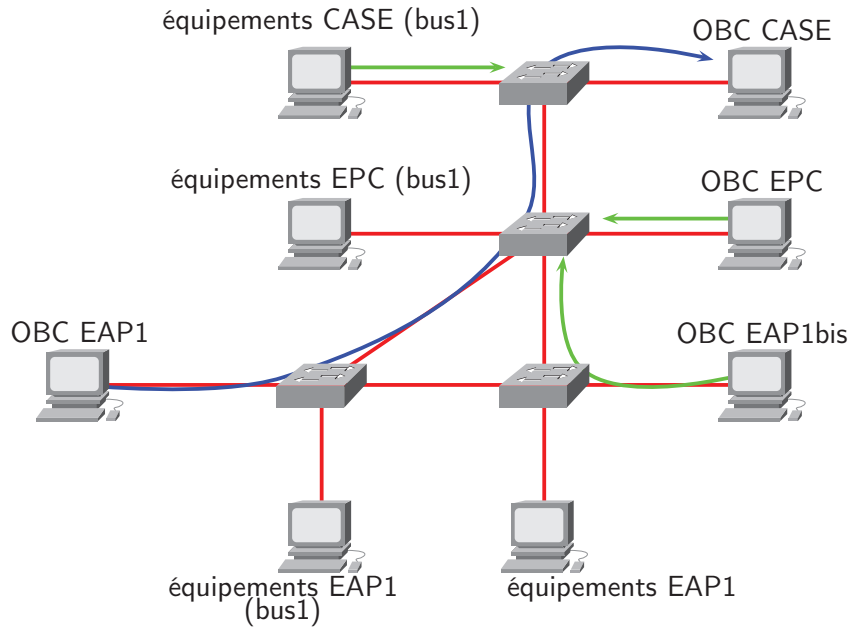


FIGURE 3.20: Cas d'un flux de l'OBC EAP1 à destination de l'OBC CASE impliquant l'application du *Pay Multiplexing Only Once*

À l'instar de l'architecture précédemment étudiée, le délai peut alors s'exprimer selon la formule présentée dans la section précédente et ainsi :

$$\bar{D} = T' + \frac{b_1}{R'} = \frac{\sum_{j=2}^n b_j + \sum_{j=2}^n r_j(\ell + t_p)}{R'} + \frac{b_1}{R - \sum_{j=2}^n r_j}$$

L'équation précédente sera alors appliquée à chacun des flux pour déterminer les délais bornés de chacun en fonction de ses caractéristiques. A titre d'exemple, le délai de bout en bout pour le flux AG10603 (émis par un équipement de l'étage EAP1) est de $52 \mu s$ (avec $\ell = 0,35 \mu s$, $b_1 = 579 \text{ bits}$, $R = 500 \text{ Mb/s}$, $\sum_{j=2}^n b_j = 5219 \text{ bits}$ et $\sum_{j=2}^n r_j = 30100 \text{ b/s}$). Il est à noter que ce flux est en concurrence avec peu d'autres flux. En revanche, un équipement de l'étage supérieur (CASE) est certainement en concurrence avec un nombre supérieur de flux et par conséquent a un délai de bout en bout supérieur (dans ce cas de figure, nous avons calculé des majorants de l'ordre de $440 \mu s$ pour cette étage).

Cependant quelque soient les flux considérés, dans cette architecture, il est important de noter que les délais « critiques » sont bien inférieurs à ceux que l'on rencontre dans une architecture centralisée à communication type producteurs/consommateur. Cette approche permet de réduire

de 2 à 15 fois les majorants des délais selon les étages et la quantité de flux par étage. En effet, chacun des flux n'est plus en concurrence avec l'ensemble des flux du lanceur mais seulement une partie (le nombre est approximativement divisé par 4) de ceux-ci.

Les stations (capteurs et actionneurs) sont réparties dans le lanceur de façon à assurer le bon fonctionnement du lanceur. Bien que d'un point de vue réseau, le déplacement d'une (ou plusieurs) station(s) peut permettre une répartition équitable de la charge par étage et donc une diminution des délais, ce degré de liberté reste réduit compte tenu du fonctionnement du lanceur et de la minimisation de la masse.

Les bornes obtenues par le calcul réseau ont également été confortées par des simulations pour les mêmes raisons évoquées précédemment.

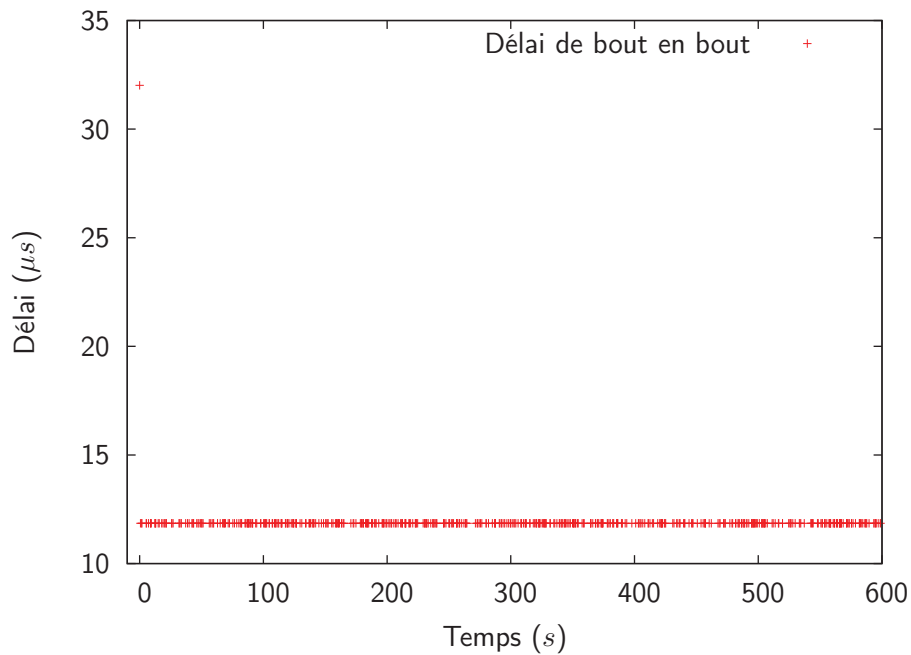
La FIGURE 3.21 correspond à l'observation des délais de bout en bout en simulation. La première figure (FIGURE 3.21(a)) se focalise sur l'observation des délais d'un flux à destination de l'OBC de l'étage. Quant à la deuxième (FIGURE 3.21(b)), elle met en évidence les délais obtenus pour un flux entre OBC : ceux-ci restent néanmoins élevés (de l'ordre de la milliseconde mais acceptables) car comme ce sont des communications inter-étages, ils vont être en concurrence avec l'ensemble des flux de chaque étage. Les mêmes observations que pour le cas précédent peuvent être faites. Ainsi, le pire cas « critique » se retrouve de nouveau au début de la simulation lorsque l'ensemble des flux sont en concurrence. Cette valeur du délai est à mettre en rapport avec la valeur obtenue par calcul réseau. Il est ainsi possible de conclure que cette architecture distribuée ouvre encore de nouvelles perspectives puisque les délais sont relativement faibles au regard des contraintes actuelles de l'application. De nouveau, on rappellera qu'il est important de bien mener l'étude de la distribution de la commande et la modélisation du système de communication pour limiter le conservatisme induit par la théorie du calcul réseau.

Néanmoins, il est nécessaire de faire attention à la manière dont la distribution sera effectuée puisqu'elle pourra introduire des réseaux cycliques. Illustrons cette problématique sur l'architecture 1. Dans le cas où la commande n'est pas distribuée (FIGURE 3.22) et que les flux ne traversent qu'un saut, le théorème du service résiduel nous permet le calcul de majorants.

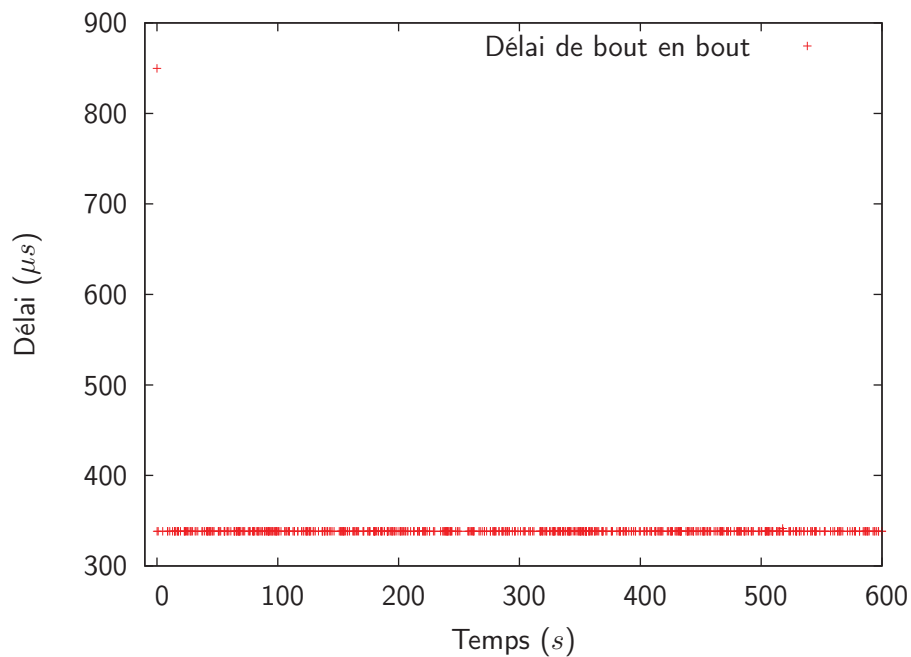
Prenons maintenant le cas où la commande est distribué sur deux OBC connectés sur un commutateur différent comme illustrée à la FIGURE 3.23. Essayons de déterminer la courbe de service offert au flux α_1 :

$$\begin{aligned}\beta'_1 &= (\beta_1 - \alpha'_2)^+ \\ \alpha'_2 &= \alpha_2 \oslash \beta''_2 \\ \beta''_2 &= (\beta_2 - (\alpha_3 + \alpha'_1))^+ \\ \alpha'_1 &= \alpha_1 \oslash \beta'_1\end{aligned}$$

Cette dernière équation met en évidence que α'_1 est fonction de β'_1 qui correspond à la courbe



(a) Délai d'un flux de commande à destination de l'OBC de l'étage



(b) Délai d'un flux entre OBC

FIGURE 3.21: Observation des délais en simulation pour l'architecture de la FIGURE 3.18

de service que l'on recherche. Par conséquent, on ne peut pas de cette manière la calculer.

Pour déterminer un majorant dans cette situation, il s'agira d'exprimer chaque rafale en tout point du réseau comme introduit dans Cruz. Cette méthode peut néanmoins conduire à un

certain sur-pessimisme et montre que la conception de la commande devra se faire en lien étroit avec la conception du réseau (co-conception).

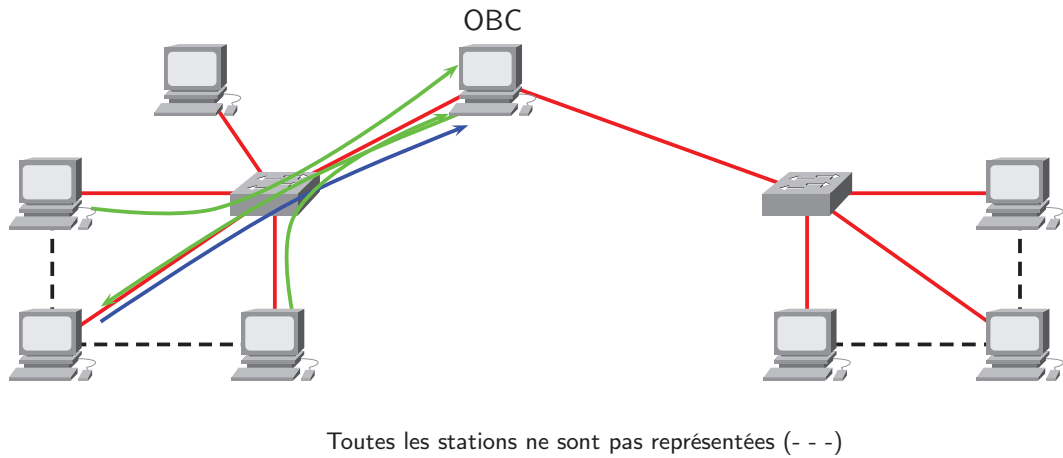


FIGURE 3.22: Exemple d'un réseau acyclique

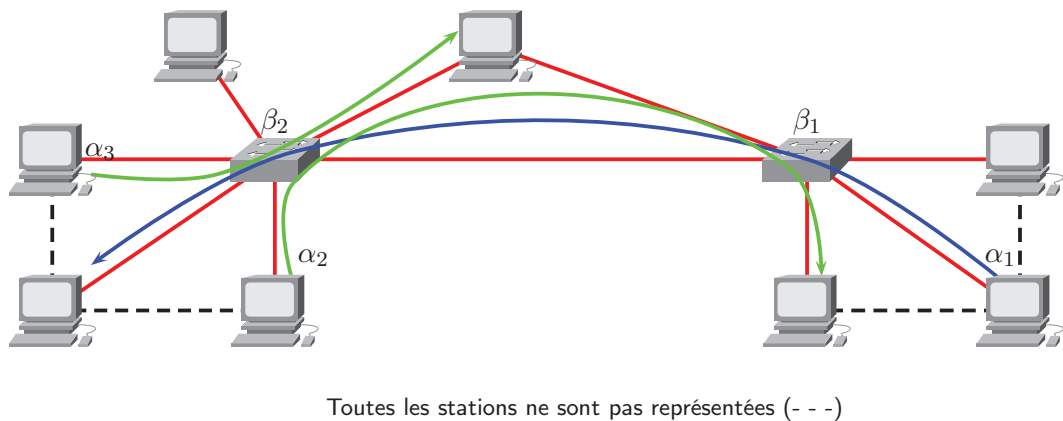


FIGURE 3.23: Exemple d'un réseau cyclique

3.4 Conclusion

Dans ce chapitre, nous avons eu recours à un mode producteur/consommateur. Le mode de communication maître/esclave avait été auparavant écarté du fait de sa non optimalité en terme de bande passante. Cela se traduit par la présence de non plus un seul mais plusieurs paquets sur le réseau qui peuvent alors être en concurrence. Par conséquent, nous avons montré qu'il est tout de même possible d'évaluer les délais de bout-en-bout grâce à la théorie du calcul réseau. Là où

l'on se retrouvait avec des délais relativement faible de l'ordre de $20 - 30 \mu s$, on se retrouve avec des délais de l'ordre de la milliseconde (somme toute acceptables en regard de l'application). En effet, la théorie du calcul réseau considère le pire cas (tous les autres flux sont considérés comme traités avant le flux étudié). Ainsi, il ne faut pas voir ces résultats comme une limitation du mode de communication producteur/consommateur mais bel et bien comme une évolution en terme de bande passante et disponibilité du réseau (par rapport au mode maître/esclave). D'autre part, bien que la distribution de la commande soit possible dans les deux architectures proposées, seule la seconde permet une distribution fine par étage. On notera que cette distribution permet la réduction des majorants des délais (relativement à la diminution des flux concurrents potentiels). Pour résumer, ce chapitre met l'accent sur le gain qu'il y a à ne plus avoir recours à un mode maître/esclave et à considérer une architecture de commande distribuée.

Par ailleurs, il est important de noter que dans la phase de conception, cette analyse des majorants des délais devra être réalisée sur l'ensemble des chemins proposés par la solution garantissant la disponibilité du réseau : à ce titre, une méthode de reconfiguration est présentée dans le chapitre suivant. En effet, l'architecture doit être robuste à la défaillance de liens de manière à satisfaire l'exigence de fiabilité (E. 5). Ainsi, comme seule l'architecture 2 propose une redondance accrue (et qu'elle est la plus évolutive en terme de distribution, modularité, ...), c'est sur cette hypothèse de redondance de liens que se basera la méthode de reconfiguration proposée dans le chapitre suivant. Notons que l'aspect temporel rentre de nouveau en compte, de manière à assurer une interruption de service la plus faible possible.

Chapitre 4

Assurer la disponibilité du système de communication

4.1 Introduction

Il a été mis en évidence dans le chapitre précédent, que la défaillance d'un lien et l'utilisation d'un autre chemin pour atteindre une destination peut amener à une augmentation des délais. Or, les garanties temporelles (majorants des délais calculés) d'une application ne doivent pas être remises en cause dans le cas d'une reconfiguration topologique en cas de défaillance de lien par exemple. Il conviendrait alors d'appliquer l'évaluation temporelle présentée précédemment sur toutes les configurations topologiques possibles et/ou acceptées. Ces différentes possibilités dépendent essentiellement de la méthode utilisée pour assurer la disponibilité du réseau.

De manière native, Ethernet s'appuie sur le protocole STP (*Spanning-Tree Protocol*, IEEE 802.1d) qui permet d'analyser la topologie afin d'inhiber un certain nombre de liens de façon à éliminer les boucles tout en assurant l'interconnexion de l'ensemble de tous les nœuds : on parle d'arbres couvrants (*spanning-tree*). De plus, STP surveille en permanence l'état des liens et des équipements. Lorsqu'une défaillance intervient (rupture d'un lien, défaillance d'un équipement, . . .), le protocole recalcule un nouvel arbre afin de maintenir la connectivité de l'ensemble des nœuds du réseau.

Dans ce cas, l'évaluation des majorants des délais devrait alors être réalisée pour chacun des arbres couvrants : la FIGURE 4.1 présente l'évolution de l'arbre couvrant en fonction de la défaillance de liens. Le nombre peut être important en fonction de la complexité de l'architecture. Outre cet aspect de garantie de délais, il est important que l'architecture soit tolérante à cette défaillance (que ce soit dans le milieu industriel, avionique ou spatial) comme énoncé dans l'exigence E. 5.

Les performances du protocole STP ne sont pas adaptées aux environnements temps-réel

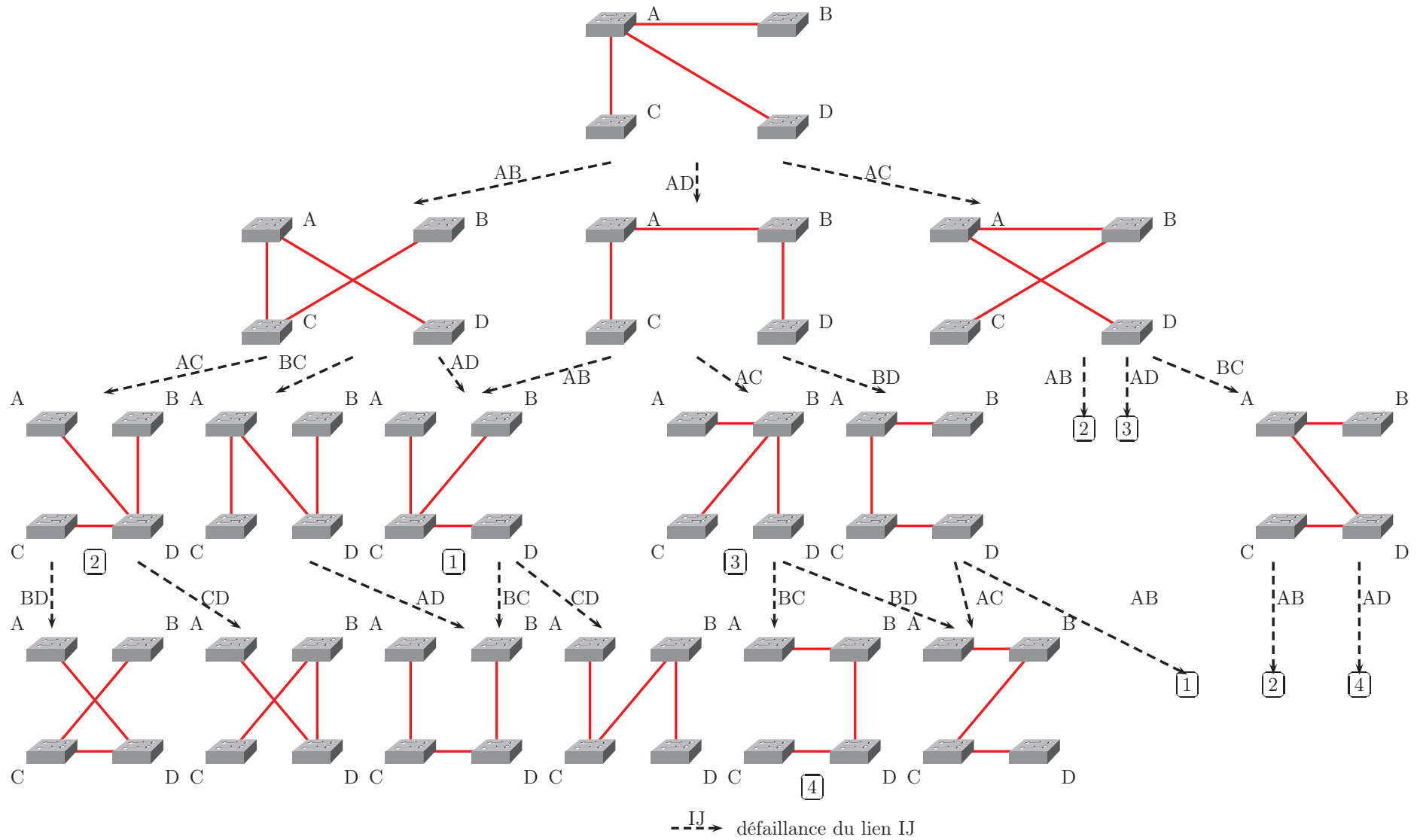


FIGURE 4.1: Evolution de l'arbre couvrant en fonction de défaillances

(plusieurs dizaines de secondes pour reconfigurer). C'est pourquoi de nombreuses propositions ont été faites pour répondre à la problématique de fiabilité / disponibilité. En effet, bien qu'il existe quelques solutions académiques (Limal *et al.*, 2007; Imtiaz *et al.*, 2010), des solutions industrielles se sont développées et standardisées (IEC, 2008). Elles se basent notamment sur l'ajout de redondance. Ces solutions peuvent alors être classées en deux catégories (Harima, 2009) : « redondance dans le réseau » (ou *in the network*) et « redondance dans les nœuds » (ou *in the nodes*) comme le montre la FIGURE 4.2. Notons que (Huynh *et al.*, 2010) propose un *survey* sur les différentes solutions académiques et industrielles.

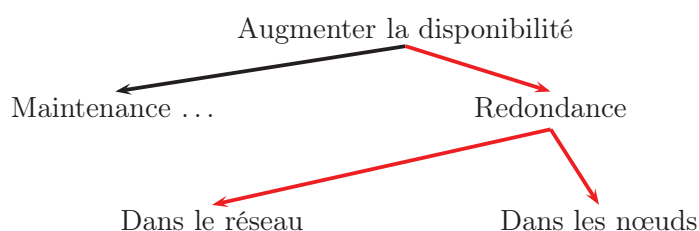


FIGURE 4.2: Taxonomie de la redondance pour les réseaux Ethernet

4.1.1 « Redondance dans le réseau »

Dans cette classe, la topologie physique du réseau est complétée par des liens et des commutateurs redondants. L'architecture du réseau sera alors (totalement) maillée ou en anneau. Les nœuds terminaux sont généralement connectés aux commutateurs par une seule interface. La redondance est alors utilisée de façon dynamique. A un instant donné, la topologie logique est totalement définie en ne considérant qu'un seul chemin entre deux équipements terminaux. Celle-ci est alors modifiée dynamiquement en fonction des défaillances pouvant se produire sur un lien ou un commutateur. C'est le cas des solutions STP (*Spanning-Tree Protocol*) (IEEE Computer Society, 1998), RSTP (*Rapid STP*) et MRP (*Media Redundancy Protocol*) (Felser, 2008) qui désactivent alors les autres chemins. Dans cette catégorie, l'objectif est de construire un réseau tolérant aux fautes. Pour assurer cette fonctionnalité, le réseau doit pouvoir détecter la défaillance et (s'auto)-reconfigurer.

La DÉTECTION doit permettre de déterminer (voire isoler) quels équipements (lien ou commutateur) sont défaillants. Cette détection peut être réalisée de façon active ou passive :

- ✓ **Les méthodes actives** : s'appuient sur la génération d'un trafic additionnel (par un ou plusieurs équipements en fonction de la distribution ou non de l'algorithme) permettant de monitorer l'état du réseau. La perte de celui-ci traduit alors une défaillance qui est dans ce cas détectée. Outre le fait que le mécanisme consomme de la bande passante, le temps de détection est important (dans STP par exemple).

✓ **Les méthodes passives** : au contraire, utilisent le mécanisme de la fonction de test d'intégrité physique de la norme 802.3 qui est communément utilisée sur les ordinateurs pour détecter l'état de connexion du câble Ethernet (parfois renseigné par une icône). Le circuit physique du *transceiver* Ethernet surveille continuellement l'activité pour vérifier si les liens fonctionnent correctement. Même lorsqu'il n'y a pas de données applicatives, une impulsion électrique est envoyée périodiquement et permet de fournir une vérification de l'intégrité des liens.

La RECONFIGURATION est réalisée par activation / désactivation des ports. L'objectif est généralement double : prévenir des boucles (pouvant entraîner des tempêtes de *broadcast*) et retrouver une topologie active (lors de défaillances). Ainsi, dans les topologies en anneau comme dans MRP, un unique port est désactivé sur un des commutateurs de cet anneau. Il sera alors réactivé dès qu'une défaillance se produira de manière à rétablir une topologie logique (linéaire). En revanche, dans le cas des architectures maillées, il peut être nécessaire d'activer / désactiver plusieurs ports sur plusieurs commutateurs. Une stratégie peut consister à construire un arbre couvrant tous les commutateurs de l'architecture comme dans (R)STP. De la même façon, un seul chemin entre deux équipements est actif.

4.1.2 « Redondance dans les nœuds »

Dans cette classe, les équipements terminaux possèdent (au moins) deux cartes de communication Ethernet connectées à deux réseaux redondants (parallèles). L'objectif principal est d'éliminer le point de défaillance unique constitué par le lien terminal (unique) en doublant l'infrastructure réseau. Il n'est cependant pas nécessaire que l'architecture soit identique sur les deux réseaux (une solution de la catégorie dans le réseau peut même y être utilisée avec le coût que cela peut entraîner). Les protocoles tels que PRP (*Parallel Redundancy Protocol*) (Kirmann *et al.*, 2007), HSR (*High availability Seamless Redundancy*) (Kirmann *et al.*, 2009), CRP (*Cross-network Redundancy Protocol*) (Huynh *et al.*, 2010), BRP (*Beacon Redundancy Protocol*) (Huynh *et al.*, 2010) relèvent de cette classe.

Deux stratégies permettent alors l'utilisation de ces deux réseaux : dupliquer les trames sur les deux réseaux disjoints ou envoyer sur un seul et utiliser le second comme réseau de secours (en cas de défaillance). Dans le premier cas, la redondance est alors utilisée statiquement alors qu'elle est utilisée dynamiquement dans le second.

4.1.3 Commentaires

Bien que des solutions comme STP ou RSTP (mettant en œuvre des arbres couvrants) sont capables de réaliser une reconfiguration de la topologie logique par recalcul d'un arbre couvrant lors d'une défaillance, le problème est que les temps de reconfiguration ne sont pas compatibles

avec des applications temps-réel. En effet, le temps durant lequel la communication est indisponible doit être le plus possible transparent pour l'application. Il correspond à une exigence temps-réel de la stratégie de reconfiguration et est qualifié par le temps de reconfiguration (cf définition dans l'exigence E. 5 dans le chapitre 2). Une application peut alors tolérer un certain temps d'interruption de service appelé *grace time* (Kirmann et Dzung, 2006). Ce temps dépend donc de l'application. Le tableau suivant illustre les différentes classes en fonction de ce temps.

TABLE 4.1: Classes d'applications (Felser, 2008)

Applications	<i>Grace Time</i>
Non critique	dizaine de secondes
Automatisme (Supervision, ...)	2 s
Temps réel (usine de papier, processus industriel, ...)	200 ms
Automatisme à temps critique (<i>motion control</i> , avionique, ...)	< 20 ms

De manière à exploiter la redondance proposée dans l'architecture 2 (qui est de type « redondance dans le réseau »), nous nous appuyons sur celle-ci pour proposer une méthode de reconfiguration en temps limité d'une architecture Ethernet commutée. Cette proposition est présentée dans la section suivante.

D'autre part, nous présenterons une méthode basée sur l'utilisation de plusieurs chemins qui permet d'augmenter la probabilité de succès concernant l'acheminement d'informations échangées entre différents équipements réseau. Pour satisfaire cet objectif, la détermination de la meilleure paire de chemin a été basée sur une fonction objectif. Cette méthode a été illustrée sur un Système Controlé en Réseau (SCR). Une discussion de l'applicabilité de cette méthode au cas des lanceurs sera alors réalisée.

4.2 Proposition centralisée pour la reconfiguration d'une architecture Ethernet en temps limité

Nous proposons d'exploiter la redondance proposée dans l'architecture 2 pour initier une solution de reconfiguration à temps limité. Pour satisfaire cet objectif, nous allons introduire un agent central capable de détecter toutes défaillances et de réaliser les actions de reconfiguration. Bien que la centralisation d'un agent soit une source majeure de défaillance (il pourra être redondé), cette solution nous permet de tester la faisabilité de notre proposition qui doit

alors satisfaire les exigences temps-réel. De plus, nous n'avons pas actuellement la possibilité d'implémenter d'algorithme dans les commutateurs utilisés (sauf à recourir à un ordinateur avec n cartes Ethernet), ce qui ne nous permet pas aujourd'hui la distribution de cette fonctionnalité (qui est clairement une perspective).

Nous présentons donc la façon dont cet agent va être capable de détecter une défaillance et de reconfigurer une topologie logique.

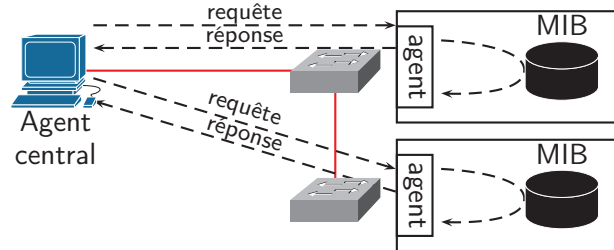


FIGURE 4.3: Interrogation périodique des agents locaux (dans les commutateurs) par notre agent central

4.2.1 Détection de la défaillance

La première étape de tout mécanisme de reconfiguration (n'utilisant pas la duplication de réseau et donc de trames) est la détection de la défaillance. Une défaillance peut être la rupture d'un lien, la perte d'un commutateur, ... entraînant une possible perte de messages (notamment s'il n'y avait pas de solution de reconfiguration). Dans le cas d'un message erroné (câble écrasé par exemple), ce n'est pas la solution proposée ici qui permettra de le détecter mais bien le protocole Ethernet qui détectera cette erreur grâce au CRC. Dans le cas d'une erreur de ce type, l'équipement (type d'interconnexion comme un commutateur ou type terminal comme le destinataire) supprimera la trame sans en avertir qui que ce soit. En effet, aucun mécanisme de retransmission au niveau de la couche 2 n'est implémenté comme il pourrait y avoir avec TCP au niveau transport.

Dans notre approche, nous proposons de centraliser les résultats de la fonction de test de l'intégrité des liens définie dans le standard 802.3. Ces informations sont accessibles dans une base de données interne au commutateur appelée MIB (*Management Base Information*). Lors d'une défaillance, le port concerné par celle-ci passe d'un état en fonctionnement (état *up*) à un état de défaillance (état *down*). Cette modification permet de détecter la défaillance et de l'isoler instantanément puisque celle-ci sera observée sur un port particulier d'un commutateur particulier.

D'un point de vue pratique, l'utilisation du protocole SNMP (*Simple Network Management Protocol* défini dans la RFC 1157) permet l'accès aux informations d'état des ports dans la MIB de chaque commutateur (FIGURE 4.3). Il existe alors deux possibilités pour les obtenir :

- visualiser périodiquement l'état des ports actifs (relatif à l'arbre couvrant) impliqués dans des commutateurs de l'architecture
- configurer des alertes informant la station de supervision d'un changement d'état d'un ou des port(s) des commutateurs de l'architecture

Le choix de l'une ou l'autre de ces solutions se fera en fonction des performances temporelles. A priori, la première solution sera retenue car elle ne dépend pas de l'équipement dont on cherche à connaître l'état de fonctionnement. De plus, les premiers tests ont montré que la solution utilisant des alertes est relativement plus longue d'un point de vue temporel (de l'ordre de la seconde). A titre de comparaison, la méthode de scrutation de l'état des ports du commutateur permet une réactivité de l'ordre de la centaine de milliseconde (cette valeur est un résultat d'expérience présenté dans la section « Expérimentations »). Le principe utilisé ici se base sur la détection électronique de la rupture de la porteuse. Une attention particulière sera nécessaire pour le choix de la période de scrutation. Ce choix aura également un impact sur la bande passante consommée.

4.2.2 Reconfiguration

Une fois la défaillance détectée, il est nécessaire de compenser l'effet de cette défaillance. Ainsi, le réseau doit réagir en reconfigurant la topologie logique.

Nous proposons de définir statiquement tous les chemins (pour atteindre chacune des destinations) dans la table de commutation. Cette table agit alors comme un filtre permettant de retransmettre un message sur un port de sortie en fonction de sa destination. Quand une défaillance est détectée, le port de sortie concerné par la défaillance, et associé à différentes destinations dans la table de commutation, doit être modifié. C'est cette modification qui permet de modifier la topologie logique.

Initialement, une topologie logique (c'est-à-dire les différents chemins pour atteindre chaque destination) a été calculée et configurée par ce biais dans la table de commutation. De manière à pouvoir les modifier lors de défaillances, les différents chemins sont pré-déterminés par un algorithme de routage comme par exemple l'algorithme de Dijkstra ou de Bellman-Ford. A la manière de (R)STP, nous nous appuyerons sur celui de Bellman-Ford.

En pratique, l'utilisation du protocole SNMP permet également de mettre à jour la table de commutation statique.

