

THÈSE DE DOCTORAT

UNIVERSITÉ PARIS XII

présentée par

Steven MARTIN

le 6 juillet 2004

pour obtenir le titre de

DOCTEUR EN SCIENCES

Spécialité Informatique

<p>MAÎTRISE DE LA DIMENSION TEMPORELLE DE LA QUALITÉ DE SERVICE DANS LES RÉSEAUX</p>
--

Directrice de thèse : Pascale MINET

Co-encadrement : Laurent GEORGE

JURY

Président : ...

Rapporteurs :	Francis COTTET	ENSMA, Poitiers
	Françoise SIMONOT-LION	INRIA Lorraine

Examineurs :	Yacine AMIRAT	Université Paris 12
	Laurent GEORGE	Université Paris 12
	Pascal LORENZ	Université de Haute Alsace
	Pascale MINET	INRIA Rocquencourt
	Samir TOHMÉ	Université de Versailles St-Quentin

Résumé

Internet est devenu incontournable dans notre vie et notre travail en proposant de nouvelles applications telles que la téléphonie sur IP, la vidéo à la demande ou encore les jeux interactifs distribués. Mais ces nouvelles applications nécessitent des garanties de la part du réseau en termes de délai, de débit et de pertes pour fonctionner correctement. Dans cette thèse, nous nous intéressons à des applications ayant des contraintes fortes en termes de temps de réponse et de gigue de bout-en-bout. Pour chaque flux généré par ces applications, l'utilisateur spécifie deux paramètres de qualité de service (QoS), à savoir :

- un degré d'importance, représentant la criticité du flux du point de vue utilisateur ;
- un paramètre temporel, utilisé pour départager les flux de même degré d'importance.

Le temps mis par un paquet pour traverser le réseau dépend essentiellement du temps passé dans les files d'attente des différents nœuds visités. La politique d'ordonnancement est donc un élément essentiel dans une architecture de QoS. Nous proposons un modèle d'ordonnancement permettant de fournir des garanties déterministes sur le temps de réponse et la gigue de bout-en-bout de tout flux parcourant le réseau. Ce modèle est basé sur un ordonnancement à base de priorités fixes (FP) combiné à un ordonnancement à base de priorités dynamiques (DP).

Tout paquet entrant dans le réseau se voit assigner la priorité fixe du flux auquel il appartient, ainsi qu'une priorité dynamique. Les paquets sont alors ordonnancés dans les différents nœuds selon les priorités attribuées sur le premier nœud visité. Plus précisément, les paquets sont traités dans l'ordre de leur priorité fixe, puis dans l'ordre de leur priorité dynamique pour ceux ayant la même priorité fixe. Nous notons un tel ordonnancement FP/DP^* .

Ainsi, nous établissons des bornes mathématiquement calculables sur les temps de réponse et les giges de bout-en-bout des flux considérés lorsque les paquets sont ordonnancés FP/DP^* dans chacun des nœuds. Pour la détermination de ces bornes, nous appliquons une approche dite "par trajectoire", qui ne considère que des scénarios possibles. L'établissement de ces bornes est effectué successivement pour trois configurations de complexité croissante :

- cas monoprocesseur ;
- cas d'une simple ligne de diffusion : tous les flux suivent la même séquence de nœuds ;
- cas général distribué : les flux suivent des lignes quelconques.

Nous appliquons ensuite nos résultats à deux ordonnancements FP/DP^* particuliers : $FP/FIFO^*$ et FP/EDF^* . Nous pouvons déduire des résultats obtenus ceux concernant les ordonnancements FP (les flux ont tous une priorité fixe différente), $FIFO^*$ et EDF^* (les flux ont tous la même priorité fixe).

En contexte monoprocesseur, nous montrons alors que nous améliorons les résultats existants pour les ordonnancements FIFO et EDF et retrouvons les résultats existants pour l'ordonnancement FP. De plus, la dominance de FP/EDF sur FP/FIFO est établie sous certaines hypothèses. Par ailleurs, nous apportons de nouveaux résultats dans le cas distribué.

Les résultats obtenus avec l'approche par trajectoire améliorent significativement les résultats obtenus par l'approche holistique. Par ailleurs, nous évaluons la précision de nos résultats par des exemples numériques en les comparant avec les valeurs exactes, fournies par un outil de validation que nous avons développé.