4.2.3 Formalisation

Nous considérons le graphe réseau $G = (X, A)$ où X correspond à la matrice contenant l'état de tous les ports et A à la matrice représentant la connectivité physique entre les commutateurs. Un élément de X noté $X_{i,j} \in \{0, 1\}$ représente l'état du port i avec $i \in \{1, m\}$ du commutateur $j \in \{1, n\}$. X est définie telle que $X = [X_1 \dots X_j \dots X_n]$ où X_j représente l'état des port du

commutateur j . Un élément de A noté $A_{(i,j),(i',j')} \in \{0,1\}$ représente la capacité physique de communiquer entre le port i du commutateur j et le port i' du commutateur j' .

Le graphe réseau est alors mis à jour périodiquement comme précisé à la ligne 3 de l'Algorithme 1. En effet, le monitoring de l'état des ports (la matrice X) permet la modification du graphe réseau relativement à celui-ci. Il sera alors important de montrer l'impact de cette période sur les performances de la solution.

Algorithme 1: Reconfiguration

Data : $G = (X, A_{(i,j),(i',j')})$: the network graph;
 D : the list of all stations;
 D' : the network attachment of the different stations;
Result : the vector $\langle @MAC, i \rangle$ with $@MAC$ the destination and i the output port to reach the destination $@MAC$ (next hop)

```

1 begin
2   Periodically
3      $G \leftarrow \text{update}(G)$ ;
4     foreach switch  $j$  do
5       foreach  $@MAC \in D$  do
6          $(i,j') \mid D'_{i,j'} = @MAC$  ;
7         next  $\leftarrow \text{Bellman-Ford}(j \rightarrow j', G)$ ;
8         if next  $\neq$  current then
9           current  $\leftarrow$  next ;

```

Rappelons que l'objectif est de remplir ou mettre à jour la table de commutation. Par conséquent, nous devons rechercher les ports de sortie associés à chaque destination sur chacun des commutateurs. Une liste D de tous les équipements terminaux (représentés par leur adresses MAC notée $@MAC$) est définie initialement. Une matrice D' représente la connexion de ces équipements au réseau telle que $D'_{i,j} \in D$ correspond à la connexion des équipements terminaux sur le port i du commutateur j . Par conséquent, pour chacune des entrées de la liste D , il est possible de connaître à quel commutateur j' , un équipement terminal est connecté (cf. les lignes 4-6 de l'Algorithme 1).

Cette information ainsi que le graphe réseau (mis à jour) sont alors les entrées de l'algorithme de Bellman-Ford (cf. ligne 7 de l'Algorithme 1). Le résultat de cette étape est alors le prochain saut pour atteindre cette destination (celui-ci aura été préalablement établi, cela ne se fera pas en ligne). Connaissant le commutateur à traverser et la connectivité physique (la matrice A), nous pouvons déduire le port de sortie à remplir dans la table de commutation (à la place de

celui en défaut). La modification est alors effectuée (cf. lignes 8-9 de l'Algorithme 1).

4.2.4 Expérimentations

4.2.4.1 Expérimentation préliminaire

La faisabilité de l'algorithme a été testée sur l'architecture illustrée à la FIGURE 4.4.

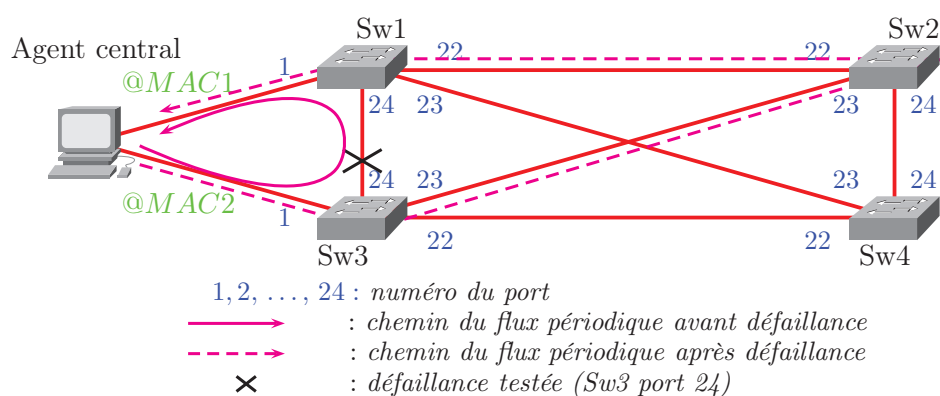


FIGURE 4.4: Architecture du test préliminaire

Pour cette expérience, les commutateurs utilisés sont des commutateurs Cisco 2950 series. Ils sont configurés pour communiquer *via* le protocole SNMP : les agents SNMP sont activés sur chacun des commutateurs et une adresse IP est définie de façon à atteindre les commutateurs (et donc les agents SNMP). De plus, les protocoles comme STP, CDP, etc sont désactivés de façon à rendre le commutateur transparent (ne générant pas de trafic inutile à l'application) et de façon à le configurer d'une manière totalement connue.

L'agent central implémente notre algorithme. Il permet également de mesurer le temps d'interruption de service relativement à la perte de trames. Pour s'affranchir de l'utilisation de protocoles de synchronisation, cet agent utilise deux cartes Ethernet et un environnement linux y est installé. Le paquet *snmpd* permet la communication SNMP avec les commutateurs. Comme pour les commutateurs, elle ne doit pas émettre de trafic non désiré. Par conséquent, la table ARP est définie préalablement aux expérimentations.

Le résultat de l'algorithme de Bellman-Ford est appliqué initialement en remplissant les tables de commutation de chacun des commutateurs. Dans ce premier exemple, nous considérons un trafic périodique envoyé par l'agent central (@MAC2) à sa seconde interface (@MAC1) comme illustré à la FIGURE 4.4. Par exemple, la table de commutation du commutateur 1 est configuré de la façon suivante :

<i>dot1dStatic- Address</i>	<i>dot1dStatic- ReceivePort</i>	<i>dot1dStatic- AllowedToGoTo</i>	<i>dot1dStatic- Status</i>
@MAC1	0	0x80	3
@MAC2	0	0x00 00 01	3

Tous les paramètres de cette table (accessible *via* SNMP) sont définis dans la MIB nommée *BRIDGE-MIB* (Decker *et al.*, 1991). Le premier paramètre (*dot1StaticAddress*) correspond à l'adresse de destination. Le second (*dot1StaticReceivePort*) permet de spécifier si désiré une règle sur le trafic d'entrée : il est possible d'autoriser le trafic entrant par tous les ports (en assignant la valeur 0) ou de spécifier un (ou plusieurs) port(s) particulier(s). La reconfiguration proposée est appliquée en modifiant l'attribut *dot1dStaticAllowedToGoTo* correspondant au port de sortie de la trame à retransmettre. Le dernier paramètre (*dot1StaticStatus*) permet de spécifier le statut de l'entrée de la table (par exemple, la valeur 3 correspond à une entrée permanente ou statique). Cette table de commutation montre que le trafic entrant (par n'importe quel port) à destination de @MAC1 sera retransmis sur le port 1 (correspondant à la valeur 0x80 dans le champ *dot1dStaticAllowedToGoTo*) en respectant la première règle (définie par la première ligne de la table). De cette façon, il est possible de définir que ce trafic traversera successivement les commutateurs 3 et 1 relativement au résultat de l'algorithme de Bellman-Ford.

Dans cette expérimentation, un trafic périodique (avec une taille de données équivalente à la taille minimale d'une trame Ethernet soit 64 *octets*) est généré. Les trames relatives à ce trafic sont estampillées et numérotées. Ces deux informations permettent le calcul de délai et d'inter-arrivée. La défaillance est réalisée en débranchant un lien (connecté au port 24 du commutateur 3 dans cet exemple comme illustré à la FIGURE 4.4) à une date aléatoire. La perte de trames a pour conséquence une augmentation de l'inter-arrivée (par rapport à la période). Ce temps correspond alors au temps d'interruption de service (temps pendant lequel aucune trame n'est reçue).

Ce temps d'interruption de service a été mesuré 10 fois (c'est-à-dire dans 10 expériences différentes) sur plusieurs périodes de scrutation : 100 μs , 1 *ms*, 10 *ms* et 100 *ms*. Les résultats obtenus sont présentés sur la FIGURE 4.5.

La FIGURE 4.5 montre que pour toute période (100 μs , 1 *ms*, 10 *ms* et 100 *ms*), le temps d'interruption est égal à environ 200 *ms*. Les variations de celui-ci s'explique par le temps de détection qui est, d'après le standard 802.3 (IEEE Computer Society, 2008) compris entre 50 *ms* et 150 *ms*. Par conséquent, il n'est pas nécessaire d'utiliser une période de scrutation inférieure à 100 *ms*. En revanche, cette valeur va *a priori* entrainer une perte de trames pour tous les flux ayant une période inférieure. L'amélioration de l'électronique embarquée future pourra potentiellement permettre de réduire le temps de détection et donc l'utilisation d'une période de scrutation plus faible.

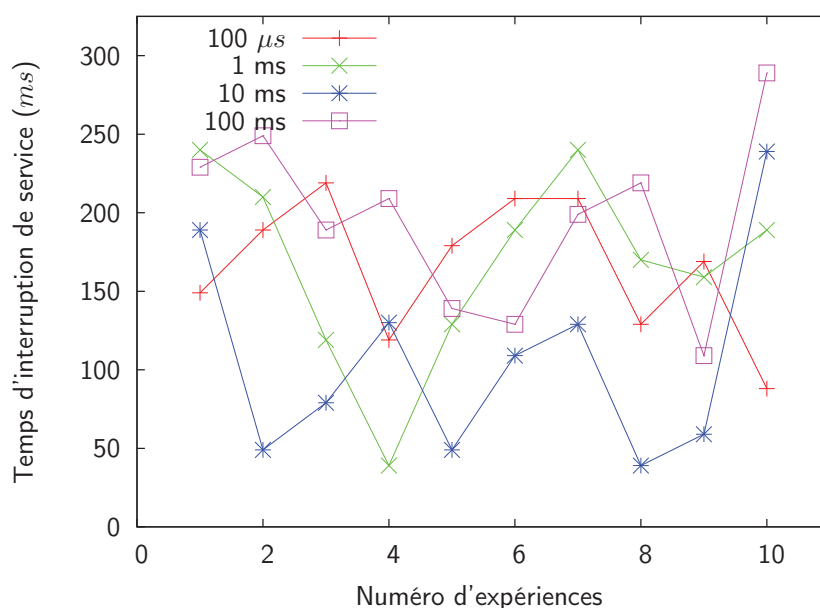


FIGURE 4.5: Temps d'interruption de service

De plus, lorsque l'ordre de reconfiguration (*via* des trames SNMP) est envoyé, nous enregistrons la date d'envoi. Celle-ci permet de déduire une approximation du temps de reconfiguration qui est la différence entre la fin de l'interruption de service et cette date. Ce temps est estimé à une vingtaine de millisecondes.

Nous pouvons conclure que le temps d'interruption de service est largement dû au temps de détection. En effet, les commutateurs utilisés n'embarquent pas une électronique adaptée à des applications temps-réel. Si le temps de détection est inférieur dans des commutateurs industriels ou si l'état d'un port est instantanément notifié, il est évident que le temps d'interruption de service sera inférieur. Néanmoins, une évaluation de la bande passante consommée et son impact sur les délais de bout-en-bout devra être réalisée (des mécanismes de Classification de Services pourront alors être utilisés).

4.2.4.2 Expérimentation sur une application lanceur

Cette section met en avant les résultats obtenus sur une partie de l'architecture 2 proposée (illustrée à la FIGURE 4.6) pour l'application lanceur.

L'objectif est d'observer l'impact de la solution de reconfiguration proposée sur le nombre de trames perdues pour chacun des flux applicatifs du lanceur mis en œuvre sur cette partie de l'architecture tout en illustrant comment est appliqué l'algorithme.

Cet exemple peut être initialement décrit par les matrices X , A , D' et le vecteur D suivants :

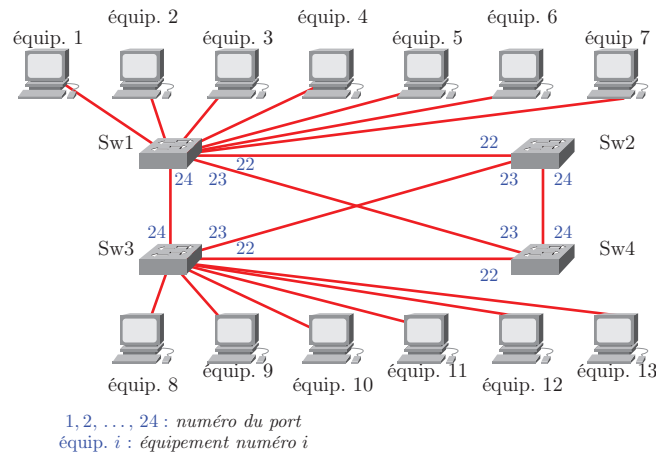


FIGURE 4.6: Architecture de test pour l'application lanceur

$$X = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ 21 \\ 22 \\ 23 \\ 24 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix} \quad D = \begin{pmatrix} @MAC1 \\ @MAC2 \\ @MAC3 \\ @MAC4 \\ @MAC5 \\ @MAC6 \\ @MAC7 \\ @MAC8 \\ @MAC9 \\ @MAC10 \\ @MAC11 \\ @MAC12 \\ @MAC13 \end{pmatrix} \quad D' = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ \vdots \\ 24 \end{matrix} & \begin{pmatrix} @MAC1 & 0 & @MAC8 & 0 \\ @MAC2 & 0 & @MAC9 & 0 \\ @MAC3 & 0 & @MAC10 & 0 \\ @MAC4 & 0 & @MAC11 & 0 \\ @MAC5 & 0 & @MAC12 & 0 \\ @MAC6 & 0 & @MAC13 & 0 \\ @MAC7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$A = \begin{matrix} & \begin{matrix} (1,1) & \dots & (22,1) & (23,1) & (24,1) & \dots & (22,2) & (23,2) & (24,2) & \dots & (22,3) & (23,3) & (24,2) & \dots & (22,4) & (23,4) & (24,4) \end{matrix} \\ \begin{matrix} (1,1) \\ \vdots \\ (22,1) \\ (23,1) \\ (24,1) \\ \vdots \\ (22,2) \\ (23,2) \\ (24,2) \\ \vdots \\ (22,3) \\ (23,3) \\ (24,3) \\ \vdots \\ (22,4) \\ (23,4) \\ (24,4) \end{matrix} & \begin{pmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Le trafic généré est décrit dans la Table 4.2.

TABLE 4.2: Trafics (la taille des données est donnée en *octets* et la période en *ms*)

Equip.	Flux 1		Flux 2		Flux 3		Flux 4		Flux 5	
	taille	période	taille	période	taille	période	taille	période	taille	période
1	-	-	-	-	-	-	-	-	-	-
2	72	12996	72	12996	-	-	-	-	-	-
3	72	288	72	72	72	144	72	144	72	1152
4	72	1152	72	1152	80	1152	72	1152	80	72
5	72	1152	94	72	72	12996	36	72	-	-
6	72	1620	72	1152	72	1152	72	1552	-	-
7	72	288	-	-	-	-	-	-	-	-
8	72	288	72	288	72	18	72	18	-	-
9	72	72	72	72	-	-	-	-	-	-
10	72	1152	72	1152	72	1152	72	1152	-	-
11	72	72	72	36	-	-	-	-	-	-
12	72	1152	72	72	72	72	72	36	72	1152
13	72	288	-	-	-	-	-	-	-	-

L'expérimentation a été menée sur trois de ces flux, à savoir le flux 1 de l'équipement 13, le flux 1 de l'équipement 9 et le flux 3 de l'équipement 8. Ils sont intéressants puisque leurs périodes d'émission sont différentes (288 *ms*, 72 *ms* et 18 *ms*). De plus, comme la destination de ces flux est l'équipement 1 et que la défaillance considérée est le lien Sw1-Sw3, un impact sera directement observé sur ces flux.

Dans cette expérimentation, la table de commutation du commutateur *Sw3* sera modifiée lors de la défaillance du lien Sw1-Sw3. Elle est initialement configurée de la façon suivante :

<i>dot1dStatic- Address</i>	<i>dot1dStatic- ReceivePort</i>	<i>dot1dStatic- AllowedToGoTo</i>	<i>dot1dStatic- Status</i>
@MAC1	0	0x00 00 01	3
⋮	⋮	⋮	⋮
@MAC7	0	0x00 00 01	3
@MAC8	0	0x80	3
@MAC9	0	0x40	3
@MAC10	0	0x20	3
@MAC11	0	0x10	3
@MAC12	0	0x08	3
@MAC13	0	0x04	3

Considérons, par exemple, le flux de l'équipement 13 et la défaillance de lien Sw1-Sw3. Comme le graphe réseau est mis à jour périodiquement (cf. lignes 2-3 de l'Algorithm 1), la défaillance sera observée dans la matrice X qui devient (une fois la défaillance électriquement détectée et reportée dans la MIB) :

$$X = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ 21 \\ 22 \\ 23 \\ 24 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

L'algorithme détermine alors (grâce aux informations pré-calculées) le prochain saut correspondant au nouveau port de sortie pour atteindre l'équipement 1. Appliquons maintenant l'algorithme : nous obtenons pour le switch $Sw3$ et l'équipement 1 (qui est la destination du flux de l'équipement 13) avec l'adresse MAC $@MAC1$ (lignes 4-5 de l'algorithme 1) :

ligne 6 : $(i,j') = (1,1)$

ligne 7 : l'algorithme de Bellman-Ford calcule le prochain saut pour atteindre le commutateur $j'=1$ depuis le commutateur $j = 3$ d'après le graphe réseau courant G . Bellman-Ford a donc le choix entre les ports 23,24 puisque la matrice donne les ports de sorties possibles relativement à la connectivité physique (i.e. ports $\{22,23,24\}$) et la matrice X donne l'état des ports de sortie (i.e. port 21 = 0 donc "up", port 23 = 1 donc "up" et port 24 = 0 donc "down"). Considérons que le résultat de l'Algorithme de Bellman-Ford est : $next = 23$.

lignes 8-9 : le vecteur $\langle @MAC1, 24 \rangle$ (avant la défaillance) devient alors le vecteur $\langle @MAC1, 23 \rangle$

Par conséquent, la table de commutation du commutateur $Sw3$ sera modifiée de la façon suivante (le paramètre $dot1dStaticAllowedToGoTo$ correspond au port de sortie devient 0x00 00 02 (i.e. le port 23) :

<i>dot1dStatic- Address</i>	<i>dot1dStatic- ReceivePort</i>	<i>dot1dStatic- AllowedToGoTo</i>	<i>dot1dStatic- Status</i>
@MAC1	0	0x00 00 02	3
⋮	⋮	⋮	⋮
@MAC7	0	0x00 00 02	3
@MAC8	0	0x80	3
@MAC9	0	0x40	3
@MAC10	0	0x20	3
@MAC11	0	0x10	3
@MAC12	0	0x08	3
@MAC13	0	0x04	3

Notons que la table de commutation du commutateur *Sw1* sera également modifiée (par l'algorithme) de telle sorte que les messages des équipements 8, 9, ..., 13 sont retransmis sur le port 22. De plus, dans cet exemple, aucune modification ne sera réalisée sur le commutateur *Sw2* puisque sa table de commutation est initialement configurée pour envoyer les messages des équipements 8, 9, ..., 13 aux équipements 1, 2, ..., 7 *via* le port de sortie 22 et les messages des équipements 1, 2, ..., 7 aux équipements 8, 9, ..., 13 *via* le port de sortie 23 (qui est la traduction du résultat initial de l'algorithme de Bellmann-Ford).

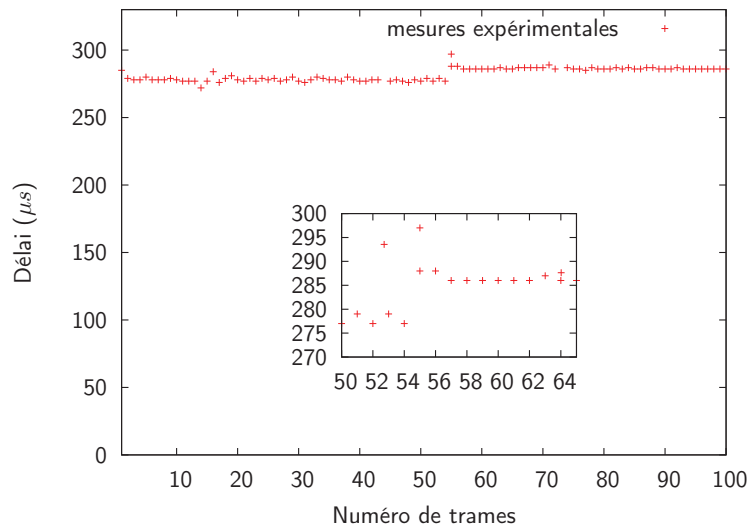


FIGURE 4.7: Délai du flux 1 sur l'équipement 13 (période : 288 ms)

Les Figures 4.7, 4.8 et 4.9 montrent l'évolution des délais de bout en bout en fonction du temps. Lors d'une défaillance (câble Sw1-Sw3 débranché), la reconfiguration entraîne un chemin plus long. Par conséquent, le délai augmente dès que la reconfiguration est effective. Dans cet exemple, cette augmentation est équivalente à la latence introduite par le commutateur (environ un temps de transmission dû au mode de commutation *store & forward*).

De plus, le zoom sur chacun des graphes permet d'estimer la perte de trames en fonction du temps d'interruption de service induit par la défaillance. Tous les flux périodiques de périodes 288 ms ne présentent aucune perte de trames. En revanche, il est observé 2 trames perdues pour les flux de périodes 72 ms et 14 trames perdues pour les flux de périodes 18 ms. Néanmoins, dans le pire cas, il pourrait être observé 1 (respectivement 3 et 15) pertes supplémentaires pour les flux de période 288 ms (respectivement 72 ms et 18 ms). Ces pertes correspondent au temps d'interruption de service mesuré dans l'expérimentation précédente. En effet, ces pertes mettent en évidence que le temps d'interruption de service varie entre 100 ms et 200 ms même avec une architecture nécessitant un nombre plus important de modification des tables de commutation.

Cette proposition s'appuie sur la reconfiguration en-ligne de l'arbre couvrant en réaction à

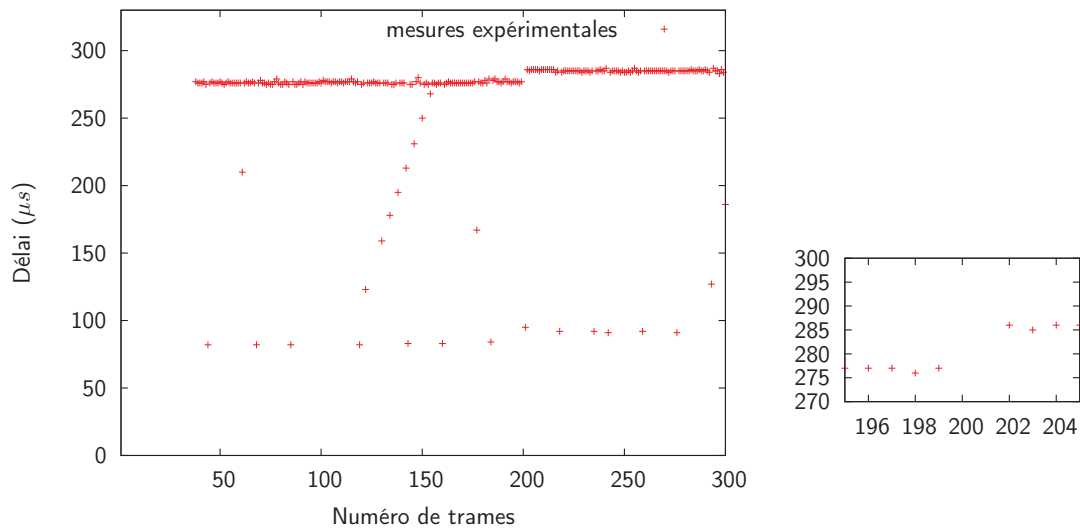


FIGURE 4.8: Délai du flux 1 sur l'équipement 9 (période : 72 ms)

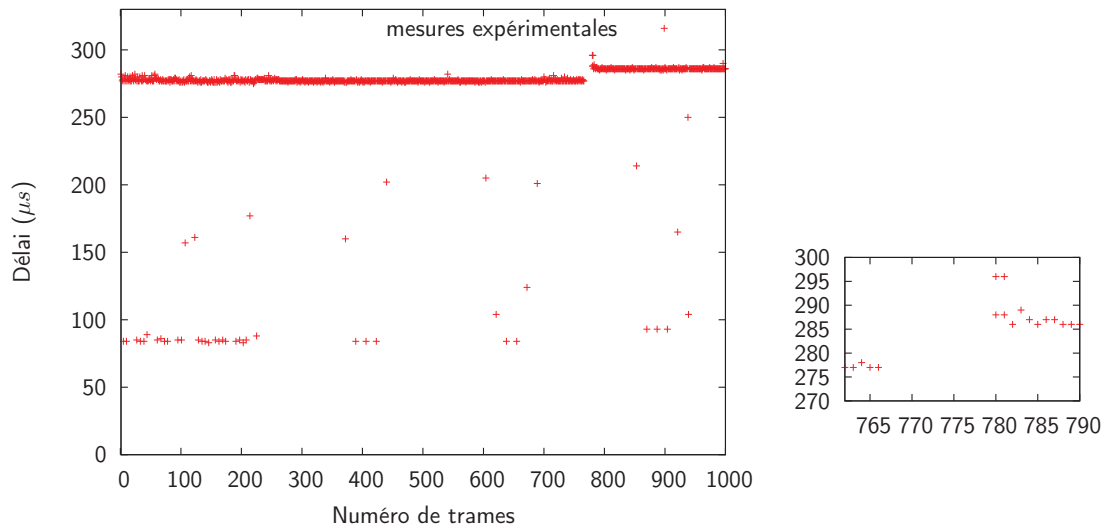


FIGURE 4.9: Délai du flux 3 sur l'équipement 8 (période : 18 ms)

une défaillance. En revanche, elle introduit un temps d'interruption non nul pouvant entraîner la perte de paquets (qui peut potentiellement être néfaste pour l'application même s'il a été vu que cette perte reste relativement acceptable). Par conséquent, la section suivante propose une méthode permettant de rendre transparente une défaillance pour l'application (sans perte de paquets).

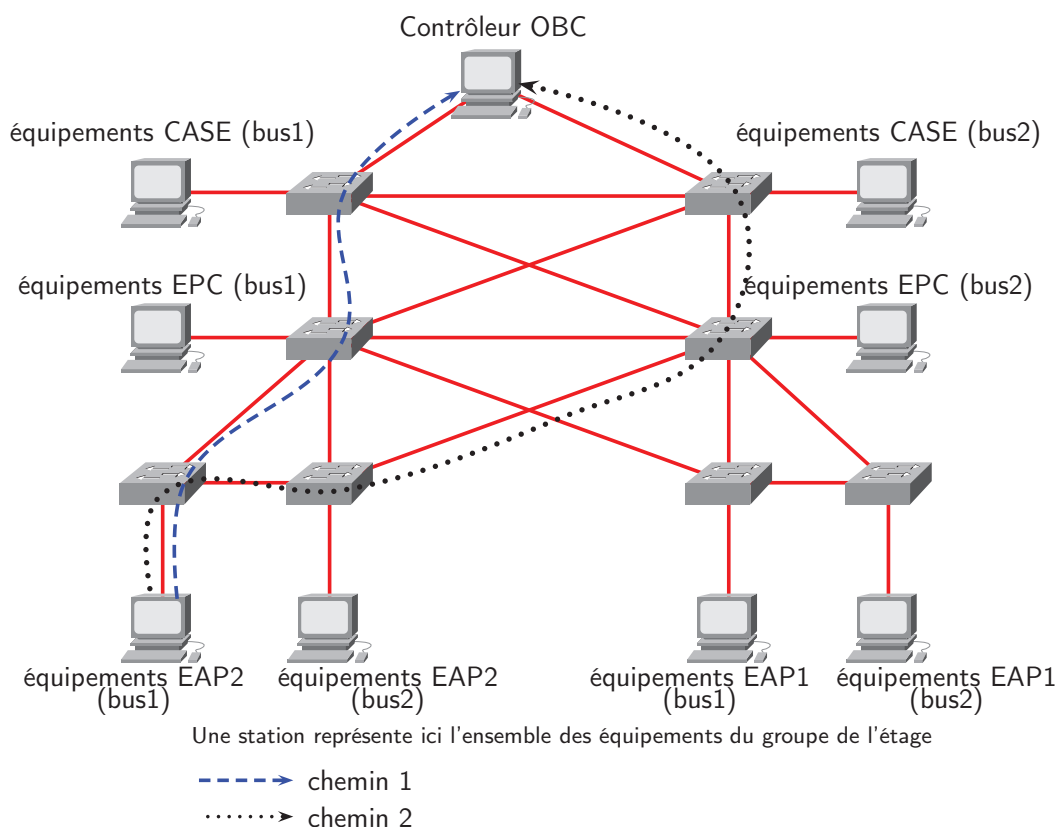


FIGURE 4.10: Exemple de chemins multiples sur l'architecture 2

4.3 Proposition d'une solution multi-chemins (sans reconfiguration topologique) (Kubler *et al.*, 2012)

4.3.1 Introduction

Cette section propose une méthode qui se base sur l'utilisation de plusieurs chemins dont la meilleure paire (au sens de la fiabilité) est alors déterminée *via* une fonction objectif. La FIGURE 4.10 présente un exemple de chemins multiples sur notre architecture 2. Dans cette approche, nous nous appuyons sur le protocole MSTP (*Multiple STP* (IEEE 802.1s) qui permet la définition d'un arbre couvrant par VLAN (*Virtual Local Area Network*) (IEEE Computer Society, 2003).

Un VLAN correspond à un domaine de diffusion rassemblant des équipements réseau indépendamment de leur localisation. Un VLAN permet donc à des stations de faire partie d'un même groupe, même si elles ne sont pas raccordées à un même réseau. Les réseaux virtuels sont utilisés principalement pour réaliser l'administration de réseaux : des règles d'accessibilité entre les équipements d'une entreprise peuvent donc être établies grâce à ce mécanisme. En effet, des équipements de VLAN différents ne peuvent pas communiquer entre eux.

Cette stratégie consiste alors à envoyer autant de trames que d'arbres couvrants définis.

4.3.2 Détermination de la fonction objectif

Le but va être de définir une expression formalisant le problème de la redondance de chemins entre les équipements d'un réseau. Un chemin est une série d'éléments réseau (lien ou commutateur) permettant de relier un équipement source (celui qui produit la donnée) à un équipement destination (celui qui utilise la donnée). La probabilité de défaillance d'un chemin dépend du nombre d'éléments réseau constituant celui-ci. Autrement dit, plus le nombre d'équipements sur un chemin est grand, plus la probabilité de défaillance de ce chemin est grande. D'autre part, si les chemins redondants partagent un même élément réseau, alors sa défaillance affectera l'ensemble de ces chemins. Par conséquent, l'évaluation de la fiabilité du réseau doit prendre en compte la longueur du chemin ainsi que le nombre d'éléments communs aux chemins redondants.

Le problème ainsi exposé va être formalisé, tout d'abord en considérant des chemins indépendants, puis dans le cas général (chemins non-indépendants).

4.3.2.1 Analyse de fiabilité pour chemins indépendants

Nous définissons notre fonction objectif comme la probabilité que le système global (le réseau) soit défaillant. Considérons, dans un premier temps, des chemins indépendants. Soit α_i le nombre d'éléments réseau composant le chemin i et λ la probabilité de défaillance par heure d'un élément réseau. $\mu = 1 - \lambda$ est alors la probabilité de non-défaillance. La probabilité de défaillance d'un chemin i est alors donnée par P_{Path_i} (équation (4.1)). La probabilité de défaillance du réseau (P_{Net_j}) composé de j chemins indépendants est donnée par l'équation (4.2).

$$P_{Path_i} = 1 - (1 - \lambda)^{\alpha_i} = 1 - \mu^{\alpha_i} \quad (4.1)$$

$$P_{Net_j} = \prod_{i=1}^j P_{Path_i} = \prod_{i=1}^j (1 - \mu^{\alpha_i}) \quad (4.2)$$

Pour réaliser l'évaluation de l'équation (4.2), nous considérons des éléments réseaux ayant une probabilité de défaillance de $\lambda = 10^{-8}$. L'objectif est d'observer le respect de la contrainte de probabilité du réseau de 10^{-8} relativement à l'influence du nombre de chemins et du nombre d'éléments réseau constituant chaque chemin.

Les résultats sont donnés à la FIGURE 4.11. La fiabilité du réseau est rapidement dégradée avec un chemin unique (ne respectant plus la contrainte de probabilité de défaillance du réseau définie précédemment), contrairement aux réseaux bénéficiant de plusieurs chemins indépendants. Plus le nombre de chemins utilisés est grand, plus la probabilité de défaillance du réseau est faible. Cependant, relativement à notre objectif de respect de la contrainte d'une probabilité du réseau de 10^{-8} , l'utilisation de deux chemins indépendants est suffisant (cela permet également de ne pas augmenter considérablement le coût). Par conséquent, l'étude des chemins partageant des éléments réseau communs (donc non-indépendants) se basera sur deux chemins.

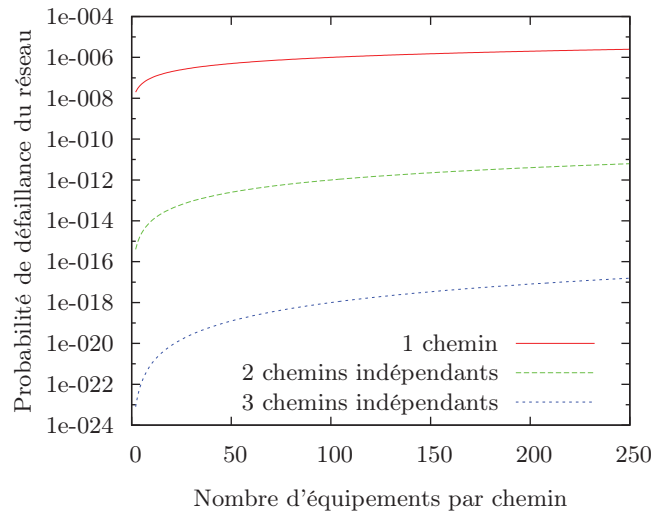


FIGURE 4.11: Analyse de fiabilité de chemins indépendants

4.3.2.2 Analyse de fiabilité pour chemins non-indépendants

Un réseau est alors composé, dans notre étude, de deux chemins pour lesquels une partie peut être commune. Considérons α_i le nombre d'éléments réseau exclusifs (donc non communs) composant le chemin i et β le nombre d'éléments réseau communs. La longueur totale d'un chemin i est donc égale à $\alpha_i + \beta$.

En s'appuyant sur l'équation (4.2), la probabilité de défaillance peut alors être déterminée par l'équation (4.3) qui constituera notre fonction objectif.

$$1 - (1 - (1 - \mu^{\alpha_1})(1 - \mu^{\alpha_2}))\mu^\beta \quad (4.3)$$

La FIGURE 4.12 présente les probabilités de défaillance définies par l'équation (4.3) relativement à la longueur du chemin et du taux d'éléments communs ($\beta/(\alpha_i + \beta)$). Cette figure met en évidence que lorsqu'au moins un élément réseau commun aux deux chemins est défaillant, le réseau est profondément affecté (sa probabilité de défaillance augmente considérablement) et ne respecte plus la contrainte de probabilité de défaillance du réseau définie précédemment (10^{-8}).

Ainsi, de manière à respecter cette contrainte, il convient donc de rechercher la meilleure paire de chemins (d'arbres) minimisant la fonction objectif. Cette recherche peut s'effectuer dans un espace de solutions relativement important. En effet, si l'on considère un graphe complet où $L = \frac{N(N-1)}{2}$ avec $N \geq 3$, l'espace de recherche d'un arbre est égal à N^{N-2} comme détaillé dans (Ali et Narasimhan, 1993). L'espace de recherche de notre problème (qui consiste en l'exploration de paire d'arbres) peut alors être égal à $C_{N^{N-2}}^2 = \frac{N^{N-2}(N^{N-2}-1)}{2}$: c'est clairement un problème combinatoire. Ce problème est une sous-classe de celui étudié dans (Jayavelu *et al.*, 2009) qui est

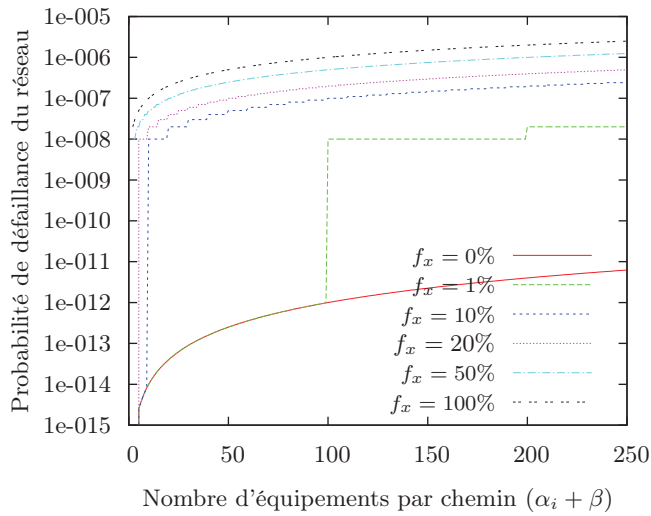


FIGURE 4.12: Probabilité de défaillance pour deux chemins composés de $f_x = \frac{\beta}{\alpha_i + \beta}$ éléments communs

alors décrit avec une complexité $O(|L||N|)$. Ainsi, comme (Jayavelu *et al.*, 2009), nous utiliserons des heuristiques.

4.3.3 Algorithmes d'optimisation

4.3.3.1 Introduction

Il existe trois classes principales d'heuristiques, à savoir, les méthodes constructives (algorithmes glouton, méthode Pilote), les méthodes à recherche locale (recuit simulé, recherche Tabou) et les méthodes considérées comme évolutionnaires (algorithmes génétiques, systèmes de fourmis). Il s'avère difficile de faire une comparaison entre ces méthodes. L'évaluation de méthodes heuristiques repose essentiellement sur des analyses statistiques sur les résultats obtenus après avoir appliqué plusieurs fois de suite ces méthodes. (Dréo *et al.*, 2006) établit une comparaison entre les méthodes du recuit simulé, la recherche de type tabou, les colonies de fourmis et les algorithmes évolutionnaires (AG). Cette comparaison repose sur des diagrammes donnant la probabilité qu'une méthode soit meilleure que l'autre. On constate dans ces travaux que pour un même problème donné, les AG donnent les meilleurs résultats. C'est cette heuristique qui a donc été retenue dans notre étude.

4.3.3.2 Codage des éléments du réseau

Les architectures réseau sont définies par un graphe $\mathcal{G} = (\mathcal{S}, \mathcal{M})$ (orientées ou non orientées) où \mathcal{S} représente l'ensemble des nœuds (les sommets d'un graphe) et \mathcal{M} la matrice d'incidence sommets-arêtes définissant les liens réseaux (les arêtes). Elle est de dimension $m \times n$ avec m le



FIGURE 4.13: Graphe et matrice adjacente : (a) graphe, (b) matrice d'incidence, (c) vecteur (nœuds terminaux)

nombre de sommets et n le nombre d'arêtes. Chaque arête est numérotée tel que pour une arête i reliant deux sommets k et l , $\mathcal{M}(j, i) = 1 \quad \forall j = k, l$ et $\mathcal{M}(j, i) = 0$ sinon. La FIGURE 4.13 illustre le codage mis en place : la matrice d'incidence est alors réduite en un vecteur où chaque ligne correspond à une arête de \mathcal{G} et la colonne aux sommets reliés par cette arête.

Il existe plusieurs méthodes de codage d'arbres couvrants (Enacheanu *et al.*, 2006; Nara *et al.*, 1992; Zhu et Tomsovic, 2002; Radha *et al.*, 2004; Ramos *et al.*, 2005; Lin *et al.*, 2000; Hong et Ho, 2005). (Enacheanu *et al.*, 2008) présente la méthode de codage basée sur la représentation des co-arbres (c'est-à-dire le codage numérique des branches ouvertes conçu par (Radha *et al.*, 2004)). En effet, lors de la construction d'un arbre, il s'agit de rejeter plusieurs liens de manière à éliminer les boucles. Ce type de codage ne s'intéresse donc qu'aux liens rejetés ce qui a pour principal intérêt, la réduction de sa taille par rapport à un codage qui chercherait à représenter toutes les branches par un '1' ou '0' selon qu'elles sont ouvertes ou non. Il reste cependant le problème de sélection des branches à ouvrir. Puisque l'objectif de la construction de l'arbre est d'éliminer les boucles, il s'agit d'analyser chaque boucle fondamentale (Lin *et al.*, 2000). Une boucle est dite fondamentale lorsqu'elle ne contient pas elle-même d'autres boucles. La méthode consiste alors à ouvrir une boucle fondamentale tout en s'assurant que cette ouverture n'entraîne pas simultanément l'ouverture d'une autre boucle fondamentale.

La FIGURE 4.14 illustre un exemple de codage numérique de branches ouvertes. Les boucles fondamentales de ce graphe sont composées des branches suivantes : 0-1-2-3 pour la première boucle, 3-4-5-6-7-8 pour la seconde et 8-9-10-11 pour la troisième. Par exemple, il est possible d'ouvrir les branches 1, 8 et 6 ou encore 0, 3 et 10, ce qui permet d'obtenir deux séquences représentant deux arbres couvrants différents.

A partir de la population d'arbres couvrants, il est nécessaire d'explorer les différents graphes de manière à déterminer le (ou les) chemin(s) entre deux nœuds d'intérêt. Il existe deux types d'algorithmes pour parcourir un graphe, le parcours en profondeur et le parcours en largeur. La différence est donnée par l'ordre de parcours des sommets. Le parcours en profondeur explore « à fond » les chemins un par un, alors que le parcours en largeur génère un arbre de recherche

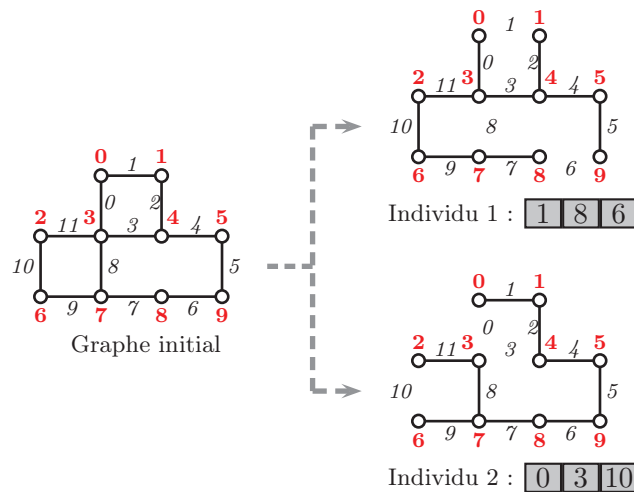


FIGURE 4.14: Codage de deux membres de la population selon le codage numérique de branches ouvertes

se faisant couche par couche. L'un ou l'autre de ces algorithmes pourrait être choisi en fonction des caractéristiques topologiques des réseaux que l'on veut analyser. Cependant le parcours en largeur nécessite beaucoup de mémoire afin de stocker toutes les alternatives à toutes les couches. Dans notre étude, l'algorithme en profondeur est mis en œuvre.

4.3.3.3 Algorithme Génétique (AG)

C'est à partir de l'observation de l'évolution des espèces que C. Darwin initie la théorie de l'évolution. La figure 4.15 illustre le principe de la sélection naturelle. Dans notre étude l'individu sera représentatif d'un arbre couvrant, dont les gènes composant le chromosome correspondront aux branches ouvertes. On remarque qu'en fonction du type de codage utilisé, les opérateurs de croisement et mutation pourront générer (avec un facteur plus ou moins élevé) des individus non valides par rapport aux contraintes topologiques. L'auteur (Enacheanu *et al.*, 2008) explique qu'en prenant la méthode de codage basée sur la représentation en co-arbre (codage numérique des branches ouvertes), on évite la détermination des boucles fondamentales permettant de tester la validité des topologies résultantes des processus de croisement-mutation. En effet, il a développé une stratégie de croisement et de mutation reposant sur des approches de la théorie des graphes.

La technique de mutation que l'auteur (Enacheanu *et al.*, 2008) a mis en place, consiste à tirer au hasard un gène d'un chromosome (représentatif d'une branche ouverte), puis à déterminer la boucle formée suite à la fermeture de cette branche. Enfin on choisit aléatoirement au sein de cette boucle une branche à ouvrir. Ce principe est illustré par la figure 4.16. La technique de croisement illustrée par la figure 4.17, consiste quant à elle à tirer au hasard un point de référence

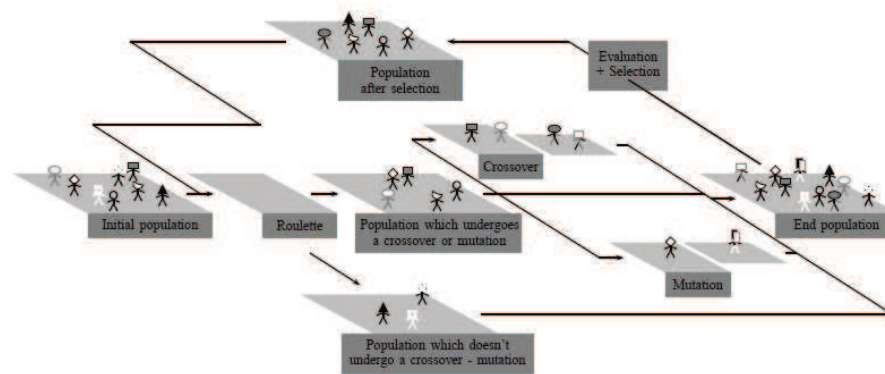


FIGURE 4.15: Principe général de la sélection naturelle

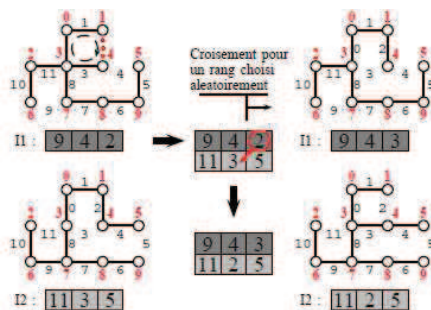


FIGURE 4.16: Technique de mutation

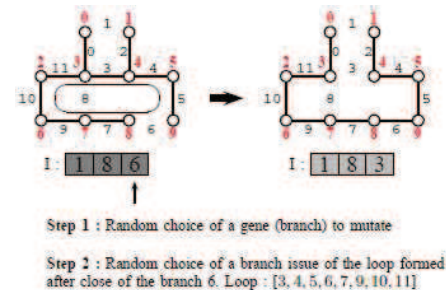


FIGURE 4.17: Technique de croisement

dans le chromosome de l'individu I1, puis déterminer la boucle formée à la suite de la fermeture de cette branche. Et enfin regarder quelles branches seraient susceptibles d'être échangées entre les 2 co-arbres. Pour cela, il faut analyser le chromosome de l'individu I2. S'il y a des gènes appartenant à la boucle, il faut alors effectuer le croisement avec l'un d'entre eux. Il faut réitérer cette démarche jusqu'à atteindre le dernier gène.

4.3.3.4 Ajustement des paramètres de l'AG

(Goldberg, 1989) a défini des paramètres conventionnels pour les AG tels que la taille de la population (t_{pop}), la probabilité de croisement (p_c) et de mutation (p_m). Ces paramètres influent beaucoup sur les performances de l'algorithme, c'est pourquoi l'étude de l'influence de chacun d'eux afin de déterminer leur valeur donnant les performances optimales, se révèle importante. (Krommenacker, 2002) met en place une démarche permettant d'effectuer ces réglages, que nous repreneons ici.

Le choix d'une trop petite population nous ferait probablement évoluer vers un minimum local. Par ailleurs, une population trop grande augmentera le temps de calcul, il nous faut donc

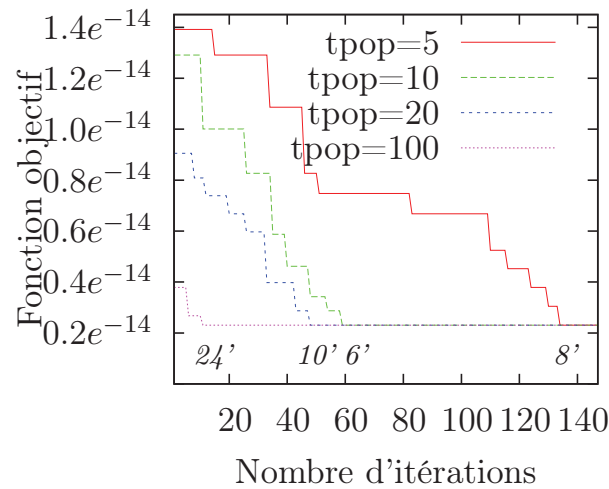


FIGURE 4.18: Ajustement du paramètre tpop

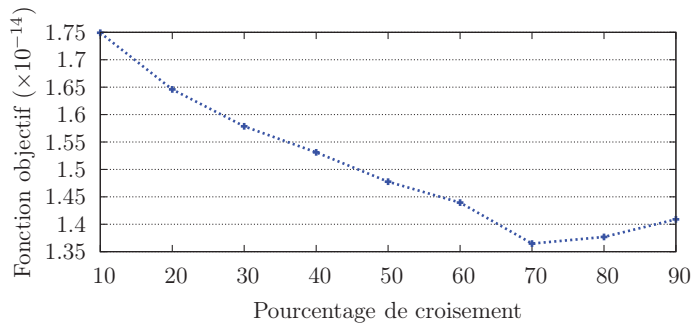


FIGURE 4.19: Ajustement du paramètre Pc

trouver le meilleur compromis. L'influence de tpop sur la qualité de la solution et le temps de calcul est représenté par 4.18. Nous avons travaillé sur un réseau composé de 100 nœuds, 240 branches et testé différentes valeurs de tpop (5, 10, 25, 60). Pour chaque valeur du paramètre tpop, nous avons tracé l'évolution de la valeur optimale de la fonction objectif en fonction du nombre de générations. Nous avons également fait figurer le temps mis dans chaque cas pour atteindre l'optimum commun. Une taille de population de 100 permet d'obtenir en très peu de générations la solution optimale, cependant le temps de calcul devient vite prohibitif (24'22). Une population de 20 permet de diviser par 2,5 ce temps de calcul. Par ailleurs, une taille de population de 5 montre que l'algorithme converge vers la solution optimale avec 11 fois plus de générations que dans le premier cas, mais réduit le temps de calcul par 3. Néanmoins, le meilleur compromis semble avoir lieu pour une taille de population de 10.

L'opérateur de croisement a pour but d'enrichir la diversité de la population. Plus le taux de croisement pc sera élevé, plus il y aura de nouvelles structures dans la population. Afin de fixer la valeur du taux de croisement, nous avons étudié l'influence de cette probabilité de croisement

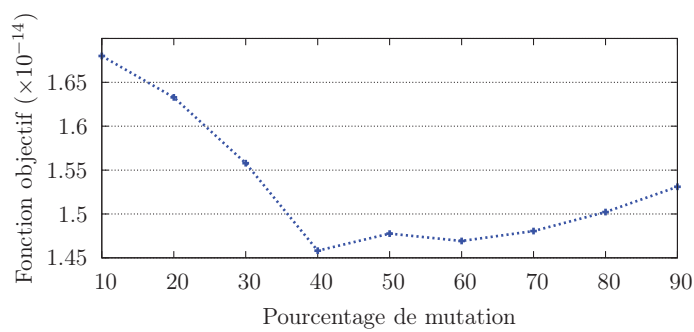


FIGURE 4.20: Ajustement du paramètre Pm

pour une série de 100 réseaux de taille fixe (100 nœuds, 240 branches). L'algorithme génétique a été exécuté pour p_c variant de 10 à 90, avec les autres paramètres fixés à : $t_{pop}=10$, $p_m=40\%$ et ce sur 150 itérations. Nous avons ainsi relevé pour chaque valeur du taux de croisement, la valeur optimale de la fonction objectif. La figure 4.19 illustre la courbe de tendance obtenue à partir des valeurs moyennes sur les 100 graphes. On constate qu'un taux de croisement trop faible ($p_c=10\%$) ralentit fortement l'évolution de l'algorithme. Par ailleurs, le tracé de $p_c=90\%$ montre qu'un taux trop élevé ne permet pas d'exploiter les individus potentiellement "bons". Dans notre cas, un taux de 70% permet d'obtenir la meilleure solution.

La diversification est également introduite par l'opérateur de mutation. Cet opérateur a pour but de libérer les situations de blocage dans des minima locaux de la fonction objectif. De même que précédemment, nous avons étudié l'influence de cette probabilité de mutation pour une série de 100 réseaux de taille fixe. L'AG a été exécuté pour p_m variant de 10 à 90, avec les autres paramètres fixés à : $t_{pop} = 10$, $p_c = 70\%$. La figure 4.20 représente la courbe de tendance obtenue à partir des valeurs moyennes des 100 graphes. Dans notre cas, un taux de mutation de 40% semble être le meilleur choix.

Pour résumer, nous avons fixé les paramètres de l'AG comme suit : $t_{pop} = 10$, $p_c = 70\%$ et $p_m = 40\%$. La section suivante présente les expérimentations menées sur un Système Contrôlé en Réseau (SCR).

4.3.4 Expérimentations sur un Système Contrôlé en Réseau (SCR)

4.3.4.1 Introduction

L'objectif de cette expérimentation est de montrer en quoi notre solution (avec redondance) est intéressante pour la commande d'un système comparativement à une solution sans redondance. Or, la commande n'est actuellement pas définie dans le cas du futur lanceur. De manière à l'illustrer, nous nous sommes appuyé sur une application académique tel qu'un Système Contrôlé en Réseau (SCR) implémenté sous Opnet.

La plate-forme réseau étudiée est illustrée par la figure 4.21. Elle est composée de 31 nœuds

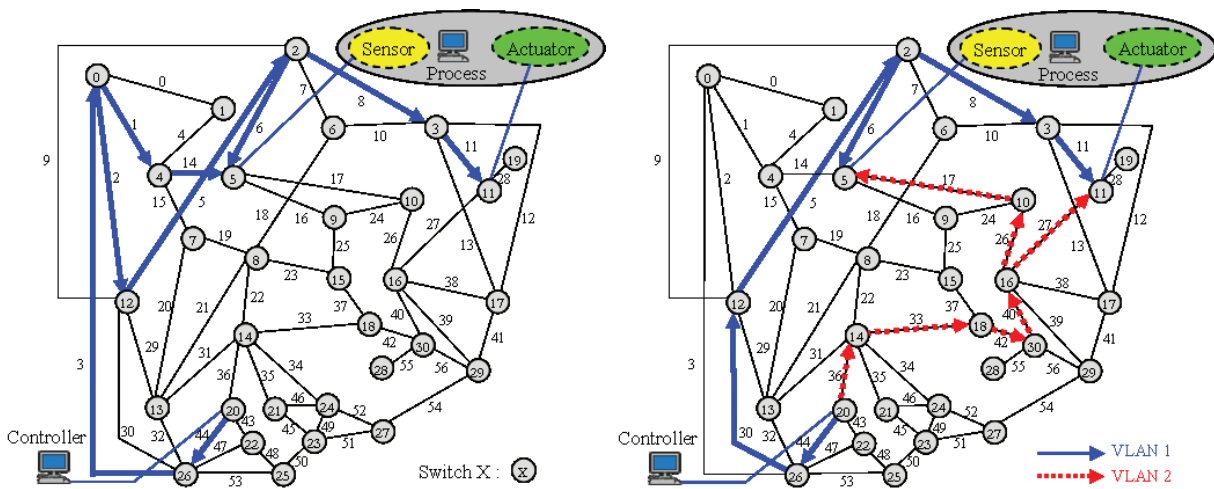


FIGURE 4.21: Réseau étudié (sans redondance de chemins) FIGURE 4.22: Réseau étudié (avec notre solution de redondance de chemins)

et 57 branches. Les équipements du SCR (contrôleur, capteur, actionneur) sont respectivement reliés aux commutateurs 20, 5 et 11. Le temps de cycle automate est de 2 ms, et les équipements envoient des paquets de 64 octets toutes les 1 ms.

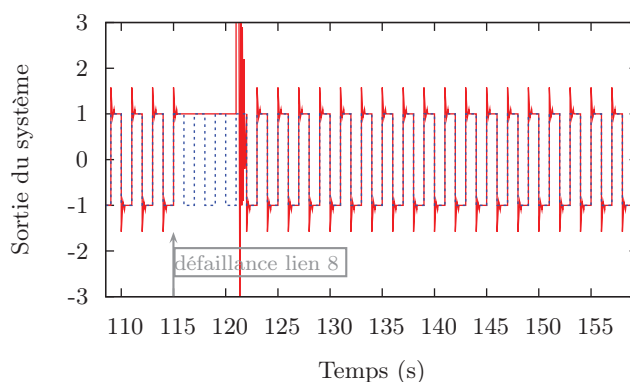
Cette étude a pour objectif de montrer les avantages de notre solution par rapport à une solution mono-chemin. Ainsi, nous présentons ici deux expériences : une sans redondance de chemins et une avec notre solution.

4.3.4.2 Solution classique (sans redondance de chemins)

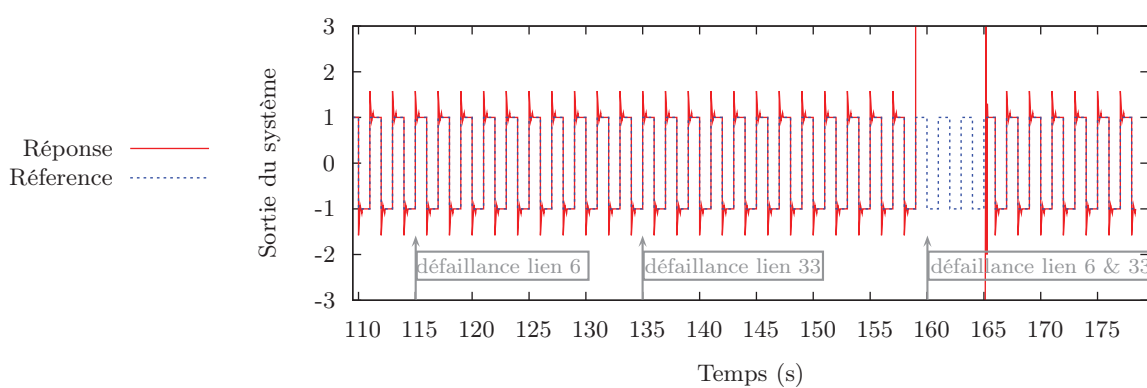
La figure 4.21 illustre le chemin issu de l'arbre couvrant défini par défaut par STP (seul le chemin entre les équipements du SCR est représenté par souci de lisibilité). Nous procédons à $t=115$ s à une défaillance du lien 8. La FIGURE 4.23(a) montre que la réponse du système qui devient alors instable du fait des informations non reçues par l'actionneur. Le SCR redevient stable lorsque le réseau a terminé la reconfiguration d'un nouvel arbre, c'est-à-dire à $t=120$ s.

4.3.4.3 Solution multi-VLAN

La meilleure paire de chemins entre le capteur, l'actionneur et le contrôleur obtenue par la méthode présentée précédemment, est mise en œuvre dans notre exemple (cf FIGURE 4.22). Nous procédons dans un premier temps à une défaillance sur le lien 6 à $t=115$ s. Le message envoyé par le capteur sur le *VLAN1* ne parvient donc pas au contrôleur. La FIGURE 4.23(b) montre que le système n'est pas perturbé par cette défaillance ; en effet, l'acheminement de ces mêmes messages est réalisé par le *VLAN2*. Nous rétablissons le lien 6 de manière à rétablir les arbres initiaux. Puis, nous procédons alors à $t=135$ s à une rupture du lien 33 (sur le *VLAN2*). De la même façon, le système reste stable puisque cette fois, les messages sont acheminés par le



(a) Sortie du système (sans redondance de chemins)



(b) Sortie du système (avec notre solution de redondance de chemins)

FIGURE 4.23: Simulation du Système Controlé en Réseau

VLAN1. Enfin, nous réalisons une rupture des liens 6 (*VLAN1*) et 33 (*VLAN2*) à $t=160$ s. Dans ce cas de figure (une défaillance sur chaque *VLAN*), le système devient instable puisqu'aucune des informations ne peut être acheminée. Celui-ci retrouve une stabilité lorsqu'un *VLAN* est reconfiguré (ici à $t=165$ s). Le SCR sera donc instable lorsqu'il y aura une rupture sur chaque *VLAN*. Par conséquent, cette expérience montre bien la nécessité d'avoir deux chemins les plus disjoints possibles. Notons, en revanche, que la solution ici proposée est alors plus consommatrice en bande passante qu'une solution mono-chemin.

4.3.5 Synthèse

Dans cette section, une méthode permettant d'améliorer la continuité de service sur Ethernet a été développée. C'est une méthode passive mettant en œuvre deux arbres couvrants sur le réseau. Ces deux arbres couvrants sont déterminés à l'aide d'algorithmes génétiques pour lesquels une fonction objectif a été définie. Cette fonction évalue des paires d'arbres couvrants, d'une part en calculant le nombre de sauts de chaque chemin et d'autre part, en évaluant le nombre d'équi-

pements communs aux deux chemins. Cette solution permet de réduire la probabilité que des applications critiques telles que les applications temps-réel, subissent des coupures de communication entraînant de longues phases de reconfiguration sur Ethernet. La contrepartie de notre solution, c'est qu'il y a un coût en bande passante qui est non négligeable et qui reste à évaluer. De plus, celle-ci a été illustrée dans le cas particulier d'un SCR. Par conséquent, une étude de cas plus approfondie au cas des lanceurs pourrait être réalisée de manière à évaluer la faisabilité dans le cas de communications multi-sources et multi-destination. Cependant, pour réaliser de manière similaire l'expérimentation de cette solution au cas des lanceurs, il est nécessaire de pouvoir observer l'état de la commande : l'observabilité est d'ailleurs une exigence qui est traitée au chapitre suivant. D'autre part, rappelons que cette commande n'est encore pas définie pour le lanceur ce qui ne permet pas actuellement la réalisation de l'expérimentation.

4.4 Conclusion

La méthode de reconfiguration présentée dans ce chapitre semble, d'après les performances obtenues dans les différentes applications, être compatibles dans des applications temps-réel (*Real-Time Process* dans la Table 4.1). Cependant, dans les applications lanceur, les équipements terminaux sont redondants et émettent les mêmes trafics. Ces trafics sont émis sur des chemins potentiellement différents selon le type d'architecture (centralisée ou distribuée) et de la position du calculateur centralisé de façon à ne perdre les données d'un flux sans perdre l'information envoyée par l'équipement redondant. Par conséquent, il pourra être nécessaire de calculer des chemins les plus disjoints possibles selon la seconde méthode présentée lors de la conception de l'application. Ainsi, en combinant ces deux approches, il serait possible d'obtenir une très grande disponibilité du réseau. La première solution proposée s'appuie sur des chemins qui ont été calculés relativement à la métrique du nombre de sauts. La seconde solution recherche la meilleure paire de chemins relativement à la métrique de probabilité de défaillances (liée au nombre d'équipements et à l'indépendance des chemins). Le fait de restreindre le nombre de sauts peut amener dans certains cas à considérer des chemins qui ne seraient plus indépendants. En vue de la combinaison des deux solutions, une piste à travailler pourra s'attacher à identifier une métrique quantifiant l'indépendance des chemins en fonction des informations d'états collectées de manière centralisée ou distribuée et de la probabilité ou du taux de défaillances des équipements. D'autre part, il restera également à évaluer le niveau de complexité / coût vis-à-vis du gain de disponibilité et de la contrainte à respecter. Cette étude pourrait également être complétée par l'évaluation du gain de la distribution et de la charge générée par le trafic supplémentaire de détection/reconfiguration. De plus, l'identification des différentes possibilités de chemins devra conduire à une évaluation temporelle de toutes ces possibilités.

Bien qu'il soit d'ores-et-déjà possible de réaliser une solution de reconfiguration topologique

dans le cas des lanceurs, celle-ci s'avère non suffisante. En effet, l'application lanceur nécessite l'observation des échanges pour la vérification des comportements au sol (notamment en vue de la certification) mais également en vol et au retour des vols de manière à connaître par exemple l'origine d'une panne. Ainsi, comme nous l'avons mentionné au chapitre 2, l'observabilité du réseau est une exigence capitale pour l'application lanceur. Sans le respect de cette exigence, la technologie ne serait pas retenue. Par conséquent, le chapitre suivant propose d'étudier les possibilités offertes par la technologie Ethernet.

Chapitre 5

Observer pour « jouer au sol » et isoler les défaillances

5.1 Introduction

L'application lanceur nécessite que les échanges puissent être observés (cf E. 6). En effet, cette fonctionnalité d'observabilité permettra alors de vérifier de manière déterministe l'état des communications lors de tests au sol préalables au vol mais également après la mission lors des analyses menées à l'issue de chacun des vols du lanceur.

Nous proposons ici de définir l'observabilité comme la capacité à observer les échanges sur le réseau. Cette fonctionnalité doit être réalisée sans perte et avec un ordonnancement des paquets conforme au comportement du réseau.

Les techniques d'enregistrement utilisées dans Ariane 5 ne peuvent plus s'appliquer car l'architecture réseau n'est plus un bus où un échange pouvait être observé en tout point du réseau (diffusion). Il n'y avait donc qu'un seul domaine d'observation comme illustré à la FIGURE 5.1. Une architecture Ethernet commutée (donc segmentée) nécessite l'instrumentation du réseau en différents points afin d'obtenir plusieurs vues locales (on parlera de collecte). Il sera alors nécessaire de concaténer (sérialiser et appairer) les traces obtenues (en ces différents points) sur une station enregistreur et de les fusionner de manière à ce qu'elles soient cohérentes vis-à-vis de la réalité (ordre des échanges notamment). Il peut également être envisagé une dernière étape consistant à compresser l'enregistrement des échanges afin de renvoyer au sol une vue réduite (mais toutefois reconstructible dans son intégralité) des échanges stockés. Cette vue serait réduite pour respecter la contrainte de bande passante bord-sol allouée pour la télémétrie.