Ensuite, nous analysons l'impact d'une remise en forme d'un flux sur son temps de réponse de bout-en-bout. Nous considérons deux techniques de remise en forme : l'annulation de gigue et le seau à jetons. Nous discutons alors l'intérêt d'une remise en forme et des mérites respectifs des deux techniques de remise en forme étudiées.

Enfin, nous présentons un exemple d'application de notre solution dans le cadre d'une architecture de QoS combinant DiffServ et MPLS. Nous dérivons, à partir des résultats établis dans cette thèse, un contrôle d'admission permettant de garantir de manière déterministe les temps de réponse et les giges de bout-en-bout des flux de la classe la plus prioritaire (la classe EF). Nous montrons ensuite comment généraliser à d'autres architectures de QoS.

Table des matières

Résumé	i
I Présentation du sujet et de son contexte	1
Chapitre 1 Introduction	3
1.1 Introduction	4
1.2 Présentation du sujet	4
1.2.1 Définition de la qualité de service	5
1.2.2 Besoins de garanties déterministes	5
1.3 Intérêt du sujet	6
1.4 Démarche suivie	8
1.5 Contributions et aspects novateurs	9
1.6 Organisation du document	10
Chapitre 2 Problématique	13
2.1 Introduction	14
2.1.1 Inadéquation d'un traitement uniforme	14
2.1.2 Dimension temporelle de la qualité de service	15
2.1.3 Importance de l'ordonnancement	15
2.2 Modèles considérés	15
2.2.1 Modèle de réseau	15
2.2.2 Modèle de flux	16
2.2.3 Modèle d'ordonnancement	17
2.3 Cas étudiés	21
2.3.1 Cas monoprocesseur	21
2.3.2 Cas d'une ligne de diffusion	22
2.3.3 Cas général	22
2.4 Outil de validation	22
2.5 Problématique traitée	23
2.6 Conclusion	24
Chapitre 3 Etat de l'art	25
3.1 Introduction	26
3.2 Ordonnancement monoprocesseur	26
3.2.1 Incidence de la non-préemption	27
3.2.2 Ordonnancement non-préemptif	27

3.2.3	Ordonnancement <i>Fixed Priority</i>	28
3.2.4	Ordonnancement <i>First In First Out</i>	30
3.2.5	Ordonnancement <i>Earliest Deadline First</i>	31
3.2.6	Ordonnancements associant priorités fixes et priorités dynamiques	32
3.3	Ordonnancement distribué	33
3.3.1	Approche holistique	33
3.3.2	Network calculus	34
3.4	Architectures de qualité de service	37
3.4.1	Les services intégrés (IntServ)	37
3.4.2	Les services différenciés (DiffServ)	40
3.5	Contrôle d'admission	43
3.6	Conclusion	44

II Nouveaux résultats pour les ordonnancements de type FP/DP* 45

Chapitre 4 *Notations et définitions* 47

4.1	Introduction	48
4.2	En contexte monoprocesseur	49
4.3	En environnement distribué	50
4.3.1	Ligne de diffusion	50
4.3.2	Cas général	51
4.4	Conclusion	51

Chapitre 5 *Contexte monoprocesseur* 53

5.1	Introduction	54
5.2	Démarche suivie	54
5.3	Analyse pire cas	55
5.3.1	Instant de démarrage au plus tard	55
5.3.2	Temps de réponse pire cas	59
5.4	Avantage de combiner priorités fixes et priorités dynamiques	62
5.5	Cas particuliers	64
5.5.1	Les flux ont tous une priorité fixe différente	64
5.5.2	Les flux ont tous la même priorité fixe	65
5.6	Ordonnancement FP/FIFO	66
5.6.1	Temps de réponse pire cas	67
5.6.2	Ordonnancement FIFO	68
5.6.3	Exemples	69
5.7	Ordonnancement FP/EDF	71
5.7.1	Temps de réponse pire cas	71
5.7.2	Ordonnancement EDF	72
5.7.3	Exemples	73
5.7.4	Région d'ordonnançabilité	74
5.8	Dominance de FP/EDF sur FP/FIFO	75
5.8.1	Propriétés	75
5.8.2	Exemples	78
5.9	Conclusion	79