Dans la littérature, peu (ou pas) de travaux traitent de l'observabilité d'un réseau Ethernet. Cette problématique est en effet très spécifique au domaine du spatial. En revanche, il est possible de trouver des articles traitant de la collecte de trames dans des réseaux WAN (*Wide Area*

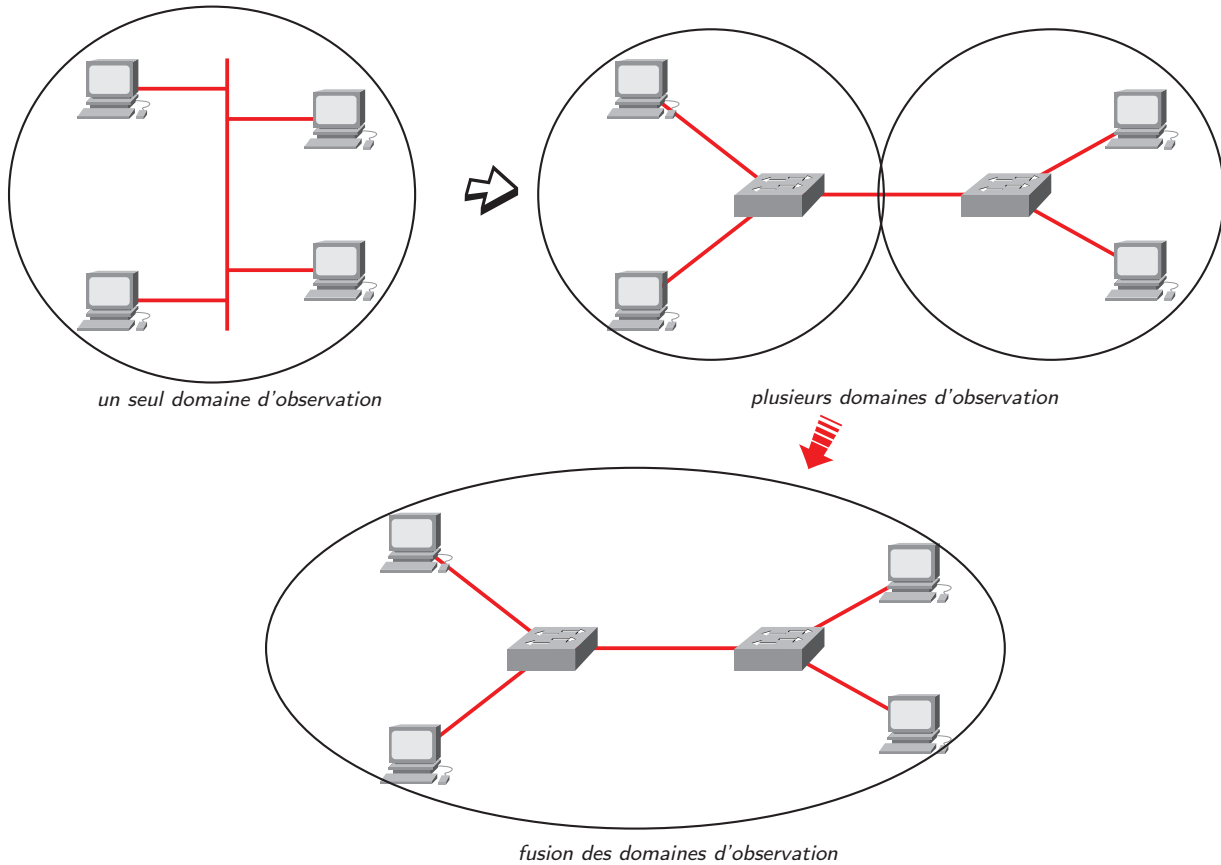


FIGURE 5.1: Evolution de l'observabilité

Network) (Anderson et Arlitt, 2006) pour de l'analyse hors-ligne. L'objectif est principalement de réaliser (et d'évaluer) l'instrument de collecte créé par leur soin de manière à réaliser la capture de traces sur plusieurs jours (de manière à les traiter hors-ligne). Le challenge réside plutôt dans le stockage d'une quantité importante de trafic. (Papadogiannakis *et al.*, 2010) propose également une méthode pour le stockage d'une capture de trafic à long-terme sur une mémoire de taille fixe. Pour cela, la quantité des trames conservées décroît en fonction de l'âge des trames. Ces techniques sont notamment utilisées dans des applications d'administration ou de sécurité de réseaux.

D'autres travaux s'intéressent à la création de sondes pour la capture de trafic (Ferrari *et al.*, 2008) dans des environnements temps-réel à des fins de mesures de caractéristiques temporelles (délais de bout en bout, synchronisation, ...). Une implémentation sur FPGA est réalisée. Bien que ces travaux sont intéressants puisque fournissant un équipement *low-cost*, ils ont l'inconvénient de demander du temps de développement (et ne profitent pas de fait de la maturité du standard). Cette technique est intrusive dans le sens où il est nécessaire de connecter l'équipement sur un segment réseau à la manière des équipements commerciaux nommés TAP (*Test Access Point*). Le TAP consiste par principe en une dérivation électrique (couche 1) : il permet la duplication du signal réseau (au sens électrique, comme il serait fait avec une pince vampire par

exemple). Une autre méthode consiste à utiliser une fonctionnalité des commutateurs appelée *port mirroring* consistant à copier le trafic d'un certain nombre de ports sur un port dédié de manière à collecter l'ensemble du trafic (Cisco Systems, 2005).

La section présente alors les choix retenus pour notre proposition d'observabilité d'un réseau Ethernet.

5.2 Observabilité d'un réseau Ethernet commuté

Il est proposé de réaliser l'observabilité d'un réseau Ethernet commuté en instrumentant le réseau en différents points pour la collecte des échanges. Puis, ces différentes collectes seront alors concaténées et fusionnées pour n'obtenir qu'une seule et unique trace de l'ensemble des échanges. Enfin, une dernière étape consisterait en une compression de celle-ci pour l'envoi au sol (lorsque l'on se trouve en vol).

5.2.1 Collecte des échanges

La collecte des échanges peut s'appuyer sur :

- le *TAP* (*Test Access Port*)
- le *port mirroring* sur les commutateurs
- ou des sondes se basant sur les travaux de la littérature comme celle de (Ferrari *et al.*, 2008) par exemple

Rappelons que cette fonctionnalité d'observabilité ne doit pas être contradictoire avec les exigences de coût, de masse, etc.

Par conséquent, les sondes de la littérature qui nécessiteraient un temps de développement (et donc un coût) non négligeable pour l'adapter à notre besoin ne seront donc pas retenues.

De plus, la technique du TAP nécessite l'ajout de TAP sur chaque lien à observer. Ainsi, l'ajout d'équipements entraîne une augmentation de la masse et du coût du système de communication.

La fonctionnalité de *port mirroring* est disponible dans les commutateurs COTS. En revanche, (Zhang et Moore, 2007) présente une comparaison de cette technique et du TAP relativement à des critères temporels, d'ordre des paquets et de pertes de paquets. Cette étude met en avant que le *port mirroring* n'est pas conforme à la réalité vis-à-vis de ces critères.

Nous avons donc mené des expériences similaires sur le *port mirroring*. Une première expérience consistait à obtenir une trace sur chaque commutateur d'une architecture de test sur laquelle il a été extrait des informations temporelles (notamment l'« erreur relative » entre la période théorique et le temps d'inter-arrivée) et d'ordre des échanges. Dans cette expérience, aucune perte n'a été observée et une variation limitée du délai d'inter-arrivée (très inférieure à la milliseconde) a été observée. Cette expérience a été renouvelée dans des situations de saturation

du réseau et de concurrence entre flux. Lors de situation de saturation du réseau, il a été observé que le *port mirroring* n'a pu retransmettre tous les paquets (pertes) mais l'ordre des échanges a été respecté.

Ainsi, nos expériences ne retrouvent pas les conclusions de (Zhang et Moore, 2007). De plus, l'application lanceur ne doit pas se retrouver dans un scénario de saturation et ce sera démontré analytiquement en phase de conception. Par conséquent, nous retiendrons pour méthode de collecte, le *port mirroring*.

Nous disposons donc d'une solution capable de réaliser sur chaque commutateur la collecte des échanges. Il s'agit ensuite de réaliser la fusion de ces différentes collectes.

5.2.2 Fusion des différentes collectes

Nous avons vu précédemment que la méthode de collecte s'appuie sur le *port mirroring*. Le problème restant est la fusion (et le « re-séquencement ») des données.

Une première proposition consiste à ajouter un commutateur destiné à concaténer ces différents points de collecte sur la station enregistreur qui sera alors connectée à celui-ci. Bien que cette solution nécessite un commutateur et des liens supplémentaires, elle permet de ne pas perturber le trafic applicatif et ne remet donc pas en cause l'analyse temporelle présentée dans le chapitre 3.

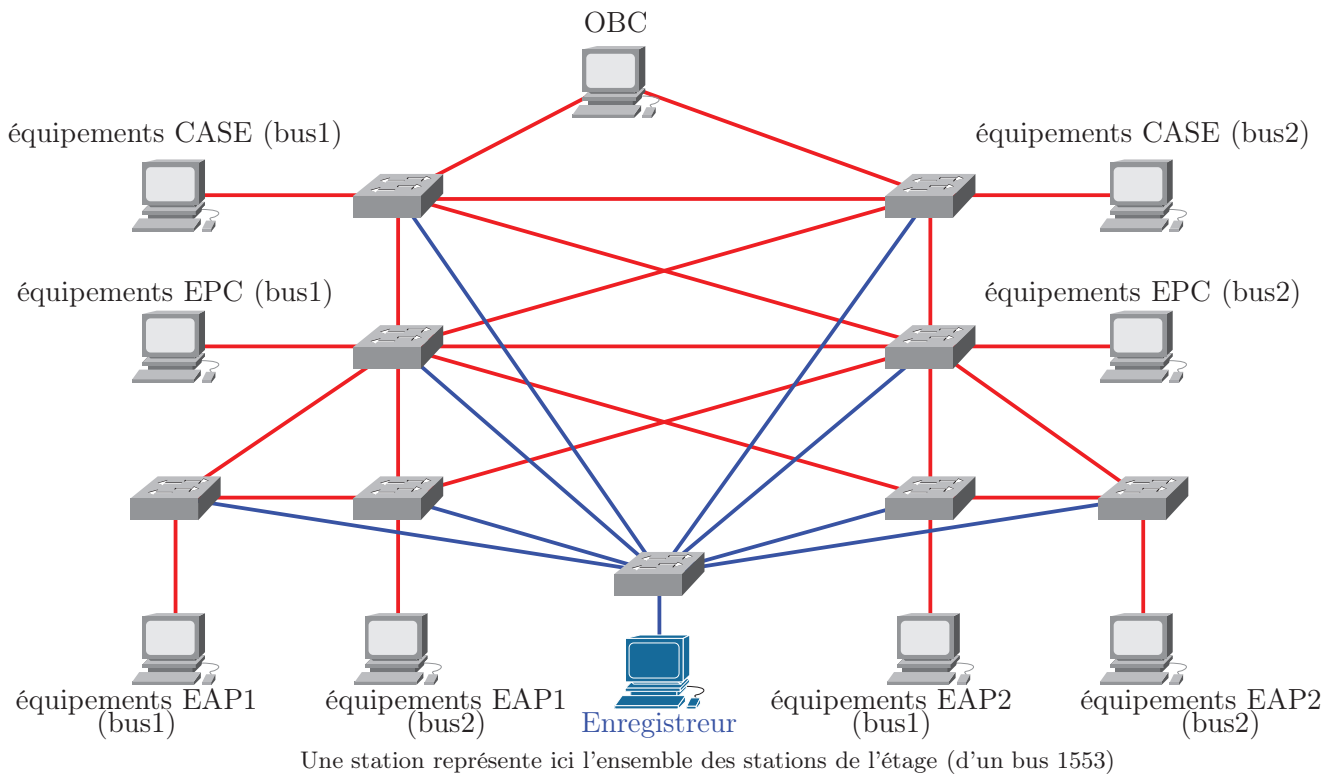


FIGURE 5.2: Architecture 2 (présentée au chapitre 2) étendue

Il y a deux inconvénients à cette méthode : l'augmentation du nombre de câbles et la méconnaissance de l'ordre des paquets. En effet, la contrainte de réduction des câbles (et donc de la masse) rend cette solution est inadaptée. De plus, lors de la fusion, il est nécessaire de conserver l'ordre des échanges (cohérence de l'observation) : il ne faut donc pas se retrouver dans la situation où un message est vu sur un point de collecte avant qu'il ne soit vu sur le point de collecte précédent (vis-à-vis de son chemin de communication connu).

Pour remédier à ces inconvénients, nous proposons alors d'utiliser un mécanisme capable de synchroniser les équipements (ceux réalisant les captures) sur une horloge de référence. Comme la synchronisation est une contrainte forte pour l'application lanceur, elle nécessite d'autant plus l'utilisation d'un protocole de synchronisation adapté aux exigences de l'application (et de la fonction observabilité). Cette synchronisation permettra la datation des captures de manière homogène en tout point de collecte du réseau (sous réserve d'une certaine précision). Ainsi, les différentes collectes, qui seront acheminées par le réseau périodiquement, pourront être aisément « re-séquencées » grâce à leur date de capture.

Une question importante soulevée par la fonction « observabilité » (et en général, dans les systèmes en réseau) porte sur la synchronisation des horloges des différents équipements distribués sur le réseau. Ethernet ne propose pas nativement un moyen d'assurer la synchronisation des horloges des équipements et des entités qu'il interconnecte. Toutefois, différents travaux se sont portés sur cette problématique. Le premier protocole proposé, NTP (Network Time Protocol) (Mills,1991), définit ainsi un cadre de synchronisation d'horloges. Les fortes contraintes des applications distribuées temps-réel ont toutefois conduit au développement d'un nouveau protocole. Le standard IEEE 1588 (IEEE Computer Society, 2002a) présente un protocole pour synchroniser des horloges indépendantes de nœuds d'un système de contrôle / mesure distribué avec une grande précision. Dans ce protocole, tous les nœuds (niveau instrumentation compris) possèdent une horloge IEEE 1588 qui est synchronisée avec tous les autres acteurs *via* le protocole PTP (*Precision Time Protocol*). Les équipements peuvent ainsi estampiller leurs données et fonctionner en suivant un référentiel horaire identique. L'homogénéité de l'horloge dépend alors de la synchronisation des horloges temps réel locales.

Pour assurer la cohérence de l'observation, il est nécessaire que la valeur de l'écart de synchronisation soit inférieure au temps de transmission minimal (entre l'émission d'un paquet de taille minimale sur un port et la capture de celui-ci sur le port du point de collecte suivant). Cette valeur peut être estimée à une dizaine de microsecondes (temps de transmission + « latence » commutateur + temps de propagation soit environ $5,76 \mu s + 5 \mu s$ pour un débit de 100 Mb/s , des trames de 72 octets et un temps de propagation négligeable car sur un médium de quelques dizaines de mètres). Notons que cette contrainte serait d'autant plus forte que le débit est important. Par exemple, à 1 Gb/s , cette valeur serait estimée à environ 1 à $2 \mu s$ (selon l'évolution de la « latence » du commutateur).

Une série d'expérimentations sur le protocole PTP a alors été menée pour étudier la précision de synchronisation en fonction du niveau de la fonction estampillage qui peut être, comme illustré à la FIGURE 5.3, au niveau de la couche applicative (point C), entre la couche MAC et les couches supérieures (point B) ou entre la couche physique et la couche MAC (point A). Cette étude est présentée dans la section suivante.

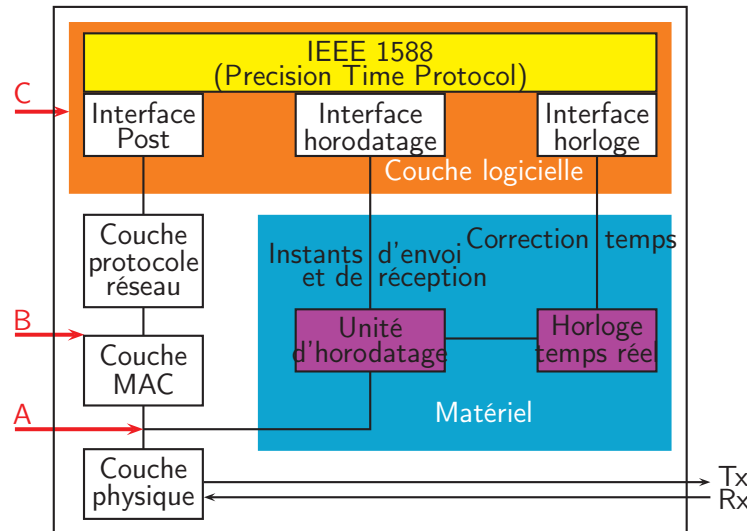


FIGURE 5.3: Niveau d'implémentation de la synchronisation en fonction du niveau de la fonction d'estampillage

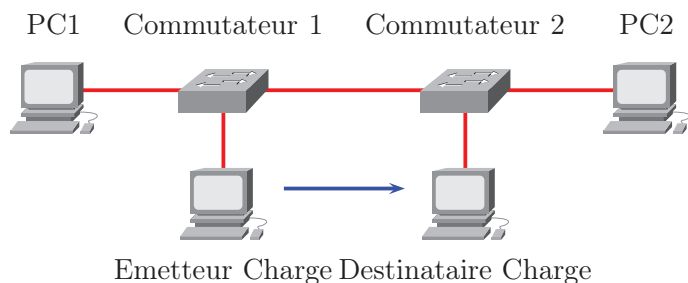
5.2.2.1 Etude de la précision du protocole de synchronisation PTP

Pour réaliser l'étude de la précision du protocole de synchronisation PTP, nous avons mené des expériences sur l'architecture présentée à la FIGURE 5.4. Notons que des commutateurs industriels Cisco® IE 3000 ont été utilisés (ces commutateurs ont été prêtés par Cisco). La synchronisation sur les stations terminales est réalisée grâce à une librairie *open-source* (PTP daemon – PTPd) implémentant de manière *software* le protocole PTP.

L'objectif de ces expériences est d'évaluer l'*offset* entre l'horloge maître et esclave dans le cas d'une synchronisation de bout en bout (implémentation *software*) et d'une synchronisation point à point (implémentation *hardware* ou hybride). Sur ces deux expériences, nous avons étudié la sensibilité à la charge de ces deux solutions. Enfin, nous avons cherché à montrer le gain de précision dans le cas où une synchronisation totalement *hardware* serait implémentée.

Synchronisation de bout en bout (*software*)

Dans cette expérience, la station PC1 (illustrée à la FIGURE 5.4(a)) est l'horloge maître et la station PC2 l'horloge esclave. Les commutateurs sont ici transparents (ils ne sont pas synchronisés



(a) Architecture de test



(b) Photo de l'architecture de test

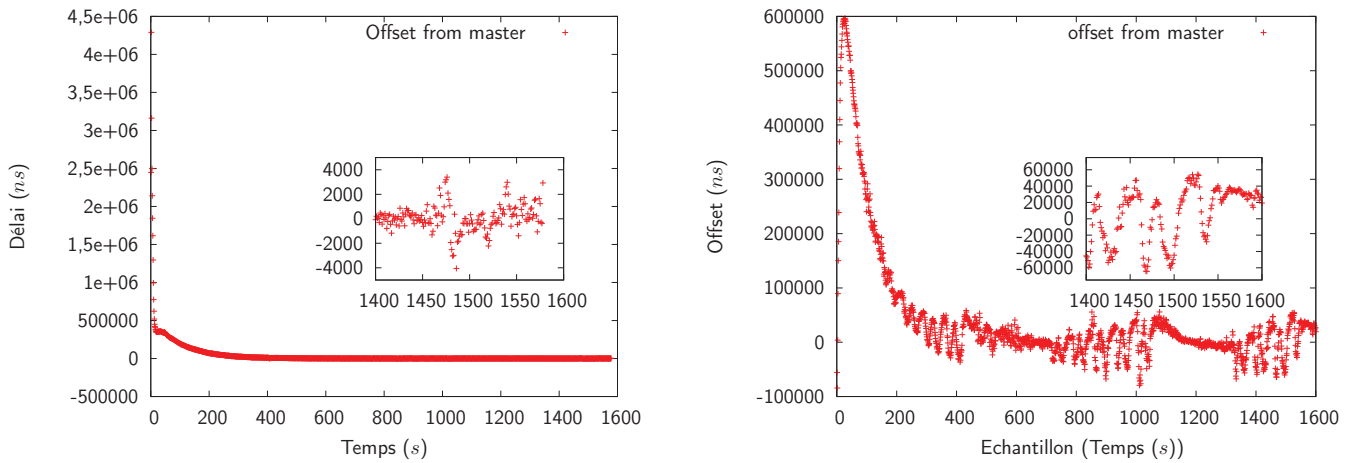
FIGURE 5.4: Expérimentation de l'observabilité

et n'interviennent pas dans le processus de synchronisation). Trois scénarios de charge sont alors testés :

1. sans charge : tous les liens sont à 100 Mb/s et la station *Emetteur Charge* n'émet pas de trafic,
2. avec une charge de 25 % : les débits de l'ensemble des ports sont diminués à 10 Mb/s (car à 100 Mb/s , il serait nécessaire d'envoyer une trame de 1526 octets toutes les $500\ \mu\text{s}$ pour obtenir les 25% de charge, or nos équipements ne le permettent pas). La station « *Emetteur Charge* » émet à $2,44\text{ Mb/s}$ (une trame de 1526 octets toutes les 5 ms) soit approximativement 25 % de charge sur le lien entre les commutateurs.
3. avec une charge de 60 % : la station « *Emetteur Charge* » émet une trame de 1526 octets toutes les 2 ms soit $6,1\text{ Mb/s}$ (les débits de l'ensemble des ports sont de 10 Mb/s)

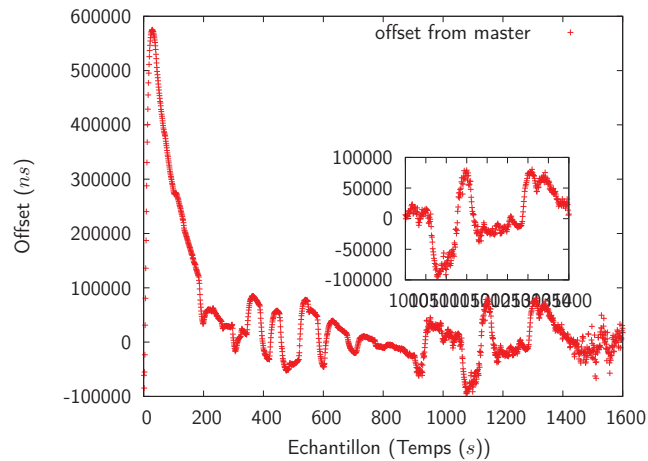
La FIGURE 5.5 montre l'observation de l'*offset* entre l'horloge Maître et l'Esclave en fonction du temps.

Ces premières expériences montrent d'ores et déjà que la synchronisation passe par une phase transitoire avant d'atteindre un *offset* (entre le maître et l'esclave) relativement « stable ». C'est la variation de l'*offset* en régime établi qui donne une indication sur la précision de synchronisation. Cette phase transitoire peut être plus ou moins longue selon l'écart initial entre les horloges. Il sera alors nécessaire de démarrer le trafic applicatif ainsi que l'observation lorsque la synchronisation sera en régime établi. Dans ce premier exemple, la phase transitoire dure approximativement 400 s . Ce temps de convergence s'explique par le temps de réponse de la boucle de contrôle de l'horloge locale : ce temps est en fait la réponse caractéristique du contrô-



(a) Sans charge

(b) Avec 25 % de charge



(c) Avec 60 % de charge

FIGURE 5.5: *Offset* entre l'horloge Maître et l'Esclave

leur PI (proportionnel-intégral) implémenté dans cette version logicielle de PTP (PTPd) (Correll *et al.*, 2005). Pour améliorer ce temps de convergence, il serait alors nécessaire d'optimiser les paramètres du contrôleur.

Sans charge (FIGURE 5.5(b)), nous observons un *offset* ayant des valeurs en régime établi comprises entre $-5 \mu s$ et $5 \mu s$. Dans ce scénario, la précision de la synchronisation serait adaptée à l'application. En revanche, dès lors que l'on ajoute de la charge (FIGURE 5.5(b) et 5.5(c)), l'*offset* varie dans une plage plus importante (de $-60 \mu s$ à $60 \mu s$ à 25 % et de $-100 \mu s$ à $100 \mu s$ à 60 %). Par conséquent, ce type de synchronisation est sensible à la charge. De plus, elle ne respecte pas la précision attendue pour l'observabilité précédemment établie (de l'ordre d'une dizaine de microsecondes).

En fait, dans ces scénarios de charge, les délais mesurés sont relativement différents selon le sens de communication (maître - esclave ou esclave - maître). Par conséquent, l'évaluation du temps de propagation (de bout en bout) est alors biaisée par la charge réseau, d'autant plus que les commutateurs sont sensés être transparents (au sens où ils ne donnent pas d'indication sur le temps de traversée des messages, considéré comme nul) et le calcul de l'*offset* (basé sur ce temps de propagation) en pâtit.

Il est alors possible de conclure qu'un équipement terminal basé sur une synchronisation implémentant un estampillage de haut niveau (*software*) ne pourra être utilisé comme référence temporelle.

Une seconde batterie de tests est réalisée avec une référence temporelle fournie par le réseau. La synchronisation est qualifiée d'hybride dans le sens où les équipements terminaux seront les mêmes (implémentant la synchronisation *software*) et les commutateurs implémenteront quant à eux une synchronisation *hardware* (estampillage de bas niveau).

Synchronisation point à point hybride

L'architecture de test reste identique. Cependant, les commutateurs sont configurés de manière à être eux mêmes synchronisés. Dans ce mode, la synchronisation est réalisée point à point : chaque port des commutateurs sera alors horloge maître ou esclave. La FIGURE 5.6 présente le rôle de chacun d'entre eux. Notons que l'*offset* observé (ou *offset* local) correspond à la différence entre l'horloge du port maître et celle du port esclave. L'*offset* global (correspondant à la différence entre la référence temporelle et l'esclave) peut alors être déduit de ces différents *offsets* locaux en les additionnant.

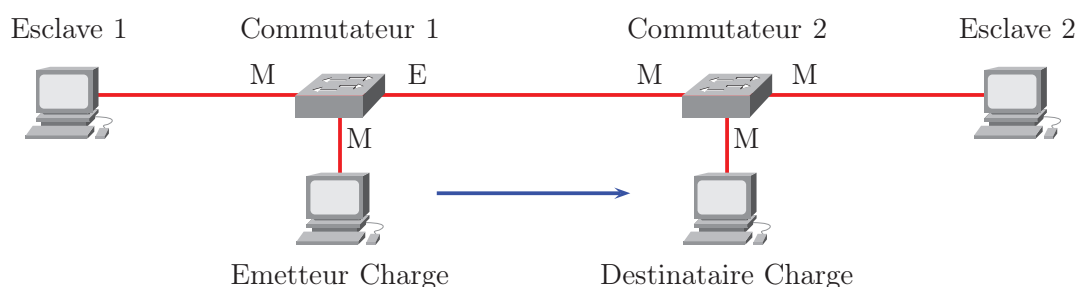
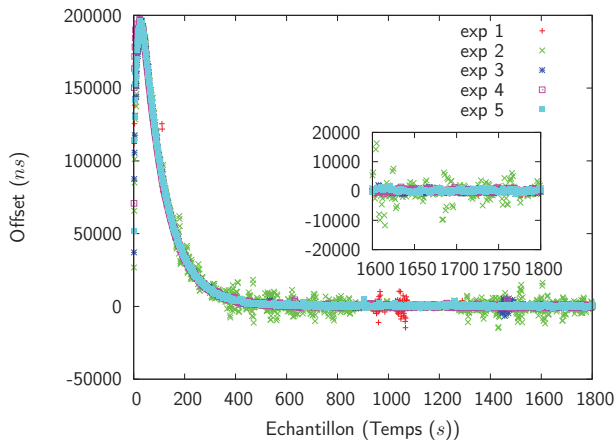


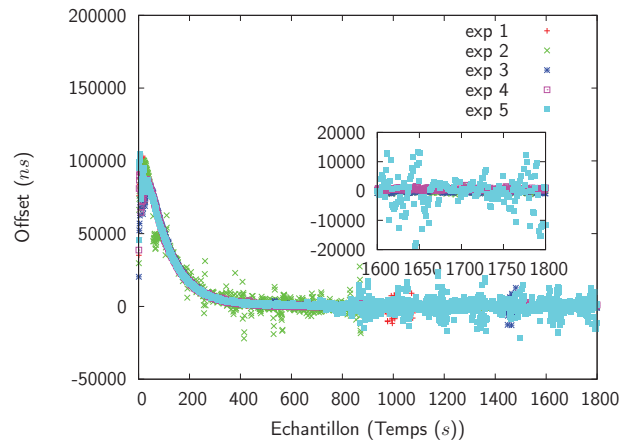
FIGURE 5.6: Architecture de test de synchronisation hybride

Les trois scénarios précédents ont de nouveau été réalisés. Comme les différentes manipulations sont reproductibles, elles ont toutes été réalisées 5 fois. La FIGURE 5.7 montre l'observation de l'*offset* entre l'horloge Maître (un commutateur) et l'Esclave (un équipement terminal) en fonction du temps.

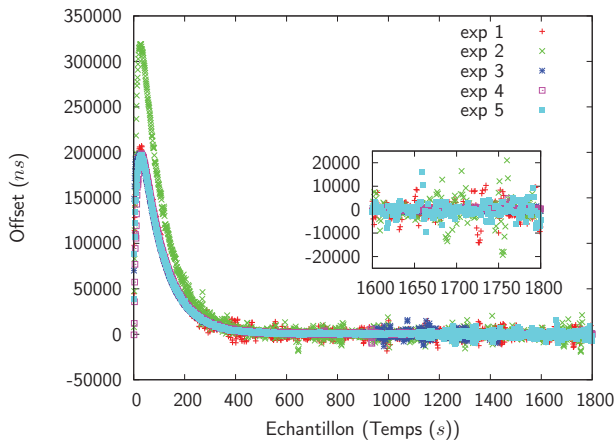
Ces expériences permettent de mettre en évidence que la synchronisation implémentée de manière hybride n'est pas sensible à la charge (quelque soit le niveau : sans charge, 25% et 60% de



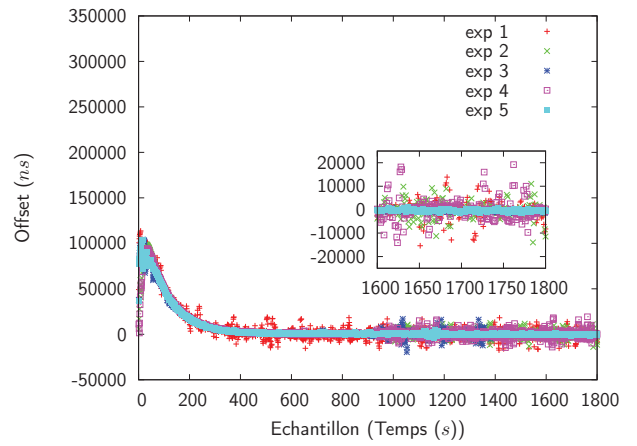
(a) Sans charge (Maître : commutateur 1 et Esclave 1)



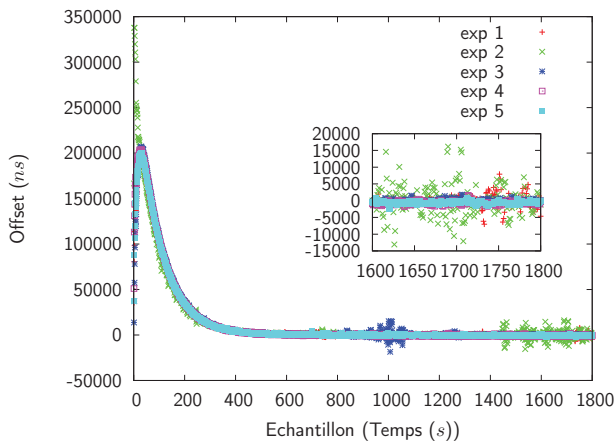
(b) Sans charge (Maître : commutateur 2 et Esclave 2)



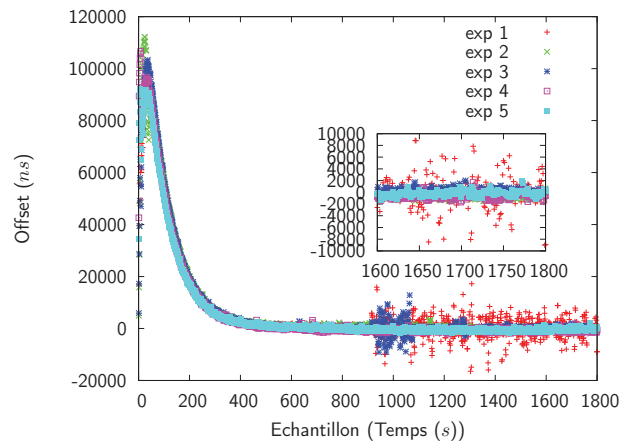
(c) Avec 25 % de charge (Maître : commutateur 1 et Esclave 1)



(d) Avec 25 % de charge (Maître : commutateur 2 et Esclave 2)



(e) Avec 60 % de charge (Maître : commutateur 1 et Esclave 1)



(f) Avec 60 % de charge (Maître : commutateur 2 et Esclave 2)

FIGURE 5.7: *Offset* entre l'horloge Maître et l'Esclave

charge). Globalement, les variations de l'*offset* local sont comprises entre $-20 \mu\text{s}$ et $20 \mu\text{s}$ quelque soit la charge. Ces variations sont principalement dûes au fait que les esclaves implémentent une version logicielle (*software*) du protocole PTP qui peut être soumise à la concurrence de tâches au niveau processeur impliquant un mauvais estampillage et donc une erreur dans le calcul de l'*offset* local. D'autre part, la charge n'était pas directement appliquée sur les liaisons point à point des esclaves dont l'*offset* était relevé. En revanche, celui-ci aurait pu être impliqué par les calculs de proche en proche. En effet, des échanges PTP sont réalisés entre les commutateurs de manière à ce qu'ils soient synchronisés et qu'ensuite le commutateur 1 soit synchronisé avec l'esclave 1 et ainsi de suite pour que tous les équipements y soient. Ainsi, dans ce cas de figure, il semblerait *a priori* que la synchronisation *hardware* (entre les commutateurs) ne soit pas sensible à la charge.

Pour s'en convaincre, il serait nécessaire de réaliser des expérimentations d'une synchronisation *hardware* pour en évaluer les apports (qui sont présagés dans ces expériences). Actuellement, nous ne possédons pas de cartes Ethernet IEEE 1588 dans les équipements terminaux mais nous avons deux commutateurs IEEE 1588 qui vont nous permettre de réaliser une expérience montrant l'apport de la synchronisation.

D'autre part, il est important de noter qu'ici les conclusions ont été faites directement sur l'*offset* local (car les valeurs observées n'étaient déjà pas compatibles avec l'exigence définie) mais celles-ci auraient dues être menées sur l'*offset* global entre les équipements terminaux implémentant la fonction d'observabilité. En effet, c'est cet *offset* global qui doit être conforme à l'exigence définie pour l'observabilité. Pour rappel, cet *offset* global correspond à la somme des *offsets* locaux préalablement observés. A titre d'exemple, l'*offset* global entre les esclaves 1 et 2 dans le cas d'une charge de 60% est illustré à la FIGURE 5.8.