Chapitre 6	<i>Ligne de diffusion</i>	81
6.1	Introduction	82
6.2	Démarche suivie	82
6.3	Analyse pire cas	83
6.3.1	Approche par trajectoire	83
6.3.2	Evaluation des différents délais	84
6.3.3	Instant de démarrage au plus tard sur le dernier nœud visité	88
6.3.4	Instants à tester	90
6.3.5	Temps de réponse pire cas de bout-en-bout	93
6.4	Cas particuliers	94
6.4.1	La ligne de diffusion est réduite à un seul nœud	94
6.4.2	Les flux ont tous une priorité fixe différente	94
6.4.3	Les flux ont tous la même priorité fixe	95
6.5	Ordonnancement FP/FIFO*	96
6.5.1	Temps de réponse pire cas de bout-en-bout	97
6.5.2	Ordonnancement FIFO*	98
6.5.3	Exemples	99
6.6	Ordonnancement FP/EDF*	101
6.6.1	Temps de réponse pire cas	102
6.6.2	Ordonnancement EDF*	102
6.6.3	Exemples	104
6.7	Conclusion	105
Chapitre 7	<i>Cas distribué</i>	107
7.1	Introduction	108
7.2	Demarche suivie	108
7.3	Analyse pire cas	109
7.3.1	Approche par trajectoire	109
7.3.2	Evaluation des différents délais	113
7.3.3	Instant de démarrage au plus tard sur le dernier nœud visité	116
7.3.4	Instants à tester	118
7.3.5	Temps de réponse pire cas de bout-en-bout	119
7.4	Généralisation	120
7.5	Cas particuliers	120
7.5.1	Les flux suivent tous une même ligne de diffusion	120
7.5.2	Les flux ont tous une priorité fixe différente	120
7.5.3	Les flux partagent tous la même priorité fixe	121
7.6	Ordonnancement FP/FIFO*	122
7.6.1	Temps de réponse pire cas de bout-en-bout	123
7.6.2	Ordonnancement FIFO*	124
7.7	Ordonnancement FP/EDF*	125
7.7.1	Temps de réponse pire cas	125
7.7.2	Ordonnancement EDF*	126
7.8	Conclusion	128

III Extensions et applications	129
Chapitre 8 <i>Impact de la remise en forme du trafic</i>	131
8.1 Introduction	132
8.2 Notations	133
8.3 Temps de réponse et gigue de bout-en-bout	133
8.4 Sans remise en forme	134
8.5 Avec remise en forme par flux	134
8.5.1 Remise en forme par annulation de gigue	135
8.5.2 Remise en forme par seau à jetons	136
8.6 Exemple	141
8.7 Etude comparative avec l'ordonnancement FIFO	142
8.7.1 Sans remise en forme	142
8.7.2 Avec remise en forme par annulation de gigue	143
8.7.3 Avec remise en forme par seau à jetons	144
8.8 Discussion	145
8.8.1 Annulation de gigue ou seau à jetons ?	145
8.8.2 Avec ou sans remise en forme ?	145
8.9 Conclusion	146
Chapitre 9 <i>Exemples d'applications</i>	147
9.1 Introduction	148
9.2 Architecture DiffServ	148
9.2.1 Modèle DiffServ	148
9.2.2 MPLS	148
9.2.3 Solution proposée	150
9.2.4 Contrôle d'admission	151
9.2.5 Exemple	153
9.2.6 Discussion	154
9.3 Autres architectures	155
9.3.1 Architecture IntServ	155
9.3.2 Architecture hybride	155
9.4 Conclusion	155
IV Conclusion	157
Chapitre 10 <i>Conclusion et perspectives</i>	159
10.1 Conclusion	160
10.2 Perspectives	161
Index	165
Glossaire	166
Liste des publications	168
Bibliographie	173