Bien que les *offsets* locaux peuvent s'annuler (en fonction de leur signe), l'*offset* global est néanmoins compris entre $-15 \mu\text{s}$ et $15 \mu\text{s}$ confirmant la non compatibilité de ce type de synchronisation (hybride) pour la fonction observabilité. La section suivante présente une expérience afin de mettre en avant l'apport d'une synchronisation *hardware*.

Apport de la synchronisation *hardware*

Dans cette expérience, nous désirons observer l'évolution de l'*offset* entre deux équipements implémentant la synchronisation de niveau *hardware*. Dans notre architecture, ce type de synchronisation est réalisé par les deux commutateurs Cisco[®] : le commutateur 2 en tant que maître et le commutateur 1, l'esclave. Ainsi, nous allons observer périodiquement, sur le commutateur 1, l'*offset* entre les deux commutateurs et voir son évolution dans le temps ainsi que sa sensibilité à la charge. Les FIGURE 5.9(a), 5.9(b) et 5.9(c) montrent l'évolution des *offsets* en fonction du temps et de la charge, respectivement sans charge, 25% et 60% de charge.

Ces expériences montrent que l'*offset* observé entre les deux commutateurs est en-dessous de

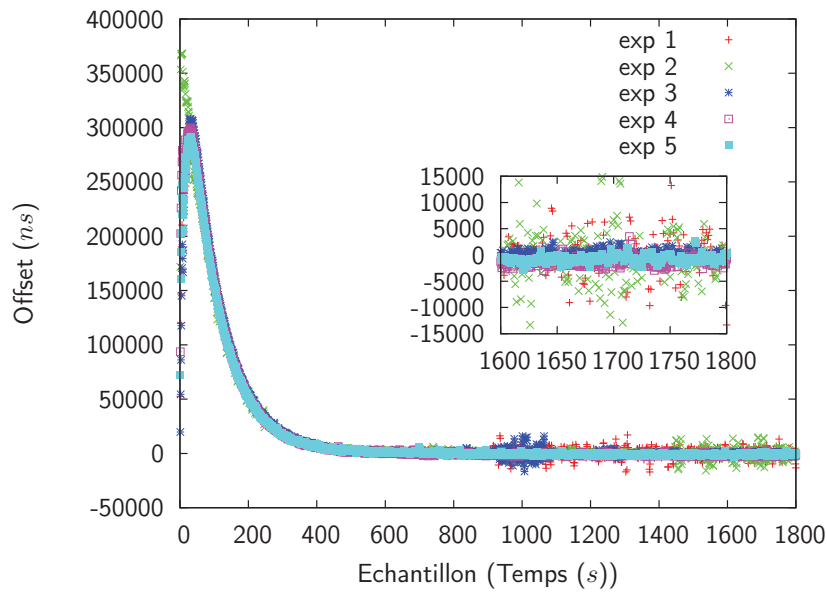


FIGURE 5.8: *Offset* global entre l’horloge de l’esclave 1 et l’Esclave 2 avec une charge de 60%

la microseconde quelque soit la charge entre ces deux commutateurs.

Ces expériences mettent en évidence le gain qui pourrait être obtenu dans le cas d’une synchronisation totalement *hardware*. En effet, à terme si l’observabilité est directement implémentée dans des commutateurs IEEE 1588, la précision de synchronisation entre commutateurs (précision sous la microseconde quelque soit la charge) sera adaptée à cette fonctionnalité. De plus, la précision de synchronisation *software* sur les équipements terminaux pourrait tout à fait convenir tant que les contraintes applicatives de périodicité ne sont pas en dessous de la milliseconde, sous réserve de leur non-sensibilité à la concurrence de tâche sur des systèmes temps réel. En revanche, dans le cas où cette fonctionnalité est déportée sur des équipements terminaux, il se peut (pour les raisons évoquées précédemment, de concurrence de tâches processeurs notamment) que la précision d’une synchronisation hybride ne soit pas suffisante pour la fonction d’observabilité.

Pour résumer, à terme et de manière à respecter la contrainte, l’estampillage devra être réalisé au plus bas niveau. Une première expérience de fusion est présentée dans la section suivante.

5.2.2.2 Expérimentation de la fusion de collecte

L’objectif de cette expérience est de réaliser sur l’architecture présentée à la FIGURE 5.10, la fusion de deux captures obtenues en deux points de collecte différents (PC1 et PC2).

Comme le *port mirroring* ne retransmet que le trafic dupliqué (ce port n’est alors pas éligible comme maître ou esclave), le point de collecte (PC) ne peut donc pas être synchronisé. Ainsi, de manière à ce que ce soit possible, nous avons recours à des équipements bénéficiant de deux cartes de communication Ethernet de façon à enregistrer le trafic observé *via* une carte et synchroniser

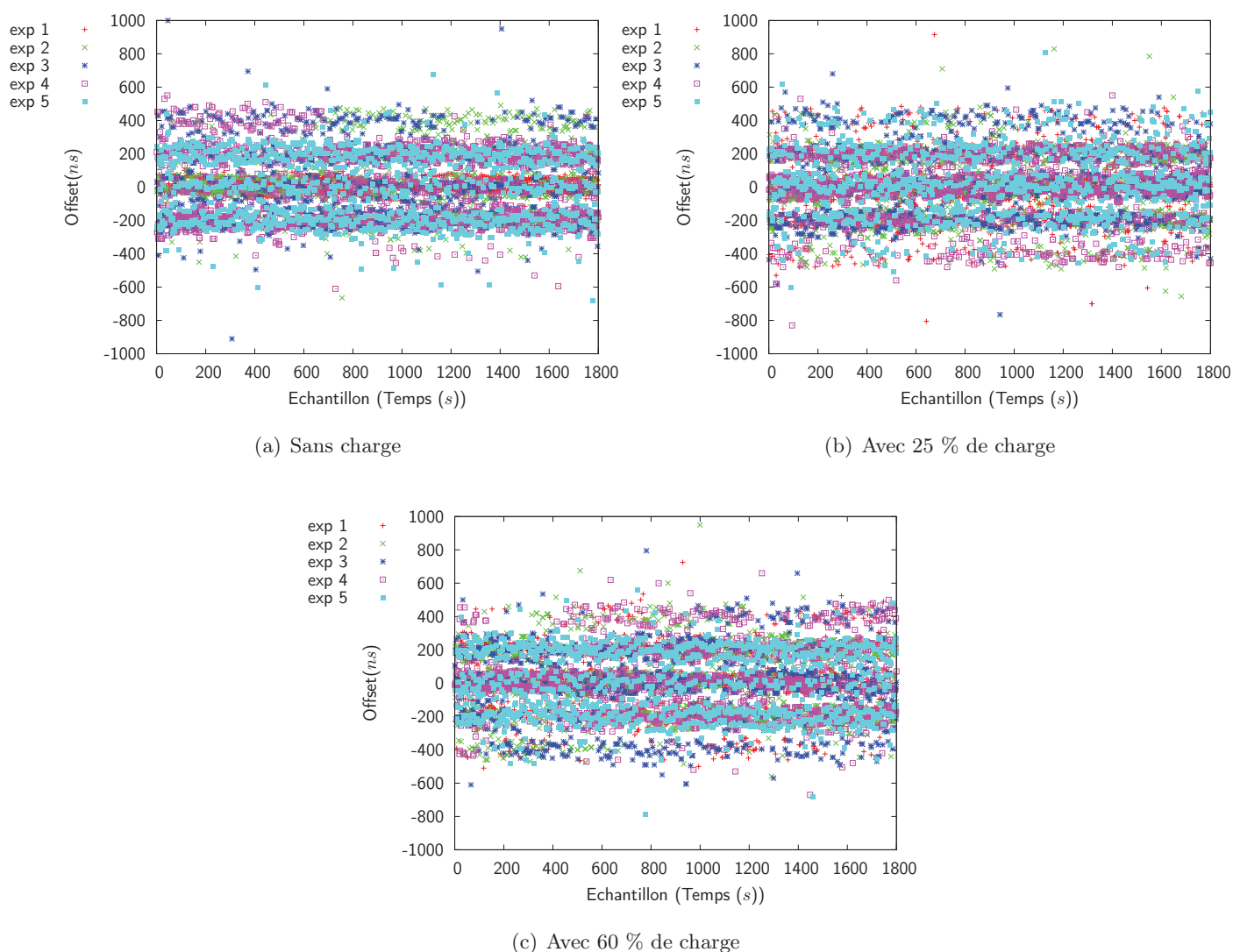


FIGURE 5.9: *Offset* entre l'horloge Maître (commutateur 2) et l'Esclave (commutateur 1)

l'équipement *via* la seconde. Notons ici qu'il s'agit d'expériences amont de faisabilité et que, à terme, cette fonctionnalité pourrait se retrouver embarquée dans les commutateurs.

Le trafic applicatif à observer est un trafic périodique de période 100 *ms* de taille maximale de 1500 octets utiles (soit 1526 octets). Il est émis par la station « Emetteur Charge » à destination de la station « Destinataire Charge ». Ce trafic correspond à une charge de 0,01% de la bande passante disponible (100 *Mbps*). Ces trafics sont de plus numérotés (dans le champ donnée).

Les stations PC1 et PC2 enregistrent le trafic à observer grâce à la librairie *tcpdump* en estampillant logiquement chaque trame capturée et le stockent dans un fichier *.pcap* permettant l'ouverture dans le logiciel de capture WIRESHARK. La FIGURE 5.11 présente les deux enregistrements obtenus séparément. Les flèches en pointillé montrent la localisation de la collecte de

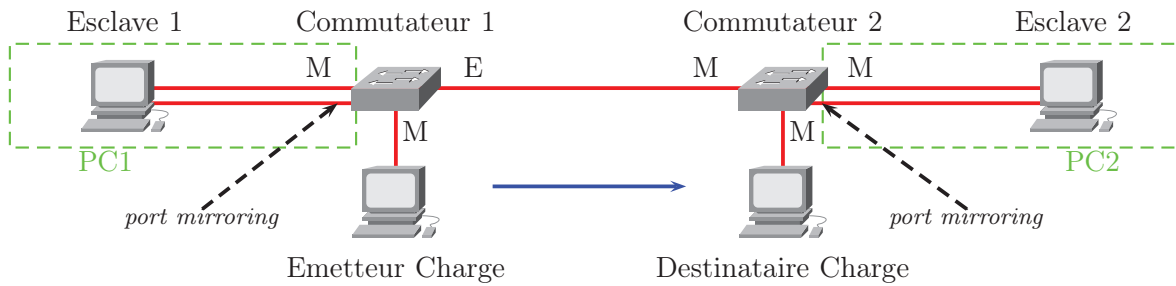


FIGURE 5.10: Architecture de test de fusion sur une architecture avec synchronisation hybride

chaque message.

Dans ces captures, nous pouvons observer les paquets applicatifs (en blanc) ainsi que l'ensemble des paquets de synchronisation PTP (en bleu). Ainsi, il est possible, par exemple, d'observer que la périodicité (100 *ms*) du flux applicatif est respectée (aux centaines de microsecondes près) sur chacun des points de collecte. Notons qu'à partir de celles-ci, d'autres statistiques auraient pu être obtenues comme le débit réel par type de trafic, le nombre de paquets, etc.

Ensuite, ces deux captures sont fusionnées grâce à la fonctionnalité proposée par le logiciel pour n'obtenir qu'un seul enregistrement. La FIGURE 5.12 présente la fusion des deux enregistrements. Bien qu'il soit possible de suivre les échanges PTP, nous nous concentrons sur les échanges applicatifs pour vérifier que l'ordre est cohérent : tout paquet applicatif doit être observé sur PC1 puis sur PC2 (le numéro de paquet dans les données permet de s'assurer qu'il s'agit bien du même paquet).

Ainsi, par exemple, les trames No.2977 et No.2978 (numéro attribué par le logiciel) sont deux observations d'un même message (le numéro de paquet dans les données est identique). Si nous nous référons aux captures précédentes (cf FIGURE 5.11, trames No.1201 et No.1777), il est possible de déduire que la trame No.2977 enregistrée à 01 :27 :51.196826 a bien été observée sur le PC1 alors que la trame No.2978 enregistrée à 01 :27 :51.196957 a bien été observée sur le PC2. Par conséquent, l'observation de ce message tout au long de son chemin est cohérente vis-à-vis de la réalité.

D'autre part, ces deux observations sont capturées à un intervalle de 131 μs correspondant approximativement au temps de transmission (122 μs). Par conséquent, il est clair que la précision de synchronisation ne doit pas être supérieure au temps de transmission. Dans les conditions de cette expérimentation, cette précision est inférieure au temps de transmission des paquets applicatifs puisque l'écart de synchronisation observé (dans les expériences préliminaires) varie a priori entre $-20 \mu s$ et $20 \mu s$.

Cette condition permet dans cette expérience, d'observer le bon ordonnancement des paquets. En revanche, si elle n'était pas respectée, il aurait pu être observé la trame No.2978 avant celle

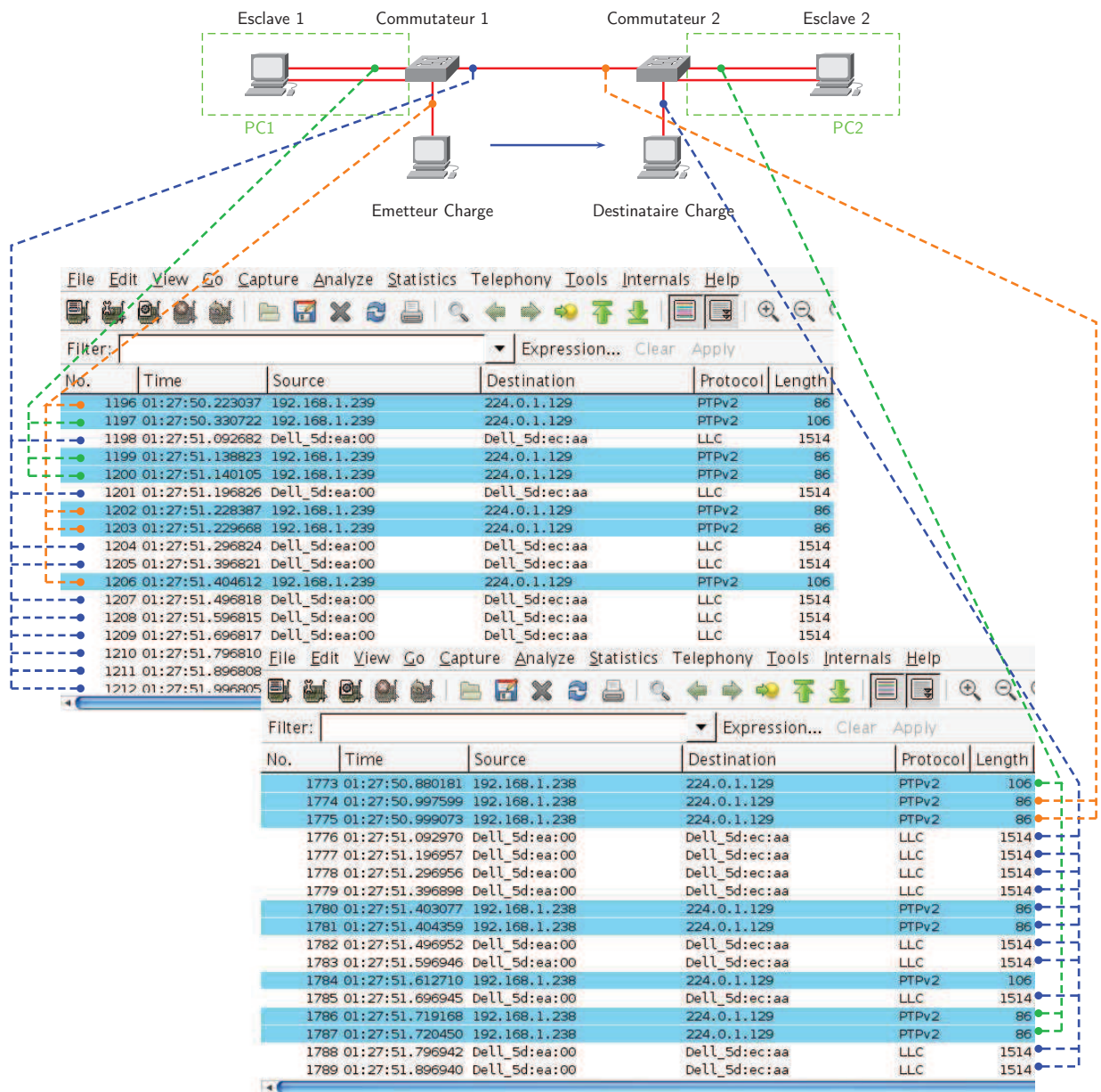


FIGURE 5.11: Captures obtenues sur les deux enregistreurs (avec les ports de collecte associés)

No.2977 ce qui ne correspondrait alors pas à la réalité. Notons que dans le cas d'une émission simultanée de trames sur plusieurs ports, le commutateur réalise un multiplexage des copies (logicielles) de chacune d'entre elles sur la *port mirroring*. Ainsi, l'estampillage de la collecte (qui est réalisé sur l'enregistreur) ne sera pas identique pour toutes les trames (l'écart entre les dates de capture sera de l'ordre d'un temps de transmission). Néanmoins, ce cas de figure ne doit pas compromettre l'observation d'un même message tout au long de son chemin (il ne doit pas être vu sur un commutateur avant qu'il ne soit vu sur le précédent). Par conséquent, de manière à minimiser l'impact de cette sérialisation des trames, il convient d'augmenter la capacité des liens

No.	Time	Source	Destination	Protocol	Length
2970	01:27:50.680161	192.168.1.238	224.0.1.129	PTPv2	86
2971	01:27:50.997599	192.168.1.238	224.0.1.129	PTPv2	86
2972	01:27:50.999073	192.168.1.238	224.0.1.129	PTPv2	86
2973	01:27:51.092682	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2974	01:27:51.092970	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2975	01:27:51.138823	192.168.1.239	224.0.1.129	PTPv2	86
2976	01:27:51.140105	192.168.1.239	224.0.1.129	PTPv2	86
2977	01:27:51.196826	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2978	01:27:51.196957	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2979	01:27:51.228387	192.168.1.239	224.0.1.129	PTPv2	86
2980	01:27:51.229668	192.168.1.239	224.0.1.129	PTPv2	86
2981	01:27:51.296824	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2982	01:27:51.296956	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2983	01:27:51.396821	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2984	01:27:51.396898	Dell_5d:ea:00	Dell_5d:ec:aa	LLC	1514
2985	01:27:51.403077	192.168.1.238	224.0.1.129	PTPv2	86
2986	01:27:51.404359	192.168.1.238	224.0.1.129	PTPv2	86
2987	01:27:51.404359	192.168.1.238	224.0.1.129	PTPv2	86

FIGURE 5.12: Capture issue de la fusion des deux enregistrements précédents

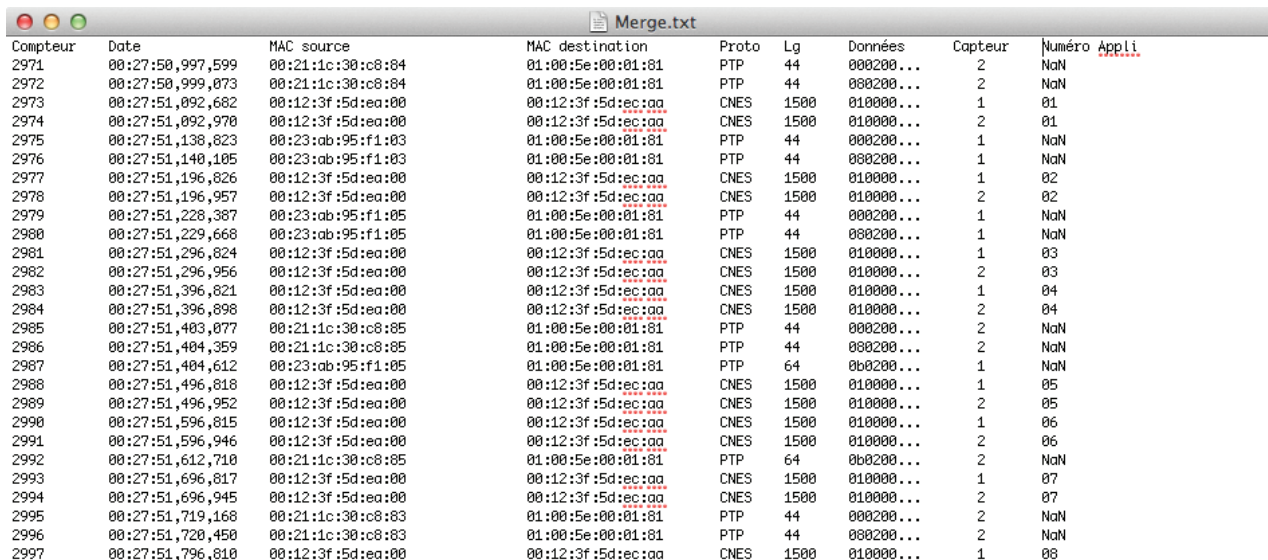
entre les commutateurs et les enregistreurs (par rapport à la capacité des autres liens).

Cette expérience montre donc la faisabilité de la fusion de différentes collectes dès que la condition de précision de synchronisation est respectée. Bien que nous avons expérimenté la fusion sur une architecture de synchronisation hybride respectant la condition de précision de synchronisation, la précision de synchronisation serait d'autant plus performante dans une architecture totalement *hardware* (c'est-à-dire implémentant un estampillage de bas niveau). D'autre part, si cette fonctionnalité d'enregistrement est déportée à terme dans les commutateurs (compatibles IEEE 1588) alors cette condition sera de fait respectée et l'observabilité sera cohérente vis-à-vis de la réalité (quelque soient les conditions).

Dans cette expérience, il n'a pas été mis en œuvre la technique d'envoi des traces des différents points de collecte sur la station enregistreur chargée de la fusion. Cet envoi sera réalisé bien entendu par le réseau. Par conséquent, il sera important de réfléchir si cet envoi doit être (ré)observé ou non (dans ce cas, un filtrage sera nécessaire), à quelle fréquence doit il être réalisé (qui sera alors un compromis entre la taille mémoire et l'impact de la charge sur le réseau), avec quel niveau de priorité, etc.

Les différents paquets ont été capturés à des points de collecte différents. Or cette information ne se retrouve pas directement dans le fichier de fusion. Les déductions ont été faites grâce au fichier de fusion et des différents fichiers provenant de chaque point de collecte. Ainsi, à terme il serait important d'ajouter un paramètre de localisation du point de collecte (par exemple, le

nom du commutateur qui a effectué la capture). Il serait alors plus facile de tracer les différents paquets, d'autant plus que ce sera la trace issue de la fusion qui sera conservée sur la station enregistreur (voire envoyée au sol). De ces différentes informations (contenues dans la fusion des collectes), il sera alors possible d'obtenir un fichier trace (cf FIGURE 5.13) dans le même esprit que celui actuel (cf trace dans le chapitre 2).



Compteur	Date	MAC source	MAC destination	Proto	Lg	Données	Capteur	Numéro	Appli
2971	00:27:50,997,599	00:21:1c:30:c8:84	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2972	00:27:50,999,073	00:21:1c:30:c8:84	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2973	00:27:51,092,682	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	01	
2974	00:27:51,092,970	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	01	
2975	00:27:51,138,823	00:23:ab:95:f1:03	01:00:5e:00:01:81	PTP	44	000200...	1	NaN	
2976	00:27:51,140,105	00:23:ab:95:f1:03	01:00:5e:00:01:81	PTP	44	000200...	1	NaN	
2977	00:27:51,196,826	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	02	
2978	00:27:51,196,957	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	02	
2979	00:27:51,228,387	00:23:ab:95:f1:05	01:00:5e:00:01:81	PTP	44	000200...	1	NaN	
2980	00:27:51,229,668	00:23:ab:95:f1:05	01:00:5e:00:01:81	PTP	44	000200...	1	NaN	
2981	00:27:51,296,824	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	03	
2982	00:27:51,296,956	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	03	
2983	00:27:51,396,821	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	04	
2984	00:27:51,396,898	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	04	
2985	00:27:51,403,077	00:21:1c:30:c8:85	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2986	00:27:51,404,359	00:21:1c:30:c8:85	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2987	00:27:51,404,612	00:23:ab:95:f1:05	01:00:5e:00:01:81	PTP	64	000200...	1	NaN	
2988	00:27:51,496,818	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	05	
2989	00:27:51,496,952	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	05	
2990	00:27:51,596,815	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	06	
2991	00:27:51,596,946	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	06	
2992	00:27:51,612,710	00:21:1c:30:c8:85	01:00:5e:00:01:81	PTP	64	000200...	2	NaN	
2993	00:27:51,696,817	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	07	
2994	00:27:51,696,945	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	07	
2995	00:27:51,719,168	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2996	00:27:51,720,450	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN	
2997	00:27:51,796,810	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	08	

FIGURE 5.13: Fichier de trace issue de la fusion des deux enregistrements précédents

Par exemple, si nous reprenons l'exemple des trames No.2977 et No.2978 (comme dans la partie précédente), nous pouvons voir que la trame N°2977 a bien été capturée sur le point de collecte 1 et la trame No.2978 sur le point de collecte 2. De plus, il a été affiché dans ce fichier de trace le numéro de paquet applicatif (seulement pour le trafic généré représentant le trafic applicatif). Ainsi, il est possible d'observer qu'il s'agit bien du même flux capturé à deux endroits différents. Par conséquent, ces différentes informations nous permettent d'affirmer que l'observation de ce message tout au long de son chemin est cohérente vis-à-vis de la réalité. Notons que pendant la collecte tous les octets de la trame sont capturés ce qui nous permet d'afficher ce que l'on désire dans ce fichier de trace (le fichier de trace fourni à la FIGURE 5.13 n'est qu'un exemple de présentation possible).

D'autre part, il sera important de choisir comment seront envoyées les différentes collectes sur la station enregistreur. En effet, cette collecte sera réalisée de façon distribuée sur les différents commutateurs (et/ou TAP supplémentaires) qu'il sera nécessaire de centraliser sur la station enregistreur en vue de la fusion, la compression, etc. Ainsi, il sera nécessaire de choisir la date d'envoi de la collecte. Celle-ci pourra être choisie en fonction du volume de la trace, d'une fréquence prédéfinie, du taux d'utilisation du réseau, etc. D'autre part, comme cette collecte sera envoyée par le réseau lui-même, il sera également nécessaire de choisir un niveau de priorité

(Classification de Service) pour ce type de flux.

5.3 Isolation des défauts

Il a été vu précédemment que l'observabilité des échanges est nécessaire pour vérifier de manière déterministe l'état des communications que ce soit en test au sol ou bien au retour de mission. Dans le cas où une défaillance intervient, il convient d'isoler le défaut (cf E. 7) de manière à pouvoir écarter (ou au contraire incriminer) le réseau. Comme la surveillance continue de l'état du réseau est requise, les méthodes d'observabilité et de visualisation de différents objets de la MIB (*via* les agents SNMP) vont être utilisées.

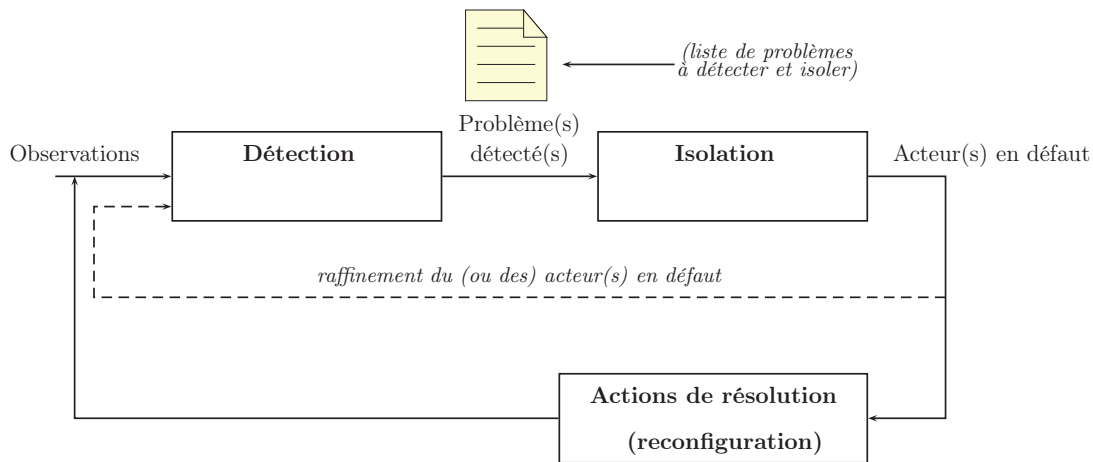


FIGURE 5.14: Chaîne de résolution de défaillances

L'isolation de défauts consiste en la recherche du (ou des) acteurs responsables d'un problème (une anomalie) qui aurait été auparavant détecté(e).

Cette définition est illustrée à la FIGURE 5.14 et met en évidence qu'il est important de lister a priori quels problèmes doivent être détectés et dont il sera nécessaire de rechercher l'origine. A partir de cette liste de problèmes, il sera alors possible d'utiliser des mécanismes adéquats d'observabilité de la dynamique des communications (au sens où l'on en parle dans le chapitre précédent) et/ou de l'état statique du système (au sens de l'instrumentation du réseau, avec des agents SNMP par exemple).

Dès lors que ces choix d'instrumentation sont réalisés, les observations obtenues grâce à cette instrumentation du réseau sont donc le point d'entrée de la fonction de détection qui permettra de déterminer (détecter) le problème (que nous aurons listé au préalable pour l'isolation). Ce problème deviendra à son tour l'entrée de la fonction isolation (Pourquoi ce problème a été détecté ? est-ce le réseau ou non ?) qui aura pour mission d'isoler la (ou les) cause(s) du problème. Cette cause pourra être raffinée en fonction du niveau nécessaire ou désiré : par exemple, est-

ce suffisant de dire qu'un port d'un commutateur est défaillant ? ou est-il nécessaire de savoir pourquoi celui-ci est défaillant ? Ces différentes interrogations montrent qu'il est important de délimiter le périmètre du réseau. *D'un point de vue physique, nous nous attacherons à arrêter cette frontière à la carte de communication des équipements terminaux.* Celle-ci est importante pour pouvoir incriminer (ou non) le réseau dont la fonction première est le transport de données selon des critères de Qualité de Service (borne temporelle, pertes, intégrité, ...). Le réseau est incriminé dans le cas où il ne rend pas le service attendu même si une cause externe n'est pas à exclure (telle que par exemple une défaillance d'alimentation qui entraînerait une défaillance de la carte de communication sur un équipement terminal et de ce fait, une défaillance réseau).

5.3.1 Expérimentation 1

Reprenons l'exemple traité dans la section précédente (cf. FIGURE 5.15). Nous nous intéressons au problème de non-acheminement du trafic applicatif. Pour observer ce problème, l'observabilité devrait se faire de bout en bout mais peut être potentiellement difficile (dans le cas où par exemple, l'équipement ne serait plus atteignable). Ainsi, il est possible d'observer par anticipation la dynamique des communications (traces obtenues par *port mirroring*) ou l'état statique du système (visualisation des différents objets de la MIB *via* les agents SNMP). L'objectif est de monter la méthode d'isolation du problème. Pour cela, nous simulons une rupture de lien terminal en débranchant le câble sur le commutateur 2 entre celui-ci et la station Destinataire Charge.

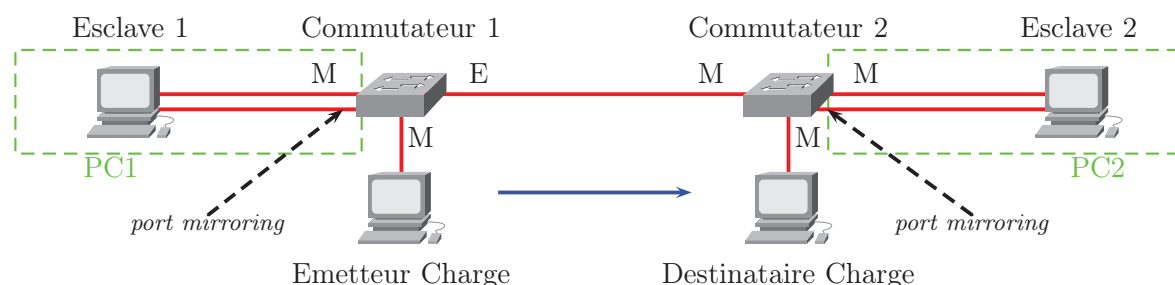


FIGURE 5.15: Architecture de test pour l'isolation de défauts

Dans cet exemple, sur la station terminale « Destinataire charge », un programme permet d'afficher le numéro de paquet (applicatif) reçu. A un instant donné, la station ne reçoit plus de message. Le dernier numéro de paquet affiché est 3157 (numéro applicatif hexadécimal : 0c54). Notons que ces informations auraient pu être datées de la même façon que les collectes le sont. Cette première observation permet d'identifier la liaison en défaut. Deux causes sont alors possibles : soit le réseau est incriminé soit la station terminale est défaillante. Dans cette expérience, la station terminale n'était pas défaillante, le réseau n'a donc pas rendu le service attendu. Néanmoins, il n'est pas possible avec cette seule observation d'isoler la cause du problème

de non-acheminement du trafic applicatif. Notons que cette observation peut être considérée comme redondante avec celle de la dynamique des communications dans le sens où si un message est vu à la « sortie » d'un commutateur alors il peut être considéré que ce message est reçu par le destinataire (reçu ne signifiant pas qu'il n'y ait pas un problème d'intégrité éventuel ou un non traitement par l'équipement terminal). Il s'agira finalement d'une observabilité par anticipation.

Grâce à l'observabilité de la dynamique des communications (cf chapitre précédent), nous obtenons le fichier de trace présenté à la FIGURE 5.16 issu de la fusion des deux enregistrements réalisés grâce au *port mirroring* sur les deux enregistreurs (PC1 et PC2).

Compteur	Date	MAC source	MAC destination	Proto	Lg	Données	Capteur	Numéro Appli
8000	00:20:0,637,614	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c4d
8001	00:20:0,637,819	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c4e
8002	00:20:0,637,863	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c4e
8003	00:20:0,644,330	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
8004	00:20:0,645,570	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c4f
8005	00:20:0,645,614	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c4f
8006	00:20:0,645,618	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
8007	00:20:0,645,819	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c50
8008	00:20:0,645,863	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c50
8009	00:20:0,653,571	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c51
8010	00:20:0,653,615	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c51
8011	00:20:0,661,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c52
8012	00:20:0,661,612	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c52
8013	00:20:0,661,818	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c53
8014	00:20:0,662,651	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c53
8015	00:20:0,669,622	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c54
8016	00:20:0,669,665	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	0c54
8017	00:20:0,670,238	00:23:ab:95:f1:05	01:00:5e:00:01:81	PTP	64	000200...	1	NaN
8018	00:20:0,677,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c55
8019	00:20:0,677,818	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c56
8020	00:20:0,685,570	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c57
8021	00:20:0,685,819	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c58
8022	00:20:0,693,570	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c59
8023	00:20:0,701,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5a
8024	00:20:0,701,818	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5b
8025	00:20:0,709,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5c
8026	00:20:0,717,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5d
8027	00:20:0,717,818	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5e
8028	00:20:0,725,568	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c5f
8029	00:20:0,725,817	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c60
8030	00:20:0,733,569	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	0c61

FIGURE 5.16: Fichier de trace issue de la fusion des deux enregistrements sur les deux enregistreurs

Cette trace permet d'observer que le trafic n'est plus acheminé à partir de la trame No.8018 (numéro applicatif : 0c55). Bien que cette observation soit faite sur le capteur 2, il n'est pas possible d'isoler la cause de la défaillance. En effet, cette non-retransmission par le commutateur 2 (dû au choix de ne copier que les paquets sortants) peut être due à la défaillance d'un port, d'un lien, de la table de commutation, d'un *buffer* plein, ... Dans ce cas de figure, le problème peut bien être détecté mais cette observation (de la dynamique des communications) n'est pas suffisante pour l'isolation et pour démarrer une action de résolution. Ainsi, de manière à pouvoir raffiner la cause de ce problème, il est nécessaire d'instrumenter le réseau. Dans cet exemple, il peut être envisagé de remonter des informations de l'état des ports (réalisé dans la solution de reconfiguration) et de la table de commutation grâce au protocole SNMP. Notons que ces informations pourront être datées de la même façon que les collectes le sont. Ces informations sont données dans les FIGURE 5.17 et 5.18.

```

IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.10001 = INTEGER: up(1)
IF-MIB::ifOperStatus.10002 = INTEGER: up(1)
IF-MIB::ifOperStatus.10003 = INTEGER: up(1)
IF-MIB::ifOperStatus.10004 = INTEGER: down(2)
IF-MIB::ifOperStatus.10005 = INTEGER: down(2)
IF-MIB::ifOperStatus.10006 = INTEGER: down(2)
IF-MIB::ifOperStatus.10007 = INTEGER: down(2)
IF-MIB::ifOperStatus.10008 = INTEGER: down(2)
IF-MIB::ifOperStatus.10101 = INTEGER: down(2)
IF-MIB::ifOperStatus.10102 = INTEGER: down(2)
IF-MIB::ifOperStatus.10501 = INTEGER: up(1)
...
BRIDGE-MIB::dot1dStaticAddress.'...'.0 = Hex-STRING: 00 10 18 5F C9 81
BRIDGE-MIB::dot1dStaticAddress.'...eG'.0 = Hex-STRING: 00 10 18 65 47 A8
BRIDGE-MIB::dot1dStaticAddress.'...?']..'0 = Hex-STRING: 00 12 3F 5D EA 00
BRIDGE-MIB::dot1dStaticAddress.'...?']..'0 = Hex-STRING: 00 12 3F 5D EC AA
BRIDGE-MIB::dot1dStaticReceivePort.'...'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...eG'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...'.0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...eG'.0 = STRING: " "
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...?']..'0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...?']..'0 = Hex-STRING: 08
BRIDGE-MIB::dot1dStaticStatus.'...'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...eG'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...?']..'0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...?']..'0 = INTEGER: permanent(3)
-----

```

FIGURE 5.17: Informations d'état des ports et de la table de commutation obtenues par SNMP sur le commutateur 2 (Avant défaillance)

Ainsi, nous pouvons observer que la table de commutation est cohérente. En revanche, nous observons que l'état du port 3 a été modifié ce qui mène à dire que la cause peut être une défaillance de la station terminale, du câble ou du port. Bien que cette isolation n'est pas la plus fine dans le sens où il n'y a pas qu'une seule cause possible, elle permet d'isoler la liaison réseau en défaut.

Dans ce premier exemple, nous pouvons d'ores-et-déjà conclure qu'une méthode d'observabilité seule (de bout en bout ou de la dynamique des communications ou de l'état statique du système) ne permet pas d'isoler la cause d'un problème. En revanche, une des méthodes d'observabilité (quelle qu'elle soit) peut permettre à elle seule d'incriminer le réseau.

5.3.2 Expérimentation 2

Prenons à présent un exemple sur lequel une rupture de câble redondant est simulé. L'architecture de test reprend l'architecture de l'exemple précédent à laquelle il a été ajouté un

```

IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.10001 = INTEGER: up(1)
IF-MIB::ifOperStatus.10002 = INTEGER: up(1)
IF-MIB::ifOperStatus.10003 = INTEGER: down(2)
IF-MIB::ifOperStatus.10004 = INTEGER: down(2)
IF-MIB::ifOperStatus.10005 = INTEGER: down(2)
IF-MIB::ifOperStatus.10006 = INTEGER: down(2)
IF-MIB::ifOperStatus.10007 = INTEGER: down(2)
IF-MIB::ifOperStatus.10008 = INTEGER: down(2)
IF-MIB::ifOperStatus.10101 = INTEGER: down(2)
IF-MIB::ifOperStatus.10102 = INTEGER: down(2)
IF-MIB::ifOperStatus.10501 = INTEGER: up(1)
...
BRIDGE-MIB::dot1dStaticAddress.'...'.0 = Hex-STRING: 00 10 18 5F C9 81
BRIDGE-MIB::dot1dStaticAddress.'...eG'.0 = Hex-STRING: 00 10 18 65 47 A8
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EA 00
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EC AA
BRIDGE-MIB::dot1dStaticReceivePort.'...'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...eG'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...'.0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...eG'.0 = STRING: " "
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 08
BRIDGE-MIB::dot1dStaticStatus.'...'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...eG'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
-----

```

FIGURE 5.18: Informations d'état des ports et de la table de commutation obtenues par SNMP sur le commutateur 2 (Après défaillance)

lien redondant entre les commutateurs. Dans cette expérience, la solution de reconfiguration topologique présentée dans le chapitre 4 est implémentée sur l'Esclave 1.

Un extrait du fichier de trace obtenu grâce à l'observabilité de la dynamique des communications est présenté à la FIGURE 5.19.

A partir de ce fichier, nous observons qu'une trame applicative (numéro applicatif : 00fa) n'a pas été observée ni sur le commutateur 1 ni sur le commutateur 2. Comme les trames sont bien envoyées au niveau de la station source et avec la connaissance que le *port mirroring* ne capture que les trames sortantes, il y a eu un problème réseau entre les commutateurs sans pouvoir l'isoler plus précisément. De la même façon que précédemment, l'observabilité de la dynamique des communications à elle seule ne suffit pas à isoler le problème. En effet, vu que le trafic n'est pas totalement interrompu (seule une trame a été perdue), il est nécessaire de connaître l'état du réseau grâce à l'observabilité de l'état statique du système. Les informations obtenues grâce à SNMP sont données dans les FIGURE 5.20 et 5.21. Notons que ces informations pourront

Compteur	Date	MAC source	MAC destination	Proto	Lg	Données	Capteur	Numéro Appli
1655	05:23:39,318,529	00:23:ab:95:f1:83	01:00:5e:00:01:81	PTP	44	000200...	1	NaN
1656	05:23:39,317,349	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1657	05:23:39,326,10	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1658	05:23:39,336,016	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1659	05:23:39,340,100	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	00f9
1660	05:23:39,340,286	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	00f9
1661	05:23:39,346,024	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1662	05:23:39,349,8	00:21:1c:30:c8:84	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1663	05:23:39,350,292	00:21:1c:30:c8:84	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1664	05:23:39,356,013	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1665	05:23:39,366,221	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1666	05:23:39,376,053	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1667	05:23:39,385,900	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1668	05:23:39,396,016	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1669	05:23:39,406,015	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1670	05:23:39,416,030	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1671	05:23:39,426,016	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1672	05:23:39,446,989	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1673	05:23:39,449,478	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1674	05:23:39,450,953	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1675	05:23:39,451,658	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	78	304c02...	1	NaN
1676	05:23:39,455,528	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	183	3081b4...	1	NaN
1677	05:23:39,523,106	00:21:1c:30:c8:83	01:00:5e:00:01:81	PTP	54	090200...	2	NaN
1678	05:23:39,540,156	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	00fb
1679	05:23:39,540,226	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	00fb
1680	05:23:39,632,913	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	44	302a02...	1	NaN
1681	05:23:39,634,7	00:21:1c:30:c8:85	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1682	05:23:39,634,933	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1683	05:23:39,635,482	00:21:1c:30:c8:85	01:00:5e:00:01:81	PTP	44	000200...	2	NaN
1684	05:23:39,636,606	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1685	05:23:39,638,200	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1686	05:23:39,639,955	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1687	05:23:39,640,205	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	1	00fc
1688	05:23:39,640,223	00:12:3f:5d:ea:00	00:12:3f:5d:ec:aa	CNES	1500	010000...	2	00fc
1689	05:23:39,641,682	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1690	05:23:39,646,444	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN
1691	05:23:39,648,125	00:23:ab:95:f1:40	00:10:18:5f:c9:81	SNMP	45	302b02...	1	NaN

FIGURE 5.19: Fichier de trace issue de la fusion des deux enregistrements sur les deux enregistreurs (expérience utilisant la fonction de reconfiguration)

être datées de la même façon que les collectes le sont de manière à facilement corréler les deux observations. Il est important de noter que dans ce cas de figure, un problème réseau ne remet pas toujours en cause l'application.

Ces informations montrent que le port 2 (noté 10002 dans ce produit Cisco) est dans l'état *down* signifiant qu'il y a une défaillance soit du port 2 d'un (ou des deux) commutateurs ou bien que le câble est défaillant ou encore qu'il y ait eu une perturbation transitoire. Il est alors confirmé que le problème vient bien du réseau et la liaison en défaut est isolée. Cependant, l'application est toujours fonctionnelle puisque la fonction de reconfiguration a modifié la table de commutation de manière à transporter l'information non plus par le port 2 (noté 10 dans le champ `dot1StaticAllowedToGoTo`) mais par le port 4 (noté 04 dans le même champ). Ainsi, nous connaissons l'état de la topologie logique après avoir détecté et isolé le problème.

5.4 Conclusion

La fonctionnalité d'observabilité sur le système de communication d'un lanceur est une exigence capitale. L'objectif est de connaître de manière déterministe l'état du réseau à travers la réalisation de traces de vol. Ces traces permettent notamment de jouer (ou de rejouer) un vol au

```

-----
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.10001 = INTEGER: up(1)
IF-MIB::ifOperStatus.10002 = INTEGER: up(1)
IF-MIB::ifOperStatus.10003 = INTEGER: up(1)
IF-MIB::ifOperStatus.10004 = INTEGER: up(1)
IF-MIB::ifOperStatus.10005 = INTEGER: down(2)
IF-MIB::ifOperStatus.10006 = INTEGER: down(2)
IF-MIB::ifOperStatus.10007 = INTEGER: down(2)
IF-MIB::ifOperStatus.10008 = INTEGER: down(2)
IF-MIB::ifOperStatus.10101 = INTEGER: down(2)
IF-MIB::ifOperStatus.10102 = INTEGER: down(2)
IF-MIB::ifOperStatus.10501 = INTEGER: up(1)
...
BRIDGE-MIB::dot1dStaticAddress.'...'.0 = Hex-STRING: 00 10 18 5F C9 81
BRIDGE-MIB::dot1dStaticAddress.'...eG'.0 = Hex-STRING: 00 10 18 65 47 A8
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EA 00
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EC AA
BRIDGE-MIB::dot1dStaticReceivePort.'...'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...eG'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...'.0 = STRING: " "
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...eG'.0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 08
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 10
BRIDGE-MIB::dot1dStaticStatus.'...'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...eG'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
-----

```

FIGURE 5.20: Informations d'état des ports et de la table de commutation obtenues par SNMP sur le commutateur 1 (Avant défaillance)

sol que ce soit pour les tests préalables au vol ou d'une analyse après vol.

Cette fonctionnalité est facilement réalisée sur le système de communication d'Ariane 5 puisqu'il s'appuie sur un bus où l'ensemble des échanges est vu en tout point du réseau grâce au mécanisme de diffusion. En revanche, dans le cas d'architectures commutées, cette fonctionnalité est plus complexe. Ainsi, ce chapitre a permis de montrer la faisabilité d'une telle fonctionnalité sur des architectures commutées en mettant en place des observateurs locaux sur les commutateurs de manière à reconstruire la trace. Nous avons mis en évidence que les points de collecte devaient être synchronisés (avec une précision inférieure à un temps de transmission) pour assurer la cohérence de la trace. Pour obtenir une précision de synchronisation inférieure à un temps de transmission entre les différents points de collecte, les tests préliminaires ont mis en évidence la nécessité de s'appuyer sur une synchronisation matérielle. Le passage à l'échelle nécessite la prise en compte de l'incertitude liée à la cascade d'équipements. En effet, si cette incertitude

```

-----
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.10001 = INTEGER: up(1)
IF-MIB::ifOperStatus.10002 = INTEGER: down(2)
IF-MIB::ifOperStatus.10003 = INTEGER: up(1)
IF-MIB::ifOperStatus.10004 = INTEGER: up(1)
IF-MIB::ifOperStatus.10005 = INTEGER: down(2)
IF-MIB::ifOperStatus.10006 = INTEGER: down(2)
IF-MIB::ifOperStatus.10007 = INTEGER: down(2)
IF-MIB::ifOperStatus.10008 = INTEGER: down(2)
IF-MIB::ifOperStatus.10101 = INTEGER: down(2)
IF-MIB::ifOperStatus.10102 = INTEGER: down(2)
IF-MIB::ifOperStatus.10501 = INTEGER: up(1)
...
BRIDGE-MIB::dot1dStaticAddress.'...'.0 = Hex-STRING: 00 10 18 5F C9 81
BRIDGE-MIB::dot1dStaticAddress.'...eG'.0 = Hex-STRING: 00 10 18 65 47 A8
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EA 00
BRIDGE-MIB::dot1dStaticAddress.'..?']..'0 = Hex-STRING: 00 12 3F 5D EC AA
BRIDGE-MIB::dot1dStaticReceivePort.'...'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'...eG'.0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticReceivePort.'..?']..'0 = INTEGER: 0
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...'.0 = STRING: " "
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'...eG'.0 = Hex-STRING: 04
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 08
BRIDGE-MIB::dot1dStaticAllowedToGoTo.'..?']..'0 = Hex-STRING: 04
BRIDGE-MIB::dot1dStaticStatus.'...'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'...eG'.0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
BRIDGE-MIB::dot1dStaticStatus.'..?']..'0 = INTEGER: permanent(3)
-----

```

FIGURE 5.21: Informations d'état des ports et de la table de commutation obtenues par SNMP sur le commutateur 1 (Après défaillance)

est supérieure au temps de transmission, l'observabilité mis en avant dans ce mémoire ne pourra plus être appliquée.

Bien qu'une trace peut donner des informations sur les acteurs potentiellement défaillants, celle-ci est nécessaire mais pas toujours suffisante pour isoler un défaut. Par conséquent, nous nous sommes appuyés sur d'autres informations telles que l'état des ports, la table de commutation, etc (obtenus notamment par l'agent central de la solution de la reconfiguration) pour montrer la faisabilité d'une isolation de défauts. Celle-ci a été illustrée sur l'exemple de non transport de l'information (perte de trame) mais d'autres problèmes auraient pu être considérés selon la profondeur de l'isolation attendue.

Au terme de ce chapitre, nous avons traité les principales exigences auxquelles devait répondre le système de communication basé sur la technologie Ethernet. Néanmoins, d'autres critères (comme les risques d'embargo, le développement de matériel spécifique pouvant résister aux

radiations, ...) peuvent entrer en ligne de compte dans le choix final d'utiliser cette technologie comme système de communication pour les lanceurs.

Conclusions & Perspectives

La première partie de cette thèse a permis de décrire le système de communication actuellement embarqué dans les lanceurs spatiaux (tels que les lanceurs type Ariane 5). Il met en évidence les caractéristiques et les limitations de ce type de bus vis-à-vis des nouveaux besoins pour la future génération de lanceurs. Il a été mis en parallèle l'évolution des réseaux industriels, qui pour les mêmes problématiques, se sont tournés vers le standard Ethernet : un bref état de l'art a été présenté pour mettre en avant cette évolution. De manière à mettre en évidence que cette même orientation peut être cohérente dans le domaine spatial, une première étude de performances a été réalisée sur quatre solutions industrielles. Suite à celle-ci, il a réalisé une étude approfondie d'Ethernet pour le domaine des lanceurs spatiaux.

Le deuxième chapitre présente les exigences que doit respecter le futur système de communication des lanceurs. Une première étude de cas permet de proposer des architectures de communication pour les lanceurs. Cette étude permet de montrer la viabilité d'Ethernet dans le contexte des lanceurs : tout en conservant le comportement intrinsèquement déterministe de l'application, il a été réalisé une première évaluation de l'exigence temps-réel de manière théorique et par simulation. Par la suite, les travaux de cette thèse ont permis de proposer des solutions avec pour objectif le respect des différentes exigences.

L'évaluation temporelle présentée dans le chapitre 3 se réfère à la théorie du calcul réseau. Ce chapitre a permis de mettre en avant le gain de la distribution sur les majorants du délais de bout en bout.

Une autre caractéristique importante est que le système de communication d'un lanceur doit être disponible. Ainsi, le chapitre 4 propose une solution de reconfiguration temps-réel s'appuyant sur la « redondance dans le réseau » de la seconde architecture proposée dans le chapitre 2. Cette solution consiste à remplir/modifier la table de commutation des différents commutateurs en fonction de l'évolution du graphe réseau (provoqué par les défaillances). Cette évolution est observée périodiquement.

Enfin, le chapitre 5 propose une solution pour l'observabilité d'un réseau Ethernet commuté et l'isolation de défauts. A travers l'utilisation d'observateurs locaux sur les commutateurs (synchronisés par le protocole PTP), il s'agit de retrouver une trace globale des échanges sur le réseaux. Différentes expériences ont permis de montrer la faisabilité de la solution.

Perspectives

Tout au long de cette thèse, il a été proposé des solutions « unitaires » sur des exigences particulières. Il a été mis en avant dans le chapitre 5, la faisabilité de l'observabilité et l'isolation de défauts sur un réseau Ethernet commuté. Néanmoins, ces deux fonctionnalités ne peuvent être réalisées sans la génération de trafic supplémentaire. En effet, le réseau doit générer du trafic pour assurer la synchronisation des commutateurs (nécessaire à la collecte des différents échanges), le transfert des fichiers de trace, l'envoi d'informations de supervision du réseau ou encore le trafic inhérent à la détection/reconfiguration de défaillances (proposée dans le chapitre 4). D'autre part, les informations de télémétrie pourront également circuler sur le même réseau. Ainsi, tous ces trafics devront alors être pris en compte lors de la conception/qualification de l'architecture pour les lanceurs. En effet, l'évaluation temporelle réalisée grâce au *network calculus* ne prenait en considération que le trafic fonctionnel.

Dans le cas d'une augmentation de la taille de l'architecture réseau (passage à l'échelle), la méthodologie basée sur la théorie du calcul réseau est capable de déterminer des majorants des délais. Cela peut amener à une sur-estimation des majorants des délais (pessimisme plus important) qui peut avoir une conséquence sur l'atteignabilité des pires majorants. Le temps de calcul sera également plus conséquent. Dans le cas des lanceurs, le nombre d'équipements reste relativement peu important (la tendance est d'ailleurs à la diminution), ce problème reste donc limité. D'autre part, les différentes défaillances vont entraîner la reconfiguration des chemins selon la méthode présentée au chapitre 4. Il sera alors nécessaire pour l'ingénieur d'évaluer les majorants des délais pour l'ensemble des chemins possibles afin de vérifier qu'ils restent inférieurs à la contrainte de délai définie par l'application.

D'un point de vue disponibilité, nous avons proposé deux solutions qui ont toutes les deux leurs avantages et leurs inconvénients. Aussi pour augmenter la disponibilité, la combinaison des deux approches peut être intéressante. Cependant, les chemins sont définis différemment dans les deux approches relativement à des métriques différentes : le nombre de sauts d'une part et le nombre d'équipements et la fiabilité d'autre part. Dans certains cas, l'indépendance des chemins ne sera pas assurée du fait de la minimisation du nombre de sauts. Une piste à travailler pourra s'attacher à identifier une métrique quantifiant l'indépendance des chemins en fonction des informations d'états collectées de manière centralisée ou distribuée et de la probabilité ou du taux de défaillances des équipements.

Pour l'observabilité, il a été mis en exergue la nécessité d'utiliser une synchronisation matérielle de manière à recouvrer une précision de synchronisation suffisante à la cohérence des informations collectées et fusionnées. Néanmoins, il serait intéressant d'étudier la prise en compte de l'incertitude lié à la cascade d'équipements lors du passage à l'échelle sur une architecture plus importante. De la même façon, l'augmentation de la taille de l'architecture aura pour consé-

quence d'augmenter le nombre de points de collecte et le volume de données d'états. Ce volume de données pouvant complexifier la recherche de défauts devra être traité.

Tout au long de ces travaux, seules des expériences unitaires de chacune des fonctions ont été réalisées. Il serait intéressant d'intégrer ces différentes briques sur une même architecture de test. Cela consiste alors à étudier de manière commune et intégrée l'ensemble des trafics (fonctionnels, supervision réseau, synchronisation, reconfiguration réseau et éventuels non fonctionnels et métrologie) avec les fonctionnalités d'observabilité et d'isolation de défaillances. En effet, ces travaux ont permis de montrer la faisabilité d'un point de vue méthodologique du respect à différentes exigences sans prendre en compte la commande qui est encore à ce jour inconnue. Par conséquent, il s'agirait de réaliser une co-conception du système de communication avec les spécialistes de la commande notamment.

L'intégration des différents travaux proposés devront être réalisés sur un démonstrateur. Dans cet objectif, le projet Avionique-X (Monchaux et al., 2012), financé par un Plan d'Investissement pour l'Avenir (PIA) avec le CNES comme maître d'ouvrage et ASTRIUM Space Transportation comme maître d'œuvre, a vu le jour. La première étape de ce projet consiste à choisir la (ou les) technologies à mettre en œuvre sur cette plateforme de démonstration. Pour cela, nous avons participé en juillet 2012 à l'évaluation de plusieurs candidats (dont Ethernet commuté standard qui a été traité dans cette thèse). Nous avons été retenus comme experts académiques au côté d'autres groupes de notation composés d'industriels du secteur spatial. Cette évaluation se présentait sous la forme d'une grille de notation composée de 22 critères. Celle-ci (présentée en Annexe) est le début de notre participation au projet. Cette évaluation a fait émerger TT-Ethernet (*Time-Triggered Ethernet*) et AFDX (*Avionics Full-Duplex switched Ethernet*) comme solution à mettre en œuvre sur ce démonstrateur. Les travaux réalisés dans cette thèse pourront néanmoins être utilisés.

D'un point de vue disponibilité, nous avons suggéré dans cette thèse que la fusion des deux solutions proposées (toutes deux dans la catégorie « redondance dans le réseau ») pourrait augmenter la disponibilité du système de communication. Or, TT-Ethernet et AFDX s'appuient sur la duplication de trames sur deux réseaux distincts (« redondance du réseau »). La fusion des deux ne pourrait donc pas être appliquée directement. Néanmoins, notre première solution de reconfiguration se basant sur la détection de la défaillance par scrutation périodique de l'état des ports des commutateurs pourra être mise en œuvre sur ces deux réseaux distincts. Ainsi, bien que la défaillance sur un réseau est tolérée par l'utilisation d'une trame redondante (acheminée par le réseau redondant), il est intéressant de reconfigurer ce réseau de manière à être robuste à plus d'une défaillance (comme suggéré dans la fusion de nos deux propositions).

Pour l'observabilité, il est tout à fait envisageable de mettre en œuvre notre proposition sur TT-Ethernet et l'AFDX dans la mesure où les deux réseaux s'appuient sur une architecture

commutée. Une piste à travailler est d'identifier dans quelles conditions la technique de synchronisation utilisée dans TT-Ethernet pourra être la remplaçante du protocole PTP employé dans notre proposition. Plus globalement, comme évoqué dans le chapitre 5 pour de l'Ethernet standard, l'identification de la charge du trafic d'observation (bande passante nécessaire) sera nécessaire. Cela se caractérisera par ailleurs à la définition des paramètres d'un canal virtuel dans AFDX.

Cette thèse rentre dans l'histoire d'Ethernet : des îles ... aux lanceurs spatiaux en passant par le bureau, puis l'atelier (Divoux et al., 2008), l'industrie et l'aéronautique...

* * *

Annexe A

Evaluation de candidats concurrents en vue d'un démonstrateur

A.1 Introduction

Tout au long de cette thèse, nous nous sommes attachés à répondre aux différentes exigences présentées au chapitre 2 auxquelles la technologie de remplacement du système de communication actuel doit répondre. Ainsi, il a été proposé une première qualification d'architectures Ethernet commutées pour les lanceurs, qui a permis de montrer le respect de contraintes temporelles des flux fonctionnels sur cette technologie. Elle a également permis de mettre en avant le gain de la distribution de la commande sur les valeurs de majorants de délais ce qui ouvrait la voie à des évolutions futures comme le transport de flux non fonctionnels. Bien que l'aspect temporel est un point crucial, la disponibilité du réseau est également importante : nous avons donc proposé une technique de reconfiguration lorsqu'une défaillance est détectée. Ensuite, l'originalité des systèmes spatiaux est la nécessité d'observer l'ensemble des échanges. Malgré la complexité d'une telle fonctionnalité lorsqu'une architecture segmentée est utilisée, nous avons proposé la mise en place d'observateurs locaux sur les commutateurs de manière à reconstituer une vue globale. Cette observabilité a permis notamment de présenter une méthode d'isolation de défaut.

Bien que les travaux présentés répondent aux exigences, il n'en reste pas moins des améliorations possibles. En effet, nous avons vu par exemple que la solution de reconfiguration est centralisée ce qui peut provoquer une défaillance majeure si celle-ci est elle-même défaillante : une étude de la distribution pourrait alors être menée. L'évaluation des architectures n'est qu'une pré-qualification puisqu'elle ne tient compte que des flux fonctionnels et que le choix de la distribution, de l'applicatif, de la télémétrie, etc ne sont pas encore définis. Ce dernier point met en avant qu'il est notamment nécessaire d'intégrer ces différentes fonctionnalités. Cette intégration pourrait être réalisée sur un démonstrateur.

Le projet de démonstrateur Avionique-X (Monchaux *et al.*, 2012) qui est financé par un Plan d'Investissement pour l'Avenir (PIA) avec le CNES comme maître d'ouvrage et ASTRIUM Space Transportation comme maître d'œuvre, pourrait faire l'objet de cette intégration. En revanche, les acteurs de ce projet n'ont pas encore choisi la technologie à mettre en œuvre sur cette plateforme de démonstration. Dans cette optique, nous avons participé à l'évaluation de plusieurs candidats (dont Ethernet commuté standard qui a été traité dans cette thèse) au même titre que d'autres groupes de notation composés d'industriels du secteur spatial.

Ce chapitre présente l'évaluation que nous avons réalisée. Celle-ci permet de situer nos travaux par rapport aux critères de sélection et permet de mettre en regard les perspectives.

A.2 Listes des critères de sélection

L'évaluation des différents candidats repose sur 22 critères (notés C00 à C21) qu'il conviendra de noter pour chaque candidat (de 0 à 5). Nous présentons ici la liste des critères tels qu'ils sont énoncés dans cette évaluation. Quelques commentaires seront faits à l'issue de la présentation de ces critères et des candidats.

C00 - Priority and ségrégation : La priorisation et la ségrégation des messages peut être considérée comme un critère de haut niveau (non spécifique d'un lanceur) sur lequel il peut être utile de comparer les protocoles candidats.

C01 - Compatibility with a launcher life cycle : Au sol l'avionique doit être mise en œuvre et sert de support à la mise en œuvre du lanceur.

C02 - Robustness to launcher stage jettisoning / reconfiguration : quelles sont les conséquences d'une séparation d'étages en termes de reconfiguration du réseau, désadaptation des lignes... ?

C03 - Integrated control mechanisms : Mécanismes de contrôles intégrés ? Par exemple, existe-t-il une fonctionnalité d'acknowledge ?

C04 - Long-term sustainability : le protocole est-il utilisé dans d'autres domaines et le sera-t-il toujours à l'horizon 2025, quelle est la taille de la communauté d'utilisation ? Il est en effet important d'évaluer les risques d'obsolescence.

C05 - Development tools : existe-t-il des outils pour aider au développement du système de communication, sont-ils nombreux, évolués... ?

C06 - Development effort : quel est l'effort à produire pour pouvoir utiliser le protocole sur lanceur ? Existe-t-il des COTS (connecteurs, switches) et sont-ils compatibles des stratégies ERN ?

C07 - Maturity : le protocole et les COTS associés (s'ils existent) sont-ils déjà utilisés dans d'autres domaines, ont-ils déjà volé sur avions, sur lanceurs ?

C08 - Flexibility, growth potential, and isolation between layers : le protocole est il flexible, peut on ajouter facilement des équipements, reconfigurer le réseau ? Existe-t-il une bonne indépendance entre les couches ce qui permet de faciliter le traitement des obsolescences ? Quel est son potentiel de croissance ? Y a-t-il de la marge vis-à-vis du débit ? Adaptabilité à des protocoles de haut niveau ?

C09 - Qualification effort : est il facile de démontrer le bon fonctionnement du système de communication, d'évaluer les pires cas ? (ce critère n'est pas forcément à évaluer du point de vue de la philosophie actuelle de qualification, mais l'effort de modification de cette philosophie peut entrer en ligne de compte). La qualification est-elle complètement remise en cause en cas d'évolution ?

C10 - Recurring cost : Effort d'AIT, minimisation des coûts de traitement d'obsolescence.

C11 - Robustness to environment (EMC, microcuts...) : le protocole et son support physique sont ils robustes aux EMC, aux modes commun, aux micro-coupures, aux vibrations ?, ERN, pannes. . .

C12 - Energy consumption : selon la nature du coupleur (optique, électrique), son type . . .

C13 - Non Recurring cost : faut il développer des équipements spécifiques ou peut-on en utiliser des déjà existants, dans le second cas sont ils spécifiques du spatial ou le marché est il plus grand ?

C14 - Redundancy : la redondance est elle native dans le protocole ?

C15 - Determinism and jitter

C16 - Communication system observability : L'observabilité du réseau est- elle facile à mettre en œuvre, la quantité de données à retransmettre au sol. . . ? Possibilité de broadcast / multicast ?

C17 - Fault isolation : est il facile de détecter et d'isoler les équipements défectueux (ex : babbling idiot) ?

C18 - Normalization & standardization : le protocole est il normalisé, la norme est elle ouverte ? La spécification est-elle propriétaire ?

C19 - ITAR risk : les équipements (ou les IP) supports du protocole (switch, hub...) présentent ils des risques d'ITARisation ?

C20 - Mass : optimisation du câblage, nombre de switches. . .

C21 - Maximum latency

Chaque critère a été commenté de manière à présenter les différents éléments pris en compte dans notre évaluation :

- C00 : Les deux termes devraient être étudiés séparément. La ségrégation est la possibilité d'isoler les flux. La priorisation doit s'entendre comme classification de service. Un même moyen peut servir à la ségrégation et/ou à la priorisation. Cinq paramètres ont été utilisés

- pour réaliser cette évaluation à savoir : - le fait si cette fonctionnalité est obtenue en ou hors ligne, - l'isolation (ou ségrégation), - l'indépendance application / communication, le support d'un champ de priorité dans la trame et - la politique d'ordonnement.
- C01 : Trois paramètres sont évalués : - la capacité de changement du mode sol au mode vol, - la capacité du système de communication d'un étage à fonctionner indépendamment du système de communication global du lanceur et - les capacités de synchronisation. Notons que la plupart de ces paramètres dépendent de choix de conception (mode de communication, topologique...)
 - C02 : La solution est d'autant plus robuste à la séparation d'étage que les fonctions applicatives et de communications sont indépendantes.
 - C03 : Il a été assigné une note de 4 si une fonctionnalité native existe (notons que celle-ci pourrait être réalisée de manière applicative, c'est-à-dire par les couches supérieures, mais cette possibilité n'est pas prise en compte dans la notation) et 1 sinon puisque d'autres mécanismes de contrôle peuvent exister (CRC, gardien de bus...)
 - C07 : La solution est d'autant plus mature qu'elle est largement utilisée et/ou a déjà volé sur avion
 - C09 : L'effort de qualification est bien évalué du point de vue de l'effort et non du résultat. Globalement, l'utilisation d'un bus est plus facile qu'une topologie commutée (cf R&T CRAN-CNES)
 - C10 : Il s'agit d'évaluer l'effort d'AIT (requalification, intégration,...) et du traitement de l'obsolescence (qui sera de fait d'autant plus minimisé que la solution est standard et utilise des COTS). La requalification est d'autant plus importante que l'accès au réseau est centralisé (vs distribué où il sera possible de requalifier partiellement et localement sans remettre en cause la totalité de l'architecture).
 - C14 : La redondance peut être native ou liée à des mécanismes additionnels. Cependant, celle-ci peut être le résultat de la conception topologique (fédérative/linéaire redondée/2ème réseaux...).
 - C15 : Il a été assigné une note de 5 aux solutions dont le déterminisme est garanti par son fonctionnement et une note de 4 à celles dont il est possible de calculer de manière déterministe un délai max.
 - C16 : L'observabilité du système est relativement simple dans les technologies s'appuyant sur un bus (identique à ce qui se faisait sur A5). En revanche, dès qu'une technologie s'appuie sur une architecture commutée, il est plus difficile. De ce fait, la notation de ces technologies prend en compte si des solutions sont déjà proposées ou non (que ce soit commerciale ou académiques).
 - C17 : La notation prend seulement en compte la facilité de l'isolation. Dès que la topologie est commutée donc s'appuyant sur des équipements supplémentaires (commutateurs) par

rapport au bus, l'isolation est plus riche.