Table des figures

1.1	Exemple d'application à fortes contraintes temporelles	7
2.1	Structure d'un nœud du réseau considéré	16
2.2	L'ordre d'ordonnancement dépend du nœud	19
2.3	Exemple d'implémentation d'un ordonnancement de type FP/DP	21
2.4	Paramétrage du réseau est des flux	22
2.5	Résultats fournis par l'outil de validation	23
3.1	Effet non-préemptif	27
3.2	Composants nécessaires dans le modèle IntServ	38
3.3	Fonctionnement du protocole RSVP	39
3.4	Composants nécessaires dans le modèle DiffServ	41
3.5	Principe du contrôle d'admission	43
5.1	Période active de niveau $PG_i(t)$ considéré	55
5.2	Comparaison des régions d'ordonnançabilité obtenues avec FP et FP/FIFO	70
5.3	Comparaison des régions d'ordonnançabilité obtenues avec FP et FP/EDF	74
5.4	Comparaison des régions d'ordonnançabilité obtenues avec FP/FIFO et FP/EDF	79
6.1	Détermination du temps de réponse de bout-en-bout avec l'approche par trajectoire	83
6.2	Temps de réponse pire cas de bout-en-bout	99
6.3	Temps de réponse pire cas de bout-en-bout du flux τ_4	100
6.4	Temps de réponse pire cas de bout-en-bout	104
6.5	Temps de réponse pire cas de bout-en-bout du flux τ_4	105
7.1	Les flux peuvent se croiser au plus une fois	109
7.2	Périodes actives de niveau $PG_i(t)$ considérées	110
7.3	Périodes actives de niveau $PG_i(t)$ considérées dans le pire cas	111
8.1	Structure d'un nœud du réseau considéré	132
8.2	Distorsion d'un flux τ_i sur les deux premiers nœuds visités	134
8.3	Technique de l'annulation de gigue	136
8.4	<i>Shaper</i> composé d'un seau à jetons	136
8.5	Technique du seau à jetons	139
8.6	Temps de réponse avec et sans remise en forme	141
9.1	Structure d'un label	149
9.2	Combinaison de DiffServ et MPLS	150

9.3	Trafic EF dans le domaine DiffServ	153
-----	--	-----

Liste des tableaux

5.1	Améliorations obtenues sur les temps de réponse pire cas avec FP/FIFO	69
5.2	Caractéristiques des flux considérés	70
5.3	Améliorations obtenues sur les temps de réponse pire cas avec FP/EDF	73
5.4	Caractéristiques des flux considérés	74
5.5	Comparaison entre les ordonnancements FP/EDF et FP/FIFO	78
5.6	Caractéristiques des flux considérés	78
6.1	Caractéristiques des flux considérés	99
6.2	Temps de traitement maximum de tout flux τ_i sur chaque nœud	100
6.3	Caractéristiques des flux considérés	104
6.4	Temps de traitement maximum de tout flux τ_i sur chaque nœud	105
9.1	Conditions à vérifier pour l'acceptation d'un nouveau flux τ_i dans la classe EF	152

Première partie

Présentation du sujet et de son contexte

CHAPITRE 1

Introduction

1.1	Introduction	4
1.2	Présentation du sujet	4
1.2.1	Définition de la qualité de service	5
1.2.2	Besoins de garanties déterministes	5
1.3	Intérêt du sujet	6
1.4	Démarche suivie	8
1.5	Contributions et aspects novateurs	9
1.6	Organisation du document	10

1.1 INTRODUCTION

Dans cette thèse, nous nous intéressons à la détermination de garanties déterministes de qualité de service dans un réseau. Ainsi, dans la section 1.2, nous présentons le sujet et précisons les notions de qualité de service (QoS) et de garantie déterministe, en décrivant les paramètres temporels de QoS étudiés. Nous montrons dans la section 1.3 l'intérêt de ce sujet, en présentant notamment plusieurs types d'applications nécessitant des garanties de QoS pour fonctionner correctement. Dans la section 1.4, nous présentons la démarche suivie dans cette thèse. Puis, dans la section 1.5, nous résumons les contributions et aspects novateurs du travail réalisé¹. Enfin, l'organisation du document est détaillée dans la section 1.6.

1.2 PRÉSENTATION DU SUJET

Internet est devenu incontournable dans notre vie et notre travail en proposant de nouvelles applications telles que la téléphonie sur IP, la vidéo à la demande ou encore les jeux interactifs distribués. Contrairement aux applications pour lesquelles Internet fut conçu (transfert de fichiers, accès à distance, courrier électronique), ces nouvelles applications nécessitent des garanties de la part du réseau en termes de délai, de débit et de pertes pour fonctionner correctement [1, 2]. Or l'architecture de base d'Internet est restée longtemps inchangée, la transmission des paquets de données s'effectuant toujours selon le principe “au mieux” (*best-effort*). Ce principe représente le type de service le plus simple qu'un réseau puisse offrir, ce dernier garantissant simplement qu'il fera de son mieux pour qu'un paquet soit délivré à son destinataire. La gestion du trafic en *best-effort* correspond en fait à l'absence de mécanisme de qualité de service. Le réseau ne peut donc offrir aucune garantie aux nouvelles applications.