- C18 : Il a été assigné une note de 5 si la technologie repose totalement sur une norme ouverte, une note de 3 si son utilisation est soumise à licence et une note de 1 si elle propriétaire.
- C20 : Il a été attribuée une note de 5 pour une topologie bus, une note de 3 pour de la fibre optique et 2 pour une topologie commutée.
- C21 : La latence maximale doit s'entendre comme le majorant du délai. Bien que la valeur de celui-ci dépend de l'applicatif et qu'il n'est pas possible de s'engager sur le respect de la contrainte dès maintenant, il est évalué la facilité d'une solution à pouvoir respecter formellement une telle contrainte.

Le tableau A.1 présente la grille d'évaluation de 9 candidats (relativement aux critères présentés précédemment). Celle-ci nous permet de faire le lien avec les exigences (E. *i*) présentées au chapitre 2 (et traitées dans ces travaux de thèse : ✓). Les 9 candidats sont *Fibre Channel* avec FC-AE-1553 (candidat 1 et 2 avec deux topologies différentes), Ethernet standard commuté (3), AFDX (4), TT-Ethernet (5), TTP/C (6), Flexray (7), Arion-100 (8) et Mémoire réfléchive (9).

Le déterminisme (E. 1) peut être vu en tant que tel (comportement de la solution : C15) ou bien en tant que méthode de qualification (C09) des délais maximums (C21) qui garantit alors le comportement temps-réel (E. 2) de manière déterministe. Bien entendu, cela dépend du paradigme qui sera alors choisi.

La reproductibilité (E. 4) peut intervenir dans le critère C10 (coût récurrent et effort d'AIT - *Assembly Integration Test*). En effet, plus une solution est reproductible, plus le coût de test est potentiellement réduit.

Les possibilités d'évolution (E. 10) sont d'autant plus grande que l'indépendance entre les fonctions applicatives et de communication est importante (C08). Cette indépendance permet alors une plus grande modularité (E.9).

La section suivante présente les différents candidats soumis à l'évaluation.

A.3 Présentation des candidats

L'objectif de cette section n'est pas de réaliser une description détaillée de chacun des candidats mais de les présenter brièvement de manière à avoir une vue globale des mécanismes utilisés par chacun d'entre eux.

A.3.1 Candidats 1 et 2 : *Fibre Channel* avec FC-AE-1553

La technologie *Fibre Channel* (FC) est un standard de l'ANSI (*American National Standard Institute*). Elle permet des débits de 130 Mb/s à 4 Gb/s. Son implémentation peut être réalisée avec de la paire torsadée ou de la fibre optique. Pour les deux premiers candidats, la fibre

TABLE A.1: Tableau support à l'évaluation des 9 candidats

Criteria	1		2		3 standard Ethernet switched fabric Copper wire		4	5	6	7	8	9
	Topolgy	Medium										
C00 : Priority and ségrégation												
C01 : Compatibility with a launcher life cycle												
C02 : Robustness to launcher stage jettisoning / reconfiguration					E. 3	✓						
C03 : Integrated control mechanisms												
C04 - Long-term sustainability												
C05 - Development tools												
C06 - Development effort												
C07 - Maturity												
C08 - Flexibility, growth potential, and isolation between layers					E. 9, E. 10							
C09 - Qualification effort					E. 1 ^a , E. 2 ^a	✓						
C10 - Recurring cost					E. 4 ^b , E. 8							
C11 - Robustness to environment (EMC, microcuts...)												
C12 - Energy consumption												
C13 - Non Recurring cost					E. 8							
C14 - Redundancy					E. 5	✓						
C15 - Determinism and jitter					E. 1, E. 2							
C16 - Communication system observability					E. 6	✓						
C17 - Fault isolation					E. 7	✓						
C18 - Normalization & standardization												
C19 - ITAR risk												
C20 - Mass												
C21 - Maximum latency					E. 1, E. 2	✓						

optique a été retenue de manière à profiter de ses avantages (haut débit, réduction du poids et de la taille du système, faible atténuation et isolation électromagnétique, meilleure résistance aux environnements difficiles, ...). Un système à fibre optique supporte plusieurs types de topologies (ASSC, 2006) : point à point, commutée (*switched fabric*) et en anneau (*Arbitrated Loop*).

Il a été retenu pour cette évaluation une topologie hybride en anneau commutée pour le candidat 1 (avec deux anneaux redondants) et commutée pour le candidat 2.

Le protocole de haut niveau (niveau applicatif) est le FC-AE-1553 développé par le groupe FC AE (*Fibre Channel Avionics Environment*). Il correspond globalement à l'implémentation du protocole MIL-STD-1553B (présenté au chapitre 1) sur fibre optique. Il permet notamment quelques extensions comme la bande passante (fournie intrinsèquement par la fibre optique), un espace d'adressage plus important, ou encore une quantité de données par transfert plus important

A.3.2 Candidat 3 : Ethernet standard commuté

Ce candidat représente la technologie étudiée tout au long de cette thèse. Les travaux menés en réponse aux exigences (E. *i*) présentées au chapitre 2 sont résumés dans le tableau A.1.

A.3.3 Candidat 4 : AFDX

Le réseau AFDX est un réseau Ethernet commuté conçu pour l'avionique et développé pour équiper l'Airbus A380. Il est standardisé dans la partie 7 de la norme ARINC 664 développée par Airbus.

Dans ce réseau, les commutateurs interconnectent l'ensemble des abonnés. Ces abonnés échangent des données grâce aux *end systems* qui sont des équipements chargés de l'émission et de la réception de données (*via* les liens full-duplex qui les relient aux commutateurs). Les échanges se font suivant un mode de communication producteurs/consommateurs. La FIGURE A.1 situe l'AFDX dans le modèle OSI.

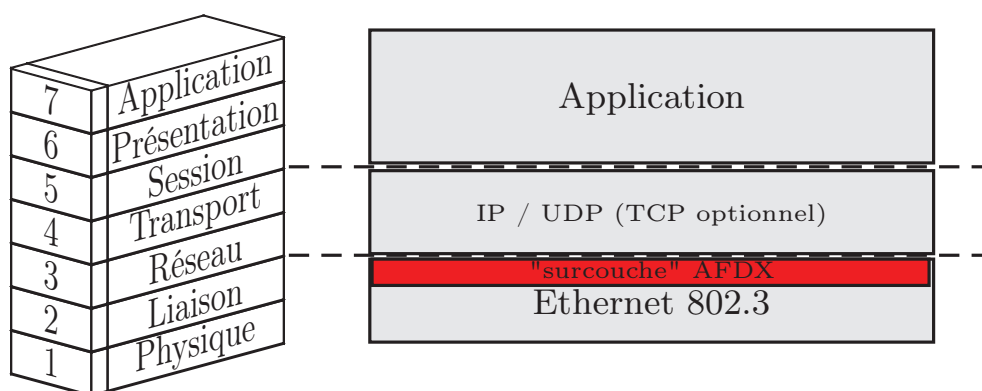


FIGURE A.1: AFDX et modèle OSI

Ces *end systems* échangent leurs informations en multicast à travers des canaux de communication nommés *virtual link* (conformément à la norme ARINC 664). Cette notion (illustrée à la FIGURE A.2) a été développée pour qu'un flux n'influence pas les autres flux sur le réseau ; on parle de ségrégation des flux (Grieu, 2004). Ces canaux de communication définissent une

connexion logique unidirectionnelle d'un *end system* source à un (ou plusieurs) *end system(s)* destination (Charara et Fraboul, 2005). Ils précisent en plus les tailles minimales et maximales des paquets ainsi que le temps entre deux trames consécutives appelées BAG (*Bandwith Allocation Gap*). Grâce aux canaux virtuels, les chemins des flux sont identifiés et statiques (pris en charge par les commutateurs qui sont conformes à la norme IEEE 802.1D mais implémentant un algorithme sous licence) offrant ainsi un comportement déterministe aux réseaux AFDX (Charara *et al.*, 2006). Cependant, avec ces commutateurs et même avec ce mécanisme de liens virtuels, il est possible de perdre des trames (congestion) ne garantissant aucune qualité de service intrinsèque en terme notamment de délai et de gigue. Or dans le milieu avionique, il est nécessaire de prouver la qualité de service requise en vue de la certification du système avionique. La méthode de preuve développée par (Grieu, 2004) s'appuie sur la théorie du calcul réseau et permet de prouver que le délai de traversée du réseau est borné pour tous les flux et qu'il n'y a pas de pertes par congestion (dans les files d'attente des commutateurs).

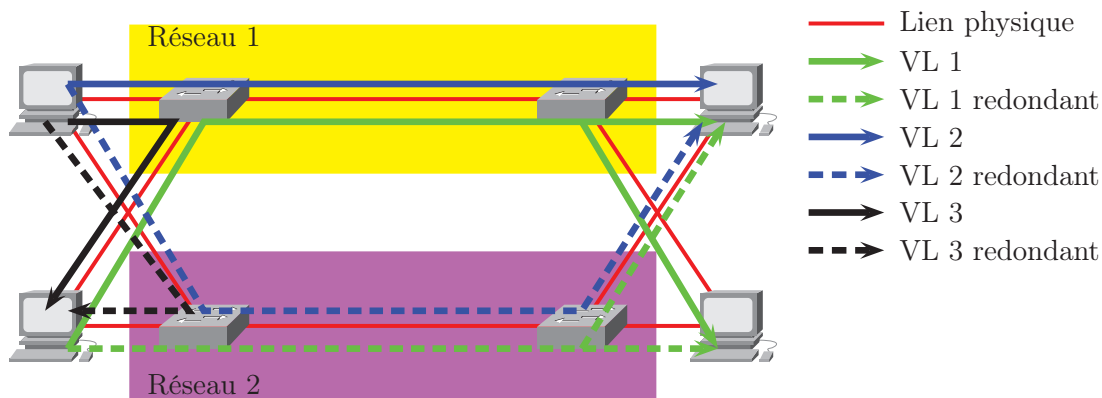


FIGURE A.2: Les canaux de communication virtuels (VL) en AFDX et redondance

Pour améliorer la disponibilité du réseau, l'AFDX utilise une redondance de réseau (FIGURE A.2)). Chaque *end system* possède alors deux ports de communication associés à deux canaux virtuels différents. Les trames sont alors émises simultanément sur les deux réseaux (duplication de la trame). Dans le nœud terminal, une unité de management de la redondance (au niveau de la couche MAC) permet de dupliquer / supprimer les trames qui sont alors identifiées par un numéro de séquence.

Une observation de l'état statique des commutateurs est réalisée périodiquement. Le commutateur vérifie en permanence le bon fonctionnement des liens physiques et son propre fonctionnement. Il rend alors compte de celui-ci à travers les réponses aux requêtes périodiques (Charara, 2007).

A.3.4 Candidat 5 : *Time-Triggered Ethernet (TT-Ethernet)*

La solution *Time-Triggered Ethernet (TT-Ethernet)* qui devrait être normalisée SAE AS6802 courant 2012 a été conçue par l'université technologique de Vienne (Kopetz *et al.*, 2005).

TT-Ethernet est destiné à unifier tout type de trafic (temps réel et non temps réel) sur une même architecture de communication. Pour satisfaire à ce choix, le protocole TT-Ethernet s'appuie sur deux classes de trafic, temps-réel ou non (nommé *time-triggered* ou *event-triggered*). Les flux temps-réels utilisent un format de message spécifique à TT-Ethernet (contrairement aux flux non temps-réel) comme illustré à la FIGURE A.3. Un ordonnancement spécifique est planifié hors-ligne de manière à conférer aux flux temps réel des garanties temporelles. Les flux non temps-réel sont alors préemptés (ce qui est très original, puisque condamne la transmission ainsi stoppée à devenir inaudible) lorsqu'un flux temps-réel doit être retransmis. Ainsi, des commutateurs spécifiques TT-Ethernet (Steinhammer *et al.*, 2006) sont nécessaires pour la gestion de la préemption (avec retransmission des messages préemptés). De plus, une différenciation du mode de commutation (*cut-through* ou *store&forward*) selon la classe de trafic (différenciée par le champ Type d'Ethernet) est utilisée dans le protocole TT-Ethernet. Or, en général, le mode de commutation est défini (par port) de manière indifférenciée par rapport à une classe de trafic (identifiée grâce au champ priorité de l'étiquette 802.1Q dans un commutateur COTS). D'autre part, le protocole TT-Ethernet fournit un délai de transmission constant (connu a priori), c'est-à-dire sans gigue. Celle-ci est éliminée d'une part par la préemption et par l'ordonnancement des messages temps réel en fonction de la priorité des ports d'entrée (dans le cas où plusieurs messages temps-réel arriveraient simultanément sur un commutateur à destination d'un même port de sortie).

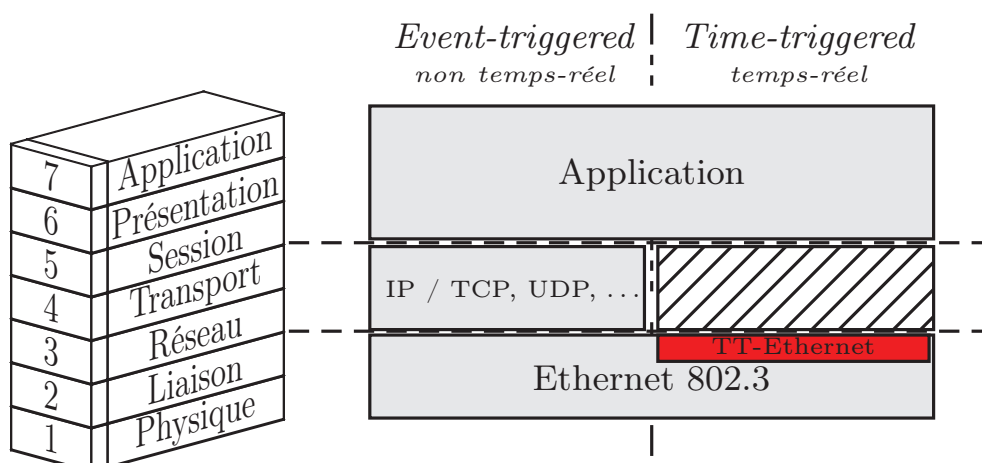


FIGURE A.3: TT-Ethernet et modèle OSI

Notons que dans le cas d'un protocole basé sur un ordonnancement temporel, il est nécessaire que tous des nœuds émetteurs aient la même référence temporelle (*global time*) ce qui implique

que tous les équipements soient synchronisés. Le protocole IEEE 1588 (IEEE Computer Society, 2002a; IEEE, 2008) est alors implémenté dans les cartes de communication TT-Ethernet (Ademaj et Kopetz, 2007).

Pour tolérer les défaillances, le protocole TT-Ethernet utilise une redondance de réseau. Chaque équipement terminal possède une carte de communication spécifique TT-Ethernet de manière à dupliquer / supprimer les trames sur les deux réseaux. Les commutateurs TT-Ethernet implémentent alors une entité spécifique (nommée *guardian*) qui a la connaissance de l'ordonnement des messages temps-réel de façon à désactiver les ports des équipements qui enverraient des messages pouvant remettre en cause cet ordonnancement (Steinhammer et Kopetz, 2006).

A.3.5 Candidats 6 et 7 : *Time triggered protocol* (TTP/C) et Flexray

Le protocole TTP a été conçu par l'université de technologie de Vienne (Kopetz *et al.*, 1999) puis commercialisé par la société TTTech. Flexray est issu d'un *consortium* d'industriels (Consortium *et al.*, 2005) tels que BMW, DaimlerChrysler ou General Motors.

Au niveau physique, les deux candidats s'appuieront sur un bus redondant (soit deux bus). Les débits sont compris entre 500 *kb/s* et 10 *Mb/s* (voire 25 *Mb/s* maximum pour TTP/C).

Au niveau MAC, les deux protocoles s'appuient sur un mécanisme TDMA (*Time Division Multiple Access*). Ce mécanisme permet de diviser un cycle de communication en *slot* (intervalle de temps) dans lequel un seul équipement peut transférer des messages sur le médium. Un gardien de bus permet dans les deux solutions le respect des différentes caractéristiques d'émission. Ce comportement est alors intrinsèquement déterministe.

Ces deux solutions sont relativement semblables et donc associées (Kopetz, 2001; Navet, 2004).

A.3.6 Candidats 8 et 9 : Arion-100 et Mémoire réflexive

Arion-100 ne correspond pas à une technologie, mais à un produit conçu pour le temps-réel par la société Arion.

Le candidat « Mémoire réflexive » n'est également pas une technologie : c'est un concept. Il s'agit de partager un ensemble de données communes qui sont stockées sur chaque équipement. Ces données sont synchronisées en permanence sur l'ensemble des équipements.

D'ailleurs, Arion-100 fournit une implémentation matérielle de ce concept sur ASIC ou FPGA. Le principe est relativement simple : un équipement produit des données et les stocke localement, ces données locales sont ensuite dupliquées *via* le système de communication à toutes les bases locales de chaque équipement de manière à rendre disponibles les données aux consommateurs. Ce fonctionnement permet notamment une indépendance de l'applicatif et de la communication. Ce produit peut être utilisé sur différent système de communication (Ethernet, CAN, MIL-STD-

1553B, ...). Il a été considéré dans cette étude un bus Ethernet.

A.4 Evaluation des candidats

Avant de présenter l'évaluation proprement dite des différents candidats, nous avons émis un commentaire général sur celle-ci.

Parmi les différentes solutions étudiées dans ce trade-off Avionique-X, certaines se singularisent par un comportement synchrone entre l'application et les couches basses. En conséquence, elles profitent dans cette évaluation de caractéristiques inhérentes à l'application et non au standard en lui même. C'est typiquement le cas des réseaux basés sur le protocole 1553. Comme dans cette étude, les autres réseaux n'intègrent pas de protocole applicatif et de mode de communication précis, ils peuvent apparaître ainsi sous performants. On pourrait néanmoins très bien recourir sur Ethernet aux différents modes, du maître esclave à la mémoire réflexive. Plus généralement, il nous paraît opportun de travailler dans une vision davantage systémique (en blocs systèmes) où l'indépendance entre les fonctions applicatives et de communication puisse être renforcée. Ainsi, les possibilités d'évolution de l'applicatif seront plus grandes que ce soit lors d'un lancement ou en termes de missions. De même, les efforts de développement et de qualification pourront être plus aisés. Dans la perspective d'un fonctionnement application/réseau asynchrone et distribué, des critères comme la priorité (C01), le déterminisme et la gigue (C15) devraient être plus fortement pondérés.

Le tableau suivant présente nos notations avec les commentaires associés.

La mémoire réflexive (*Reflexive Memory Systems*) est un concept plutôt qu'une technologie. Ainsi, il pourra être attribué la note associée au système de communication qui sera évalué conjointement à ce concept. C'est pourquoi, elle n'apparaît plus dans le tableau d'évaluation.

Evaluation des candidats									
Criteria	<i>Fiber Channel with 1553 Arbitrated Loop + switched fabric Optical Fiber</i>	<i>Fiber Channel with FC-AE-1553 switched fabric Optical Fiber</i>	<i>standard Ethernet</i>	<i>AFDX</i>	<i>TT-Ethernet</i>	<i>TTP/C</i>	<i>Flexray</i>	<i>Arion 100</i>	
Topology									
Medium									
<i>C00 : Priority and ségrégation</i>	<p>2 : Dans le cadre d'un mode de communication maître/esclave, la priorisation sera réalisé hors-ligne par une gestion du pooling en fonction de la priorité désirée. La ségrégation est native (il n'y aura alors qu'une requête ou une réponse à la fois sur le réseau). Dans le cas du mode de communication maître/esclave, il n'y aucune indépendance entre l'application et la communication. Le champ CS_CTL (propre à Fiber Channel) permet de gérer des priorités.</p>	<p>2 : Dans le cadre d'un mode de communication maître/esclave, la priorisation sera réalisé hors-ligne par une gestion du pooling en fonction de la priorité désirée. La ségrégation est native (il n'y aura alors qu'une requête ou une réponse à la fois sur le réseau). Dans le cas du mode de communication maître/esclave, il n'y aucune indépendance entre l'application et la communication. Le champ CS_CTL (propre à Fiber Channel) permet de gérer des priorités.</p>	<p>5 : L'utilisation du standard 802.1D/Q permet une ségrégation grâce aux VLANs (mais n'apporte pas les mêmes garanties intrinsèques que des liens virtuels, un TDMA ou encore un maître/esclave). 2 octets sont ajoutés à la trame Ethernet. 3 bits sont destinés à coder la priorité du message. Cette priorisation est réalisé en ligne (dans les commutateurs) de façon indépendante de l'applicatif et relativement à une politique d'ordonnement des commutateurs.</p>	<p>4 : La ségrégation est réalisée grâce aux liens virtuels (qui permettent d'isoler les flux tout en leur garantissant de la Bande Passante). Aucune priorité ne peut être assignée dans la trame mais celle-ci peut être assignée aux liens virtuels dans les commutateurs (et relativement à une politique d'ordonnement). En revanche, l'établissement des liens virtuels n'est pas indépendant de l'applicatif.</p>	<p>3 : La ségrégation est réalisée de façon temporelle grâce au TDMA. Pour le trafic TT (déjà prioritaire et différencié par son champ type), un autre niveau de priorité est gérée par rapport au ports d'entrée. Ces mécanismes sont réalisés hors ligne et de façon dépendante de l'applicatif (les messages TT sont choisis relativement à l'applicatif). D'autre part, la préemption permet également la gestion des priorités dans les commutateurs.</p>	<p>2 : La ségrégation est réalisée de façon temporelle grâce au TDMA. Pas de champs priorité dans la trame, en revanche, le priorisation peut être réalisé dans l'établissement du cycle TDMA (qui n'est alors pas indépendant de l'applicatif).</p>	<p>2 : La ségrégation est réalisée de façon temporelle grâce au TDMA. Pas de champs priorité dans la trame, en revanche, le priorisation peut être réalisé dans l'établissement du cycle TDMA (qui n'est alors pas indépendant de l'applicatif).</p>	<p>2 : La ségrégation dépend du choix de mode de communication utilisé sur le bus. La technologie s'appuie sur 16 niveaux de priorités.</p>	

Criteria	Fiber with 1553 Arbitrated Loop + switched fabric Optical Fiber	Channel FC-AE-1553 Loop switched fabric Optical Fiber	Fiber with 1553 switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Copper wire	standard Ethernet Copper wire	AFDX Copper wire	TT-Ethernet Copper wire	TTP/C Copper wire	Flexray Copper wire	Arion 100 Copper wire
Topolgy								bus	bus + hub	hub
Medium										
<i>C01 : Compatibility with a launcher life cycle</i>	<p>3 : Le mode de communication maître/esclave implique une reconfiguration de manière à changer de maître. Un étage peut être fonctionnel du moment où un maître est assigné. Les équipements sont synchronisés par les événements réseau (si datation nécessaire, des mécanismes de synchronisation seront nécessaires).</p>	<p>3 : Le mode de communication maître/esclave implique une reconfiguration de manière à changer de maître. Un étage peut être fonctionnel du moment où un maître est assigné. Les équipements sont synchronisés par les événements réseau (si datation nécessaire, des mécanismes de synchronisation seront nécessaires).</p>	<p>3 : Le mode de communication producteur/consommateur permet de ne pas avoir à reconfigurer les équipements terminaux. En revanche, dans le cas d'un mode centralisé, il sera nécessaire de reconfigurer les chemins dans les commutateurs à l'instant du changement de mode (qui pourra être vérifié par SNMP). Les étages pourront d'autant plus fonctionner indépendamment si la commande est distribuée. Le réseau et les équipements peuvent être synchronisés par le protocole PTP.</p>	<p>3 : Bien que s'appuyant sur un mode de communication producteur/consommateur, les équipements terminaux devront potentiellement être reconfigurés (fermeture/ouverture de liens virtuels). Il sera également nécessaire de reconfigurer les liens virtuels dans les commutateurs à l'instant du changement de mode (qui pourra être vérifié par SNMP). Les étages pourront d'autant plus fonctionner indépendamment si la commande est distribuée. Le réseau et les équipements peuvent être synchronisés par le protocole PTP.</p>	<p>3 : Bien que le TDMA peut ne pas être remis en cause, il sera notamment nécessaire de reconfigurer les commutateurs à l'instant du changement de mode (qui pourra être vérifié par SNMP). Les étages pourront d'autant plus fonctionner indépendamment si la commande est distribuée. Le réseau et les équipements doivent être synchronisés pour les besoins du protocole (TDMA)</p>	<p>4 : Comme le fonctionnement est TDMA et que la topologie est un bus (broadcast), il semble qu'aucune reconfiguration ne soit nécessaire pour le changement de mode. Aux problème de désadaptation près, il semble qu'un étage peut fonctionner indépendamment. Synchro nécessaire par le protocole (TDMA)</p>	<p>4 : Comme le fonctionnement est TDMA et que la topologie est un bus (broadcast), il semble qu'aucune reconfiguration ne soit nécessaire pour le changement de mode. Aux problème de désadaptation près, il semble qu'un étage peut fonctionner indépendamment. Synchro nécessaire par le protocole (TDMA)</p>			

Criteria	Fiber with 1553 Arbitrated Loop + switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Optical Fiber	Fiber with 1553 switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Copper wire	standard Ethernet switched fabric Copper wire	AFDX switched fabric Copper wire	TT-Ethernet switched fabric Copper wire	TTP/C bus Copper wire	Flexray bus + hub Copper wire	Arion 100 hub Copper wire
<i>C02 : Robustness to launcher stage jettisoning / re-configuration</i>	<p>3 : Dans le cadre d'un mode de communication maître/esclave (de niveau applicatif), la reconfiguration devra être réalisé au niveau applicatif de manière à ne plus interroger un équipement perdu (suite à la séparation d'étage). Il est important de faire attention cependant à la manière dont la topologie sera modifiée (dû à la séparation d'étage) et notamment dans la boucle!</p>	<p>3 : Dans le cadre d'un mode de communication maître/esclave (de niveau applicatif), la reconfiguration devra être réalisé au niveau applicatif de manière à ne plus interroger un équipement perdu (suite à la séparation d'étage).</p>	<p>3 : Dans le cadre d'un mode de communication producteur/consommateur, lorsqu'un équipement est perdu, il n'émettra plus de messages. L'application, qui aura eu connaissance de la séparation d'étage, n'aura plus besoin des données des équipements perdus (et cela sans reconfiguration préalable).</p>	<p>5 : Dans le cadre d'un mode de communication producteur/consommateur, lorsqu'un équipement est perdu, il n'émettra plus de messages. L'application, qui aura eu connaissance de la séparation d'étage, n'aura plus besoin des données des équipements perdus (et cela sans reconfiguration préalable).</p>	<p>4 : Dans le cadre d'un mode de communication producteur/consommateur, lorsqu'un équipement est perdu, il n'émettra plus de messages. L'application, qui aura eu connaissance de la séparation d'étage, n'aura plus besoin des données des équipements perdus (et cela sans reconfiguration préalable). D'autre part, les liens virtuels étant initialement configurés pour garantir une certaine bande passante à chacun des flux, devront être reconfigurés de manière à ne pas perdre la bande passante réservée.</p>	<p>3 : Dans le cadre d'un TDMA, lorsqu'un équipement est perdu, il est nécessaire de reconfigurer l'ordonnancement temporel (slot) dans l'ensemble des équipements ayant la connaissance de celui-ci : contrôleur (OBC), commutateurs, équipements terminaux.</p>	<p>3 : Dans le cadre d'un TDMA, lorsqu'un équipement est perdu, il est nécessaire de reconfigurer l'ordonnancement temporel (slot) dans l'ensemble des équipements ayant la connaissance de celui-ci : contrôleur (OBC), équipements terminaux.</p>	<p>3 : Dans le cadre d'un TDMA, lorsqu'un équipement est perdu, il est nécessaire de reconfigurer l'ordonnancement temporel (slot) dans l'ensemble des équipements ayant la connaissance de celui-ci : contrôleur (OBC), équipements terminaux.</p>	<p>3 : Dans le cadre d'un TDMA, lorsqu'un équipement est perdu, il est nécessaire de reconfigurer l'ordonnancement temporel (slot) dans l'ensemble des équipements ayant la connaissance de celui-ci : contrôleur (OBC), équipements terminaux.</p>	<p>? : Dépend du mode de communication choisi.</p>
<i>C03 : Integrated control mechanisms</i>	<p>4 : mot de status du protocole applicatif 1553</p>	<p>4 : mot de status du protocole applicatif 1553</p>	<p>1 : CRC</p>	<p>1 : CRC</p>	<p>1 : CRC</p>	<p>1 : CRC, gardien dans les commutateurs et équipements terminaux</p>	<p>1 : CRC, gardien de bus</p>	<p>1 : CRC, gardien de bus</p>	<p>1</p>	
<i>C04 - Long-term sustainability</i>	<p>3 : Spécification ouverte mais communauté restreinte.</p>	<p>3 : Spécification ouverte mais communauté restreinte.</p>	<p>5 : La note maximale est assignée au standard Ethernet puisqu'étant totalement standard et utilisé dans beaucoup de domaines, la solution est alors pérenne.</p>	<p>4 : Utilise un intergiciel sous licence...</p>	<p>2 : Propriétaire</p>	<p>2 : Propriétaire</p>	<p>3 : Spécification ouverte mais la communauté est principalement le secteur de l'automobile.</p>	<p>1 : Propriétaire</p>		

Criteria	Fiber with 1553 Arbitrated Loop + switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Optical Fiber	Fiber with 1553 switched fabric Copper wire	Channel FC-AE-1553 switched fabric Copper wire	standard Ethernet	AFDX	TT-Ethernet	TTP/C	Flexray	Arion 100
<i>C05 - Development tools</i>	2 : Sans doute des outils ASTRIUM	2 : Sans doute des outils ASTRIUM	5 : Opnet, NS2, Omnet, calcul réseau, RtaW, ...	4 : Calcul réseau, RTaW ...	2 : Il est nécessaire de réaliser une étude d'ingénierie pour la pré-qualification de la solution.	2 : Il est nécessaire de réaliser une étude d'ingénierie pour la pré-qualification de la solution.	3 : Produits TTTech, TimeCriticalNetworks	3 : Produits TT-Tech	4 : Vector, Dg-Test,...	
<i>C06 - Development effort</i>	4	4					1	4	4	1
<i>C07 - Maturity</i>	4 : Fiber Channel utilisé dans du stockage de données commercial. FC-AE-1553 dans programmes militaires (F16, F18, E-2D, JSF...),... 1 : Dans le cadre d'un mode de communication maître/esclave, l'ajout d'un équipement demande une reconfiguration de l'applicatif de manière à pouvoir interroger ce nouvel équipement.	4 : Fiber Channel utilisé dans du stockage de données commercial. FC-AE-1553 dans programmes militaires (F16, F18, E-2D, JSF...),... 1 : Dans le cadre d'un mode de communication maître/esclave, l'ajout d'un équipement demande une reconfiguration de l'applicatif de manière à pouvoir interroger ce nouvel équipement.	5 : Le standard au plus large spectre utilisé dans le monde entier (bureau, Industrie, ...). C'est la technologie la plus mature.	4 : avion (A380), industrie ...			3 : industrie	4 : Industrie, avion (A380, Boeing 787)...	3 : automobile	4 ? : A priori, des projets avec Dassault, Eurocopter, etc
<i>C08 - Flexibility, growth potential, and isolation between layers</i>			5 : L'ajout d'un équipement ne nécessite pas de reconfiguration du réseau. D'autre part, le support d'un protocole de haut niveau est totalement envisageable.	4 : L'ajout d'un équipement nécessite la reconfiguration des liens virtuels de manière à réserver de la bande passante à ce nouveau flux.		3 : L'ajout d'un équipement nécessite la reconfiguration de l'ordonnancement (slot supplémentaire) sur tous les équipements qui en ont la connaissance.	2 : L'ajout d'un équipement nécessite la reconfiguration de l'ordonnancement (slot supplémentaire) sur tous les équipements qui en ont la connaissance.	2 : L'ajout d'un équipement nécessite la reconfiguration de l'ordonnancement (slot supplémentaire) sur tous les équipements qui en ont la connaissance.		3 ? : Dépend du mode de communication choisi.
<i>C09 - Qualification effort</i>	4 : Relativement facile du fait du mode de communication maître/esclave : 1 seul paquet sur le réseau.	4 : Relativement facile du fait du mode de communication maître/esclave : 1 seul paquet sur le réseau.	3 : Solution commutée moins facile mais néanmoins possible grâce au calcul réseau qui permet d'anticiper les évolutions (marge connue).	3 : Solution commutée moins facile mais néanmoins possible grâce au calcul réseau qui permet d'anticiper les évolutions (marge connue).			4 : Relativement facile du fait du TDMA : 1 seul paquet sur le réseau	4 : Relativement facile du fait du TDMA : 1 seul paquet sur le réseau	4 : Relativement facile du fait du TDMA : 1 seul paquet sur le réseau	? : Dépend du mode de communication choisi.
<i>C10 - Recurring cost</i>	2 : 1+1 : accès au médium centralisé	2 : 1+1 : accès au médium centralisé	4 : 2+2 : distribué + COTS standards	3 : 2+1 : distribué + produits sous licence			2 : 1+1 : centralisé	2 : 1+1 : centralisé	2 : 1+1 : centralisé	
<i>C11 - Robustness to environment (EMC, micro-cuts...)</i>										
<i>C12 - Energy consumption</i>										

Criteria	Fiber with 1553 Arbitrated Loop + switched fabric Optical Fiber	Channel FC-AE-1553	Fiber with 1553 switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Copper wire	standard Ethernet switched fabric Copper wire	AFDX switched fabric Copper wire	TT-Ethernet switched fabric Copper wire	TTP/C bus Copper wire	Flexray bus + hub Copper wire	Arion 100 hub Copper wire
<i>C13 - Non Recurring cost</i>	3 : Composants existants : marché avionique. Marché du spatial ?	3 : Composants existants : marché avionique. Marché du spatial ?	3 : Composants existants : marché avionique. Marché du spatial ?	3 : Beaucoup d'équipement à adapter aux lanceurs. Marché du spatial ?	3 : Composants de l'avionique. Marché du spatial ?	3 : Des équipements. Marché du spatial ?	3 : Composants de l'avionique. Marché du spatial ?	3 : Composants de l'automobile. Marché du spatial ?		
<i>C14 - Redundancy</i>	2 : Pas de solutions natives mais ici le choix topologique de l'anneau est de la redondance.	2 : Pas de solutions natives mais le choix de la topologie peut permettre la redondance...	2 : De façon native, les protocoles STP/RSTP peuvent gérer la redondance. Beaucoup de solutions additionnelles existent (notre proposition R&T CRAN-CNES, PRP, MRP, HSR,...)	3 : Redondance de réseaux	4 : possibilité de réseaux parallèles + des mécanismes associés comme le gardien	4 : bus physique redondant + des mécanismes associés comme le gardien de bus	4 : possibilité de bus physique redondant + des mécanismes associés comme le gardien de bus			
<i>C15 - Determinism and jitter</i>	5 : Dans le cadre d'un mode de communication maître/esclave, le comportement est déterministe	5 : Dans le cadre d'un mode de communication maître/esclave, le comportement est déterministe	4 : Délai max et gigue max obtenu par la théorie (déterministe) du network calculus	4 : Délai max et gigue max obtenu par la théorie (déterministe) du network calculus	5 : fonctionnement déterministe : TDMA	5 : fonctionnement déterministe : TDMA	5 : fonctionnement déterministe : TDMA	5 : fonctionnement déterministe : TDMA	5 : fonctionnement déterministe : TDMA	? : Dépend du mode de communication choisi.
<i>C16 - Communication system observability</i>	5 : Dans ce mode de communication maître/esclave, tout peut être observé sur le maître (requêtes et réponse).	3 ou 4 : Selon que ce soit multi réseaux (avec redondants) ou mono maître (tout peut être observé sur le maître).	3 : L'observabilité n'est pas facile (topologie commutée) MAIS la R&T CRAN-CNES a permis de mettre en avant qu'il serait possible de réaliser une fonctionnalité d'observabilité. De plus, le broadcast/multicast est possible (au niveau Ethernet).	2 : L'observabilité n'est pas facile (topologie commutée) MAIS il est potentiellement possible de réutiliser la fonctionnalité présentée dans l'Ethernet Standard. De plus, le broadcast/multicast est possible (au niveau Ethernet et liens virtuels)	2 : L'observabilité n'est pas facile (topologie commutée) MAIS il est potentiellement possible de réutiliser la fonctionnalité présentée dans l'Ethernet Standard. De plus, le broadcast/multicast est possible (au niveau Ethernet).	5 : L'observabilité est aisée car la topologie est un bus.	5 : L'observabilité est aisée car la topologie est un bus.	1 : L'observabilité est aisée car la topologie est un bus. Cependant, il faut tout observer dans chacune des mémoires pour s'assurer de la cohérence spatio-temporelle rendant beaucoup difficile l'observation.		

Criteria	Fiber with 1553 Arbitrated Loop + switched fabric Optical Fiber	Channel FC-AE-1553 Loop switched fabric Optical Fiber	Fiber with 1553 switched fabric Optical Fiber	Channel FC-AE-1553 switched fabric Copper wire	standard Ethernet Copper wire	AFDX Copper wire	TT-Ethernet Copper wire	TTP/C Copper wire	Flexray Copper wire	Arion 100 Copper wire
<i>C17 - Fault isolation</i>	4 : Le mode de communication et la structure en anneau peut faciliter l'isolation des nœuds.		3 : Le mode de communication peut faciliter l'isolation des nœuds, en revanche la topologie commutée la rend plus difficile.		3 : L'isolation des nœuds n'est pas facile (topologie commutée) MAIS la R&T CRAN-CNES a permis de mettre en avant qu'il serait possible de réaliser une fonctionnalité d'isolation.	2 : L'isolation des nœuds n'est pas facile (topologie commutée) MAIS il pourrait être utilisé ce qui a été mis en avant pour l'Ethernet standard.	3 : Bien qu'il n'y ait pas d'isolation, il existe des mécanismes pour se prémunir de ces fautes. D'autre part, il pourrait être utilisé ce qui a été mis en avant pour l'Ethernet standard	4 : Bien qu'il n'y ait pas d'isolation, il existe des mécanismes pour se prémunir de ces fautes (détection d'erreur protocolaire, gardien de bus...). D'autre part, la topologie en bus peut faciliter l'isolation.	4 : Bien qu'il n'y ait pas d'isolation, il existe des mécanismes pour se prémunir de ces fautes (détection d'erreur protocolaire, gardien de bus...). D'autre part, la topologie en bus peut faciliter l'isolation.	
<i>C18 - Normalization & standardization</i>	4 : draft OK mais fouilli		4 : draft OK mais fouilli		5	3	3 : standardisation récente (SAE AS6802). Sous demande de permission à TTTech donc licence???	3 : standardisation récente (SAE AS6003). Sous demande de permission à TTTech donc licence???	5 : standardisé.	1 ?
<i>C19 - ITAR risk</i>										
<i>C20 - Mass</i>	3		3		2	2	2	5	5	5
<i>C21 - Maximum latency</i>	4 : Le mode de communication maître/esclave garantit un délai max puisqu'un seul message est sur le réseau. En revanche, la conception du pooling est importante pour respecter les contraintes de rafraichissement.		4 : Le mode de communication maître/esclave garantit un délai max puisqu'un seul message est sur le réseau. En revanche, la conception du pooling est importante pour respecter les contraintes de rafraichissement.		3 : L'ingénierie de conception facilitera la garantie de la contrainte. Le délai max sera calculé par la théorie du calcul réseau.	3 : L'ingénierie de conception facilitera la garantie de la contrainte. Le délai max sera calculé par la théorie du calcul réseau.	4 : Facilité d'évaluation car TDMA. A prendre en considération : Une variable peut être produite juste après que son slot soit terminé et sa consommation pourra alors être retardé d'un cycle applicatif (dû au retard d'un cycle TDMA).	4 : Facilité d'évaluation car TDMA. A prendre en considération : Une variable peut être produite juste après que son slot soit terminé et sa consommation pourra alors être retardé d'un cycle applicatif (dû au retard d'un cycle TDMA).	4 : Facilité d'évaluation car TDMA. A prendre en considération : Une variable peut être produite juste après que son slot soit terminé et sa consommation pourra alors être retardé d'un cycle applicatif (dû au retard d'un cycle TDMA).	? : Dépend du mode de communication choisi.

A.5 Discussions

Dans cette thèse, trois grandes idées ont été développées :

- distribuer l'avionique sur Ethernet
- assurer la disponibilité du système de communication
- observer pour « jouer au sol » et isoler les défaillances

Nous avons mis en avant le gain que pouvait apporter la distribution de la commande. Ce gain a été montré sur un mode producteur/consommateur où plusieurs trames pouvaient être sur le réseau. Bien que la distribution est possible sur des solutions maître/esclaves (*Fibre Channel* avec FC-AE-1553) et TDMA (TT-Ethernet, TTP/C ou Flexray) où un seul paquet est sur le réseau à un instant donné, ces solutions ne seront pas optimales en termes de bande passante.

Pour assurer la disponibilité du système de communication, nous avons proposé deux méthodes utilisant la « redondance dans le réseau ». La première se base sur la détection de la défaillance par scrutation périodique de l'état des ports des commutateurs. Nous avons vu que cette solution nous permet d'obtenir des temps d'interruption de service très inférieurs à ceux de solutions tels que STP ou RSTP. Elle nous permet également de recouvrer les mêmes performances que le protocole MRP qui n'est utilisable que sur un anneau. La seconde méthode s'appuie sur la duplication de trames sur deux chemins les plus disjoints. Alors que nous utilisons le même réseau physique sur lequel nous configurons logiquement nos chemins, l'AFDX ou encore TT-Ethernet utilisent deux réseaux distincts (« redondance du réseau ») tout en recouvrant les mêmes performances. En revanche, nos équipements terminaux ne possèdent qu'une seule carte de communication qui peut alors être un point unique de défaillance. Nous sommes bien conscient de celui-ci mais nous l'avons toléré puisque ces équipements terminaux sont eux mêmes redondés.

En ce qui concerne l'observabilité, les technologies reposant sur des bus (par exemple TTP/C ou Flexray) bénéficient du mécanisme de diffusion qui permet de n'avoir qu'un seul point d'observation. En revanche, l'utilisation d'une topologie commutée implique de nombreux domaines d'observation. Il s'agit alors de retrouver un seul domaine d'observation. Nous avons donc proposé une méthode pour la fusion de ces domaines d'observation. Comme l'AFDX ou TT-Ethernet reposent sur une architecture commutée (et ne présentant pas, dans leur définition, de mécanisme pour l'observabilité), notre proposition pourrait être exploitée sur ces solutions. Rappelons que la synchronisation des équipements de collecte est nécessaire. TT-Ethernet utilise déjà un mécanisme de synchronisation pour le respect du cycle de communication TDMA, cette caractéristique pourrait alors être exploitée.

A.6 Conclusion

Après application de pondérations, deux concepts ont été retenus pour leur mise en œuvre sur le démonstrateur Avionique-X : un concept synchrone reposant sur la solution TT-Ethernet et un concept asynchrone s'appuyant sur l'AFDX.

Ces deux choix reposent sur la technologie Ethernet commutée. Nous avons montré dans cette thèse que le déterminisme et la disponibilité du système de communication ne sont pas à eux seuls suffisants puisque des problématiques comme l'observabilité sont à considérer. Cette thèse est à la rupture entre un environnement avec un unique domaine d'observation global et un environnement segmenté comptant de nombreux domaines d'observation locaux. Nous avons proposé une méthode pour fusionner ces différents domaines en un seul afin de pouvoir « jouer au sol » et isoler les défaillances. Dans le cas d'un environnement commuté, notre méthode pourra être appliquée. Il en va ainsi des réseaux TT-Ethernet et AFDX.

Le démonstrateur est alors une porte ouverte aux tests de l'applicabilité de nos propositions (et plus particulièrement celle de l'observabilité) aux solutions retenues. Outre le fait que cette évaluation a permis de faire émerger deux solutions, elle met en évidence des critères qu'il reste à traiter. A titre d'exemple, aucun mécanisme de priorités (C00) n'a été utilisé tout au long de ces travaux. En effet, l'évaluation de performances réalisée se basait exclusivement sur les flux fonctionnels. Or, l'objectif est de transporter tous les flux non fonctionnels sur le même système de communication. Ainsi, une évaluation de performances devra être réalisée en intégrant tous ces autres flux. D'autre part, celle-ci pourra prendre en compte des mécanismes de Classification de Services (Cds) de manière à protéger les flux fonctionnels des flux non-fonctionnels.

Liste des publications

Revue internationale avec comité de lecture

Kubler, S., Robert, J., Rondeau, É. et Georges, J.P. Dual path communications over multiple spanning trees for networked control systems. *Engineering Applications of Artificial Intelligence*. *Online* : <http://dx.doi.org/10.1016/j.engappai.2012.05.001>

Robert, J., Georges, J.-P., Rondeau, É., Divoux, T. Minimum Cycle Time Analysis of Ethernet-Based Real-Time Protocols. *International Journal Computers, Communications & Control*. à paraître novembre 2012. *Online* : http://www.uav.ro/files/RTF/ijcccv7n4_draft.pdf

Conférences internationales avec comité de lecture

Robert, J., Georges, J.-P., Divoux, T., Miramont, P., Rmili, B. Ethernet networks for real-time systems : application to launchers. *DATA System In Aerospace (DASIA)*. 2011.

Conférences nationales avec comité de lecture

Robert, J., Georges, J.-P., Rondeau, É., Divoux, T. Analyse de performances de protocoles temps-réel basés sur Ethernet. 6ème Conférence Internationale Francophone d'Automatique (CIFA). 2010.

Groupes de travail nationaux

Robert, J., Georges, J.-P., Divoux, T. Méthodologie de mise en œuvre d'un réseau Ethernet dans les futurs lanceurs spatiaux. Colloque Automatique et Réseau de communications (ARC). 2011.

Robert, J., Georges, J.-P., Divoux, T., Effet de l'écart de synchronisation sur le temps de cycle de solutions basées sur Ethernet, Ecole d'été Temps Réel (ETR), Brest. 2011

Rapports de contrats

Robert, J., Georges, J.-P., Divoux, T., *Livrable 1 : Performances temporelles et disponibilité du réseau Ethernet*. R&T CRAN-CNES n°1, octobre 2009.

Robert, J., Georges, J.-P., Divoux, T., *Livrable 2 : Etude de cas sur les lanceurs*. R&T CRAN-CNES n°1, avril 2010.

Robert, J., Georges, J.-P., Divoux, T., *Livrable 3 : Ingénierie de mise en œuvre d'une architecture Ethernet pour les lanceurs*. R&T CRAN-CNES n°1, novembre 2010.

Robert, J., Georges, J.-P., Divoux, T., *Livrable 4 : Plateforme expérimentale Ethernet commuté CRAN - CNES 2011*. R&T CRAN-CNES n°1, juin 2011.

Robert, J., Georges, J.-P., Divoux, T., *Livrable 1 : Observabilité d'un réseau Ethernet commuté embarqué dans un lanceur*. R&T CRAN-CNES n°2, novembre 2011.

Robert, J., Georges, J.-P., Divoux, T., *Livrable 2 : Observabilité d'un réseau Ethernet commuté embarqué dans un lanceur et isolation de défauts*. R&T CRAN-CNES n°2, juin 2012.

Bibliographie

- Ademaj, A. et H. Kopetz (2007). Time-triggered ethernet and ieee 1588 clock synchronization. In : *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS'07)*. pp. 41–43.
- Alessandria, E., L. Seno et S. Vitturi (2007). Performance analysis of ethernet/ip networks. In : *IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (FET'07)*. pp. 391–398.
- Ali, Y. et S. Narasimhan (1993). Sensor network design for maximizing reliability of linear processes. *AIChE journal*, **39**(5), 820–828.
- Alliance, Industrial Automation Open Networking et A. Lüder (2005). *IAONA Handbook-Industrial Ethernet :(with a List of Producers)*. IAONA.
- Alves, M., E. Tovar et F. Vasques (2000). Ethernet goes real-time : a survey on research and technological developments. Technical Report HURRAY-TR-2K01. Groupe de recherche IPP-HURRAY, Polytechnic Institute of Porto (ISEP-IPP).
- Anderson, E. et M. Arlitt (2006). Full packet capture and offline analysis on 1 and 10 gb networks. *HP laboratories Palo Alto*, pp. 1–15.
- ASSC, Avionic Systems Standardisation Committee-Network Systems Subcommittee (2006). Guide to digital interface standards for military avionic applications. Technical report. ASSC Avionic Systems Standardisation Committee.
- Bauer, H. (2011). Analyse pire cas de flux hétérogènes dans un réseau embarqué avion. PhD thesis. Université de Toulouse.
- Brahimi, B. (2007). Proposition d'une approche intégrée basée sur les réseaux de Petri de haut niveau pour simuler et évaluer les systèmes contrôlés en réseau. PhD thesis. Université Henri Poincaré Nancy 1.
- Brooks, P. (2001). Ethernet/ip-industrial protocol. In : *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '01)*. pp. 505–514.
- Cervin, A., D. Henriksson, B. Lincoln, J. Eker et K.-E. Arzen (2003). How does control timing affect performance?. *IEEE Control Systems magazine*, **23**(3), 16–30.

- Chang, C.-S. (2000). *Performance Guarantees in Communication Networks*. Telecommunication Networks and Computer Systems. Springer Verlag.
- Charara, H (2007). Évaluation des performances temps réel de réseaux embarqués avioniques. PhD thesis. Institut National Polytechnique de Toulouse.
- Charara, Hussein et Christian Fraboul (2005). Modelling and simulation of an avionics full duplex switched ethernet. In : *Advanced Industrial Conference on Telecommunications (AICT'05)*.
- Charara, Hussein, Jean-Luc Scharbarg, Jérôme Ermont et Christian Fraboul (2006). Methods for bounding end-to-end delays on an afdx network. In : *Euromicro Conference on Real-Time Systems (ECRTS'06)*. pp. 193–202.
- Cisco Systems (2005). Using the cisco span port for san analysis.
- Consortium, FlexRay et al. (2005). Flexray communications system-protocol specification. *Version*, **2**(1), 198–207.
- Correll, K., N. Barendt et M. Branicky (2005). Design considerations for software only implementations of the ieee 1588 precision time protocol. In : *Conference on IEEE*. Vol. 1588.
- Cruz, R.L. (1991). A calculus for network delay. i. network elements in isolation. *Information Theory, IEEE Transactions on*, **37**(1), 114–131.
- Decker, E., A. Rijsinghani, K. McCloghrie et P. Langille (1991). Definitions of managed objects for bridges, rfc1286.
- Decotignie, J.-D. (2001). A perspective on ethernet-tcp/ip as a fieldbus. In : *IFAC International Conference on Fieldbus Systems and their Applications (FET'01)*. pp. 138–143.
- Decotignie, J.-D. (2005). Ethernet-based real-time and industrial communications. In : *Proceedings of the IEEE*. Vol. 93. pp. 1102–1117.
- Department of Defense (1978). Military standard aircraft internal time division command/response multiplex data bus. MIL-STD 1553B.
- Diouri, I. (2010). Propositions de méthodes pour adapter le réseau aux contraintes d'applications temps-réel. PhD thesis. Université Henri Poincaré Nancy 1.
- Divoux, T., E. Rondeau, J.P. Georges et al. (2008). Ethernet : des îles au bureau, puis dans l'atelier. *La Revue de l'Electricité et de l'Electronique 1*, pp. 95–101.
- Dréo, J., A. Petrowski, P. Siarry et E. Taillard (2006). *Metaheuristics for hard optimization : methods and case studies*. Springer.
- Enacheanu, B., B. Raison, R. Caire, O. Devaux, W. Bienia et N. Hadjsaid (2008). Radial network reconfiguration using genetic algorithm based on the matroid theory. *Transactions on Power Systems*, **23**(1), 186–195.

- Enacheanu, Bogdan, Bertrand Raison, Daniela Ivanova, Raphaël Caire, A. Aubry et Nouredine Hadjsaid (2006). Flexible Electric Infrastructures for Advanced Distribution Automation. In : *International Conference on Critical Infrastructures, CRIS'06*.
- Felser, M. (2008). Media redundancy for profinet io. In : *IEEE International Workshop on Factory Communication Systems (WFCS'08)*. pp. 325–330.
- Felser, M. et T. Sauter (2004). Standardization of industrial ethernet - the next battlefield?. In : *IEEE International Workshop on Factory Communication Systems (WFCS'04)*. pp. 413–420.
- Ferrandiz (2012). Maîtrise des latences de communication dans les réseaux bord SpaceWire. PhD thesis. Université de Toulouse.
- Ferrari, P., A. Flammini, D. Marioli et A. Taroni (2008). A distributed instrument for performance analysis of real-time ethernet networks. *IEEE Transactions on Industrial Informatics*, **4**, 16–25.
- Georges, J.-P., E. Rondeau et T. Divoux (2002). Evaluation of switched ethernet in an industrial context by using the network calculus. In : *IEEE International Workshop on Factory Communication Systems (WFCS'02)*. pp. 19–26.
- Georges, J.-P., T. Divoux et E. Rondeau (2003). Comparison of switched ethernet architectures models. In : *IEEE Conference on Emerging Technologies and Factory Automation (ET-FA'03)*. Vol. 1. Lisbonne, Portugal. pp. 375–382.
- Georges, J.P. (2005). Systèmes contrôlés en réseau : Evaluation de performances d'architectures Ethernet commutées. PhD thesis. Université Henri Poincaré Nancy 1.
- Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.. Boston, MA, USA.
- Grieu (2004). Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques. PhD thesis. Institut National Polytechnique de Toulouse.
- Harima, T. (2009). Introduction of iec activity on digital communication subsystem (overview of iec/sc65c). In : *ICCAS-SICE*. pp. 1427–1431.
- Hong, Ying-Yi. et Saw-Yu. Ho (2005). Determination of network configuration considering multiobjective in distribution systems using genetic algorithms. *IEEE Transactions on Power Systems*, **20**(2), 1062 – 1069.
- Huynh, M., S. Goose et P. Mohapatra (2010). Resilience technologies in ethernet. *Computer Networks*, **54**(1), 57–78.
- IEC (2003a). Iec 61158 : Digital data communications for measurement and control-fieldbus for use in industrial control systems.
- IEC (2003b). Iec 61784 : Digital data communications for measurement and control.

- IEC (2006a). Digital data communications for measurement and control – fieldbus for use in industrial control systems : Part 3 : Data link service definition. IEC Standard 61158, Part 3.
- IEC (2006b). Digital data communications for measurement and control – fieldbus for use in industrial control systems : Part 4 : Data link protocol specification. IEC Standard 61158, Part 4.
- IEC (2008). Industrial communication networks : high availability automation networks. IEC62439.
- IEEE (2008). Ieee standard for a precision clock synchronization protocol for networked measurement and control systems. ANSI/IEEE standard 1588-2008.
- IEEE Computer Society (1998). Ieee standards for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - common specifications - part 3 : Media access control (mac) bridges. ANSI/IEEE Std 802.1D, Edition 1998.
- IEEE Computer Society (2002a). Ieee standard for a precision clock synchronization protocol for networked measurement and control systems. ANSI/IEEE standard 1588-2002.
- IEEE Computer Society (2002b). Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 3 : Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications. IEEE standard 802.3, Edition 2002.
- IEEE Computer Society (2003). Ieee standards for local and metropolitan area networks - virtual bridged local area networks. IEEE standard 802.1Q, Edition 2003.
- IEEE Computer Society (2008). Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 3 : Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications. IEEE standard 802.3, Edition 2008.
- Imtiaz, J., J. Jasperneite et K. Weber (2010). Redundant structures for a generic real-time ethernet system. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'10)*. IEEE. pp. 1–4.
- Jasperneite, J. et E. Elsayed (2004). Investigations on a distributed time-triggered ethernet realtime protocol used by profinet. In : *International Workshop on Real-Time Networks (RTN'04)*. pp. 69–72.
- Jasperneite, J., P. Neumann, M. Theis et K. Watson (2002). Deterministic real-time communication with switched ethernet. In : *IEEE International Workshop on Factory Communication Systems (WFCS'02)*. pp. 11–18.

- Jasperneite, Jürgen, Markus Schumacher et Karl Weber (2007). Limits of increasing the performance of industrial ethernet protocols. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07)*. pp. 17 – 27.
- Jayavelu, Giridhar, Srinivasan Ramasubramanian et Ossama Younis (2009). Maintaining colored trees for disjoint multipath routing under node failures. *IEEE/ACM Transactions on Networking*, **17**(1), 346–359.
- Kirrmann, H. et D. Dzung (2006). Selecting a standard redundancy method for highly available industrial networks. In : *IEEE International Workshop on Factory Communication Systems (WFCS'06)*. pp. 386–390.
- Kirrmann, H., K. Weber, O. Kleineberg et H. Weibel (2009). Hsr : Zero recovery time and low-cost redundancy for industrial ethernet (high availability seamless redundancy, iec 62439-3). In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'09)*. pp. 1–4.
- Kirrmann, H., M. Hansson et P. Muri (2007). Iec 62439 prp : Bumpless recovery for highly available, hard real-time industrial networks. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07)*. pp. 1396–1399.
- Kopetz, H. (2001). A comparison of ttp/c and flexray. *Research Report*.
- Kopetz, H., A. Ademaj, P. Grillinger et K. Steinhammer (2005). The time-triggered ethernet (tte) design. In : *IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*. pp. 22–33.
- Kopetz, H., T. Galla, H. Angelow, E. Fuchs, T. Führer et R. Hexel (1999). Specification of the ttp/c protocol.
- Krommenacker, N. (2002). Heuristiques de conception de topologies réseaux : application aux réseaux locaux industriels. PhD thesis. Université Henri Poincaré, Nancy 1.
- Kubler, Sylvain, Jérémy Robert, Jean-Philippe Georges et Éric Rondeau (2012). Dual path communications over multiple spanning trees for networked control systems. *Engineering Applications of Artificial Intelligence*.
- Le Boudec, J.-Y. (1996). Network calculus made easy. Technical Report EPFL-DI 96/218. Ecole Polytechnique Fédérale de Lausanne (EPFL).
- Le Boudec, J.-Y. et P. Thiran (2001). *Network calculus, a theory of deterministic queueing systems for the Internet*. Lecture Notes in Computer Science. Springer Verlag.
- Limal, S., S. Potier, B. Denis et J.J. Lesage (2007). Formal verification of redundant media extension of ethernet powerlink. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07)*. pp. 1045–1052.
- Lin, W.-M., F.-S. Cheng et M.-T. Tsay (2000). Distribution feeder reconfiguration with refined genetic algorithm. *Transactions on Power Systems Proceedings - Generation, Transmission and Distribution*, **147**(6), 349–354.

- Liu, L. et G. Frey (2007). Simulation approach for evaluating response times in networked automation systems. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'07)*. pp. 1061–1068.
- Marsal, G. (2006). Evaluation of time performances of Ethernet-based automation systems by simulation of high-level Petri nets. PhD thesis. ENS Cachan.
- Modbus-IDA (2006). Modbus application protocol specification v1.1b.
- Modlovansky, A. (2002). Utilization of modern switching technology in ethernet/ip networks. In : *International Workshop on Real-Time LANs in the Internet Age (RTLIA'02)*. pp. 35–37.
- Monchaux, D., P. Gast et J. Sangare (2012). Avionic-x : A demonstrator for the next generation launcher avionic. In : *Embedded Real-Time Software and Systems (ERTS2'12)*.
- Nara, K., A. Shiose, M. Kitagawa et T. Ishihara (1992). Implementation of genetic algorithm for distribution systems loss minimum re-configuration. *IEEE Transactions on Power Systems*, **7**(3), 1044–1051.
- Navet, N. (2004). Présentation du réseau flexray - comparaison avec ttp/c. In : *GDR "Spécification Temporelle et Stochastique et Systèmes Temps Réel" (STRQDS'04)*.
- Papadogiannakis, A., M. Polychronakis et E.P. Markatos (2010). RRDtrace : Long-term Raw Network Traffic Recording using Fixed-size Storage. In : *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS'10)*. pp. 101–110.
- Pereira, N., E. Tovar et L.M. Pinho (2004a). Indepth : Timeliness assessment of ethernet/ip-based systems. In : *International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'04)*. pp. 192–201.
- Pereira, N., E. Tovar et L.M. Pinho (2004b). Timeliness in cots factory-floor distributed systems : what role for simulation?. In : *IEEE International Workshop on Factory Communication Systems (WFCS'04)*. pp. 13–21.
- Prytz, Gunnar (2008). A performance analysis of ethercat and profinet irt. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'08)*. pp. 408–415.
- Radha, B., RTF King et HCS Rughooputh (2004). A modified genetic algorithm for optimal electrical distribution network reconfiguration. In : *IEEE Congress on Evolutionary Computation (CEC'04)*. Vol. 2. pp. 1472–1479.
- Ramos, E. R., A. G. Exposito, J. R. Santos et F. L. Iborra (2005). Path-based distribution network modeling : application to reconfiguration for loss reduction. *IEEE Transactions on Power Systems*, **20**(3), 556–564.
- Ridouard, F., J.L. Scharbarg et C. Fraboul (2007). Stochastic network calculus for end-to-end delays distribution evaluation on an avionics switched ethernet. In : *IEEE International Conference on Industrial Informatics*. pp. 559–564.

- Robert, J., JP Georges, E. Rondeau et T. Divoux (2012). Minimum cycle time analysis of ethernet-based real-time protocols. *International Journal of Computers, Communications & Control*, p. 242.
- Robert, J., J.P. Georges, T. Divoux, P. Miramont, B. Rmili et al. (2011). Ethernet networks for real-time systems : application to launchers. In : *DATA Systems in Aerospace (DASIA '11)*.
- Robert, Jérémy, Jean-Philippe Georges, Eric Rondeau et Thierry Divoux (2010). Analyse de performances de protocoles temps-réel basés sur Ethernet. In : *Sixième Conférence Internationale Francophone d'Automatique, (CIFA '10)*.
- Ruel, S. (2009). Évaluation des bornes des performances temporelles des Architectures d'Automatisation en Réseau par preuves itératives de propriétés logiques. PhD thesis. ENS Cachan.
- Rüping, S., E. Vonnahme et J. Jasperneite (1999). Analysis of switched ethernet networks with different topologies used in automation systems. In : *Proceedings of Fieldbus Conference (FeT'99)*.
- Schiffer, V. (2006). The common industrial protocol (CIP™) and the family of CIP networks. Technical Report PUB00123R0. Open DeviceNet Vendor Association, Inc. (ODVA).
- Schmitt, J.B., F.A. Zdarsky et M. Fidler (2008). Delay Bounds under Arbitrary Multiplexing : When Network Calculus Leaves You in the Lurch... In : *IEEE Conference on Computer Communications (INFOCOM'08)*. pp. 1669–1677.
- Schmitt, Jens B., Frank A. Zdarsky et Ivan Martinovic (2006). Performance bounds in feed-forward networks under blind multiplexing. Technical Report 349/06. University of Kaiserslautern, Germany.
- Seno, Lucia et Claudio Zunino (2008). A simulation approach to a real-time ethernet protocol : Ethercat. In : *IEEE Conference on Emerging Technologies and Factory Automation (ETFA'08)*. pp. 440–443.
- Song, Y.Q. (2001). Time constrained communication over switched ethernet. In : *IFAC International Conference on Fieldbus Systems and Their Applications (FET'01)*. pp. 152–169.
- Steinhammer, K., P. Grillinger, A. Ademaj et H. Kopetz (2006). A time-triggered ethernet (tte) switch. In : *Conference on Design, automation and test in Europe*. pp. 794–799.
- Steinhammer, Klaus et Hermann Kopetz (2006). Time-triggered ethernet. In : *Junior Scientist Conference (JSC'06)*. pp. 73–74.
- The Modbus Organization (n.d.). Modbus.
- Zaitsev, D.A. et T.R. Shmeleva (2006). Switched ethernet response time evaluation via colored petri net model. *Proceedings of International Middle Eastern Multiconference on Simulation and Modelling*, pp. 68–77.

- Zhang, J. et A. Moore (2007). Traffic trace artifacts due to monitoring via port mirroring. In : *Workshop on End-to-End Monitoring Techniques and Services (E2EMON'07)*. pp. 1–8.
- Zhu, Y. et K. Tomsovic (2002). Adaptive power flow method for distribution systems with dispersed generation. *IEEE Transactions on Power Delivery*, **17**(3), 822–827.

Résumé

Les lanceurs spatiaux actuels, et en particulier Ariane 5, utilisent le réseau déterministe MIL-STD-1553B. Compte tenu des nouveaux objectifs de réduction des coûts et de la masse du système global ainsi que du support de trafic de télémesure multimédia, cette technologie n'est plus forcément optimale. Parmi les différents candidats de remplacement, cette thèse met en évidence les capacités du standard Ethernet à répondre à ces nouveaux objectifs tout en s'appuyant sur une utilisation de composants matériels et d'outils de développement existants à moindre coût (COTS). La première contribution porte sur l'évaluation de performances temporelles des architectures commutées. Les travaux mettent ainsi en évidence les gains et limites liés au choix d'une part du mode de communication (de maître/esclaves à producteurs/consommateur) et d'autre part à une future distribution de l'avionique. Pour cela, cette étude s'appuie sur une évaluation déterministe des délais de bout en bout par calcul réseau, simulations et expérimentations. Ces résultats ont été validés pour deux architectures commutées pour un scénario représentatif des vols actuels. La seconde contribution concerne l'amélioration de la disponibilité du système de communication. Il est proposé une stratégie de reconfiguration « temps réel » des chemins par supervision active du réseau. Dans le cadre d'exigences critiques, il est également proposé l'utilisation d'arbres couvrants multiples permettant d'anticiper la défaillance d'éléments d'interconnexion. Enfin, le choix d'architectures commutées et segmentées ne permet plus d'observer en tout point le réseau comme sur un bus. Pour cela, cette thèse met en avant les conditions et performances dans lesquelles un protocole de synchronisation d'horloges pourra contribuer à générer à partir de plusieurs points de captures une trace unique des échanges sur le réseau. Ce travail permettra d'identifier les tests pour une future validation du standard Ethernet pour les lanceurs spatiaux.

Mots clés

Systèmes spatiaux, systèmes temps-réel, évaluation de performances, Network Calculus, Ethernet commuté

Abstract

Current space launchers, and particularly Ariane 5, use the deterministic network MIL-STD-1553B. According to the new objectives of cost and system mass reduction and of multimedia traffic support, this technology is not optimal anymore. Among the potential candidates, this thesis highlights the fact that such objectives can be achieved through the use of Ethernet standard based on components-on-the-shelf. The first contribution focuses on time performance evaluation of switched architectures. The gain and limits related to the communication mode (from master/slaves to producers/consumers) and future avionic distribution are studied. This study relies on a deterministic evaluation of the end-to-end delay by using network calculus, simulations and experiments. These results are validated with two switched architectures by using a scenario considered as representative of current flights. The second contribution is the network availability improvement. A real-time path reconfiguration strategy is proposed through active network supervision. Based on critical requirements, it is also suggested to use multiple spanning-trees for anticipating network element failures. The last contribution deals with the issue that in switched and segmented architectures it is not possible to collect all the traffic as in a bus. In order to do so, this thesis introduces the configurations under which a clock synchronization protocol could contribute to generate a single network trace from many collecting points. This work will enable to identify the tests for a future Ethernet standard validation in the framework of space launchers.

Keywords

Spatial systems, real-time systems, performance evaluation, Network Calculus, switched Ethernet