Les partisans d'un réseau basé sur le principe du *best-effort* considèrent que l'évolution rapide des débits satisfait les besoins sans qu'il soit nécessaire de mettre en œuvre une gestion complexe du trafic. Si cette pratique est envisageable sur un réseau local, l'exercice devient rapidement onéreux dans de grands réseaux, malgré une baisse constante du coût des liaisons. De plus, l'abondance de bande passante est rapidement comblée par de nouvelles applications encore plus “gourmandes” et par des usagers toujours plus nombreux. Par ailleurs, le surplus de bande passante ne privilégie pas les applications sensibles (au délai d'acheminement par exemple), qui reçoivent toujours un service de type *best-effort*. Il est donc nécessaire de privilégier le traitement d'applications sensibles, sans pour autant surdimensionner le réseau.

Afin de répondre à ces nouveaux besoins à forte valeur ajoutée, les prestataires de service doivent intégrer les deux aspects suivants :

- l'*assurance de performance*. L'architecture de base d'Internet a peu de capacités pour gérer les ressources dans le réseau et ne peut donc pas fournir de garanties, notamment en termes de délais de bout-en-bout. La téléphonie sur IP, par exemple, est sensible au délai. Lorsque celui-ci devient trop important, l'application est inutilisable ;
- la *différenciation de service*. Internet traite tous les paquets de la même manière, il ne peut donc offrir qu'un seul et unique niveau de service. Les besoins des utilisateurs sont pourtant différents. Par exemple, une entreprise se servant d'Internet pour des transactions bancaires voudra certainement payer davantage pour obtenir un traitement préférentiel de son trafic.

La notion de qualité de service, définie dans la section suivante, devient donc primordiale [3].

¹Les contributions seront présentées en détail dans la conclusion de cette thèse.

1.2.1 Définition de la qualité de service

Intuitivement, l'idée de la qualité de service, ou QoS (*Quality of Service*), est d'offrir mieux que le service *best-effort* afin que les applications ayant des exigences spécifiques fonctionnent correctement. Cette idée n'est pas nouvelle ; elle est en réalité aussi ancienne que les réseaux eux-mêmes. Les réseaux téléphoniques, les réseaux synchrones fédérateurs, puis les réseaux de type ATM, ont par exemple toujours intégrés des mécanismes de QoS. En effet, ces réseaux ont été historiquement construits pour transporter de la voix. Dans le monde de l'Internet, la première proposition de qualité de service date de 1979 [4]. Plus récemment, l'IETF (*Internet Engineering Task Force*), organisme responsable de la coordination des propositions ayant valeur de standards pour les protocoles de l'Internet, fut le siège de nouveaux protocoles et architectures de qualité de service² : IntServ [5] (avec le protocole de réservation RSVP [6]), puis DiffServ [7].

Comme l'observe [8], les termes "qualité" et "service" sont individuellement vagues et permettent bon nombre d'interprétations. Nous considérerons par la suite que la qualité de service d'un réseau est sa capacité à fournir un service adapté aux besoins spécifiques des applications. D'un point de vue technique, la QoS proposée à une application est caractérisée par un ensemble de paramètres, à savoir :

- le temps de réponse de bout-en-bout, correspondant au temps que requiert un paquet pour traverser le réseau d'un point d'entrée à un point de sortie ;
- la gigue de bout-en-bout, représentant la variation maximale des temps de réponse de bout-en-bout des paquets dans le réseau ;
- la bande passante disponible, devant être suffisante pour absorber le trafic généré par les utilisateurs ;
- le taux de pertes subi par les paquets, représenté par le rapport entre le nombre d'octets émis et le nombre d'octets reçus ;
- la disponibilité du service rendu, qui peut s'exprimer comme la probabilité que le service soit disponible à un instant t quelconque.

La bande passante est le paramètre généralement adopté comme critère de QoS. En effet, les solutions proposées sont principalement axées sur ce paramètre, en argumentant que la garantie d'un grand débit est synonyme de faibles temps de réponse. Malheureusement, cette affirmation est coûteuse car elle introduit un couplage délai / bande passante et implique un surdimensionnement du réseau lorsque les applications sensibles au délai ne nécessitent pas un débit important [9]. Il est donc nécessaire de s'intéresser plus finement à la dimension temporelle de la QoS, c'est pourquoi nous nous focaliserons dans cette thèse aux deux paramètres correspondants : le temps de réponse de bout-en-bout et la gigue de bout-en-bout.

1.2.2 Besoins de garanties déterministes

Nous nous intéressons aux applications temps-réel, c'est-à-dire aux applications ayant des exigences fortes en termes de délai d'acheminement des données et de variation de ce délai d'acheminement. Typiquement, une application temps-réel a une échéance sur le temps de réponse de bout-en-bout (et éventuellement une autre sur la gigue de bout-en-bout). Si cette échéance n'est pas respectée, l'application se dégrade ou devient inutilisable. A titre d'exemple, une application de téléphonie IP [10], consistant à transmettre la voix dans des paquets IP, ne nécessite pas une bande passante importante (au maximum 8 kb/s avec un bon algorithme de compression) mais a les contraintes suivantes :

²Ces architectures sont décrites dans la section 3.4.

- le délai de transmission total des paquets doit être suffisamment faible pour ne pas interférer avec les conversations. [11] recommande un délai inférieur à 150 ms ;
- la variation du délai de transmission total ne doit pas excéder quelques dizaines de millisecondes pour ne pas nuire à la qualité audio en réception.

Les applications à fortes contraintes temporelles, telle que la voix, nécessitent donc des garanties de QoS pour fonctionner correctement. Ces garanties peuvent être probabilistes ou déterministes. En effet, une application peut se contenter d'une qualité que l'on puisse suffisamment prédire pour qu'elle soit convenablement satisfaite. Garantir, par exemple, pour le contrôle à distance d'un télescope, que le temps de réponse des commandes n'excédera pas quelques centaines de millisecondes dans 99% des cas semble satisfaisant. D'autres applications, en revanche, nécessitent des garanties déterministes, c'est-à-dire le respect sans faute des contraintes imposées par le service demandé. C'est le cas des applications en bases de données gérant des données répliquées et pour lesquelles le temps de mise-à-jour d'une donnée sur toutes les copies doit être strictement borné.

Les résultats présentés dans ce document permettent d'établir des garanties déterministes sur les temps de réponse de bout-en-bout et les gigue de bout-en-bout des différentes applications coexistant dans un réseau.

1.3 INTÉRÊT DU SUJET

L'avènement de technologies de plus en plus performantes rend possible le développement d'applications de plus en plus exigeantes en terme de qualité de service [12]. Par ailleurs, la numérisation de la voix et de l'image permet, au-delà de la rationalisation des coûts, le développement de nouvelles applications réellement multimédia (c'est-à-dire associant voix, données et images). L'ensemble des investissements consentis par les constructeurs et les opérateurs dans le domaine de la voix sur IP attestent de cette réalité. Mais l'utilisation d'un même réseau pour plusieurs types d'applications, dont les contraintes et les profils peuvent être très différents, nécessite de définir et de mettre en place des règles pour que ces applications se partagent efficacement les ressources réseaux et la bande passante disponible, tout en répondant au mieux à leurs exigences de QoS.

Pour résoudre les problèmes de QoS sur Internet, l'une des techniques consiste à augmenter fortement la bande passante. Ainsi, les réseaux d'entreprise, qu'il s'agisse des réseaux locaux ou des réseaux étendus, ont été souvent surdimensionnés pour répondre à ce besoin de QoS, au détriment d'une rentabilité maximale. Aujourd'hui, les constructeurs d'équipements de télécommunications et les opérateurs investissent beaucoup dans le domaine de la QoS, afin d'exploiter au mieux les ressources et donc permettre une réduction des coûts généraux de l'infrastructure. Ce constat d'applique à Internet mais également à la conception de systèmes embarqués (type véhicule) disposant de plusieurs nœuds interconnectés par plusieurs réseaux.

Les réseaux d'entreprise actuels sont encore relativement simples et leurs trafics majoritairement issus d'applications peu sensibles à la QoS (messagerie, réplification de bases de données, transfert de fichiers ou accès à des serveurs Web intranet). Mais face à la généralisation d'applications de type ERP³, *e-business* ou vidéoconférence, les entreprises se tournent de plus en plus vers des opérateurs proposant une offre d'interconnexion comportant au minimum trois niveaux de services : un service best-effort pour les applications non sensibles, un service assuré pour les applications critiques de l'entreprise et un service temps-réel pour les applications de voix ou vidéo sur IP.

³Un progiciel de gestion intégré, ou ERP (*Enterprise Resource Planning*), est un ensemble complet de programmes standard utilisables directement pour la gestion d'une entreprise après une adaptation sous forme de paramétrage.

La gestion préconisée par les constructeurs d'équipements réseau consiste à disposer de mécanismes inhérents aux équipements réseau permettant de favoriser certains flux prioritaires, grâce à leur analyse préalable (demande implicite) ou à leur demande explicite de QoS (préalablement au transfert) [9]. Les flux moins prioritaires sont alors mis en file d'attente dans les nœuds du réseau et écoulés dès que possible, à moins qu'une saturation trop importante du réseau n'oblige à les détruire partiellement ou en totalité.

Cette gestion implique que les nœuds du réseau soient en mesure d'assurer les trois fonctionnalités suivantes [13] :

- la classification ;
- la mise en file d'attente conformément à la classification ;
- l'ordonnancement en fonction des classes.

Afin de simplifier les équipements, certains modèles de QoS proposent d'effectuer la classification complexe des paquets à l'entrée du réseau. Ainsi, les paquets sont marqués sur leur nœud d'entrée puis traités dans les éléments centraux du réseau en fonction de leur marquage. Cependant, il est important de rappeler qu'à ce jour, la plupart des applications ne savent pas signaler dynamiquement leurs besoins de QoS pour recevoir le niveau de service adapté.

Dans cette thèse, nous proposons une solution (voir section 2.2.3, page 17) proposant des garanties déterministes en termes de délai de traversée du réseau et variation de ce délai, basée sur une politique d'ordonnancement spécifique. Ces garanties sont essentielles pour la mise en place d'applications à fortes contraintes temporelles. Par ailleurs, notre solution ne nécessite de la part des applications que deux paramètres de QoS, à savoir un degré d'importance et un paramètre temporel. Enfin, cette solution est conforme aux orientations actuelles car les traitements complexes sont réalisés sur les nœuds en bordure du réseau.

Pour illustrer les besoins de QoS en termes de temps de réponse et de gigue de bout-en-bout, outre les applications temps-réel classiques, nous pouvons citer le projet *Concert Virtuel Réparti* [14] mené par le CNAM (*Conservatoire National des Arts et Métiers*) et illustré à la figure 1.1⁴.

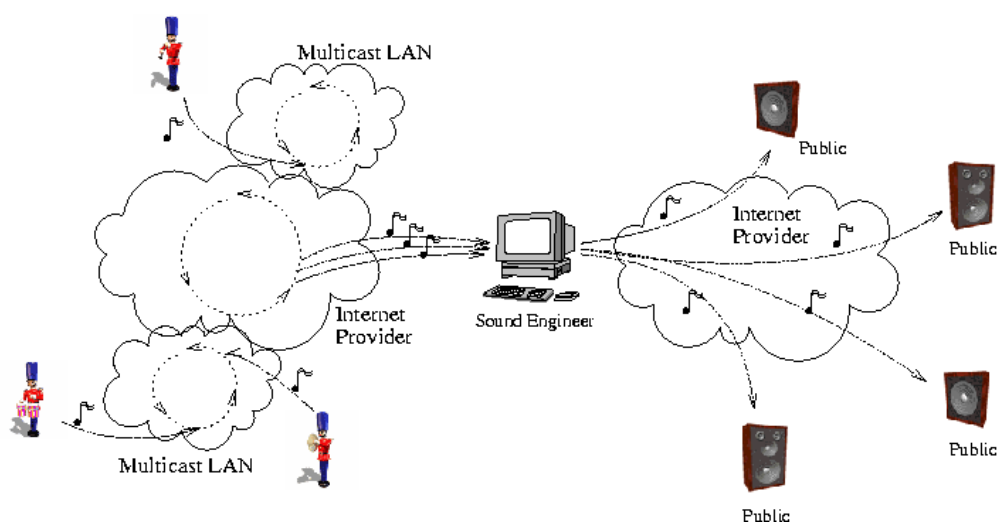


FIG. 1.1: Exemple d'application à fortes contraintes temporelles

⁴Cette image est extraite de [14].

Des musiciens, physiquement répartis, doivent jouer “virtuellement ensemble” en temps-réel. Une telle application nécessite des bornes strictes sur le temps de réponse de bout-en-bout afin de s’assurer d’une bonne *cohérence auditive*. Mais une telle garantie déterministe ne suffit pas. En effet, pour assurer une simultanéité entre les différents musiciens, la gigue de bout-en-bout doit également être bornée.

Un autre exemple d’application est la vidéo à la demande, ou VoD (*Video on Demand*), permettant à un utilisateur de choisir un contenu multimédia à visualiser et de contrôler sa transmission par des commandes classiques du type : *play*, *pause*, etc. Des études [15] affirment qu’en 2006, 40% de la population bénéficiant d’une connexion Internet haut débit sollicitera un tel service. En Amérique du Nord, ce marché représentera environ 7,6 millions d’utilisateurs et générera 820 millions de dollars de revenus. Commercialement, les systèmes multimédia doivent être capables de fournir les services demandés avec la qualité de service requise. Pour l’utilisateur, cette qualité de service s’exprimera principalement par une visualisation fluide et une bonne réactivité du serveur aux commandes transmises [16]. La maîtrise de la dimension temporelle de la QoS est donc, là encore, primordiale.

Cette thèse s’inscrit donc dans la problématique actuelle de QoS dans les réseaux et, plus précisément, de garanties déterministes pour la mise en place d’applications temps-réel. Ces garanties sont essentielles pour le bon fonctionnement de ces applications. Elles sont également nécessaires aux fournisseurs de service et aux constructeurs informatiques qui commercialisent ainsi une qualité de service à forte valeur ajoutée. En effet, ceux-ci doivent fournir des mesures de QoS orientées de plus en plus vers la disponibilité et le respect de contraintes temporelles de bout-en-bout [17].

1.4 DÉMARCHE SUIVIE

Dans cette thèse, nous souhaitons offrir des garanties déterministes à des applications ayant des contraintes temps-réel. Nous nous intéressons plus particulièrement aux flux générés par ces applications. Nous proposons une solution basée sur un ordonnancement combinant priorités fixes et priorités dynamiques. Les priorités fixes constituent le critère principal d’ordonnancement, les priorités dynamiques ne servant qu’à départager les paquets de même priorité fixe. Nous notons un tel ordonnancement FP/DP. Deux exemples d’ordonnancement FP/DP sont FP/FIFO et FP/EDF. Le premier est le plus utilisé en raison de sa simplicité. Le second tire avantage de l’optimalité d’EDF en monoprocesseur.

Dans ce contexte, nous considérons successivement trois configurations de complexité croissante :

- 1) cas monoprocesseur ;
- 2) cas d’une simple ligne de diffusion, c’est-à-dire tous les flux suivent la même séquence de nœuds ;
- 3) cas général distribué.

Pour chaque configuration, nous réalisons une analyse pire cas afin de déterminer les temps de réponse pire cas des flux. Cette analyse pire cas est réalisée avec une approche dite “par trajectoire” (voir chapitre 6). Cette approche se base sur les scénarios pouvant se produire, c’est pourquoi elle fournit des résultats précis.

Afin de valider les résultats que nous obtenons, nous procédons comme suit :

- **Validation par la configuration de moindre complexité**

Dans le cas d’une simple ligne de diffusion, nous montrons que nous retrouvons les résultats établis dans le cas monoprocesseur lorsque cette ligne se réduit à un seul nœud. De même, dans le cas général distribué, nous montrons que nous retrouvons les résultats établis dans le cas d’une simple ligne de diffusion lorsque tous les flux suivent la même séquence de nœuds.

- **Validation par l'état de l'art**

Pour chaque configuration, nous considérons deux ordonnancements particuliers, à savoir FP/FIFO et FP/EDF. Nous considérons ensuite deux cas : (i) les flux ont tous une priorité fixe différente et (ii) les flux ont tous la même priorité fixe. Nous établissons ainsi des résultats pour les ordonnancements FP, FIFO et EDF. Ces résultats sont alors comparés à ceux présentés dans l'état de l'art.

- **Validation par comparaison avec l'approche exhaustive**

Pour chaque configuration, nous présentons des exemples numériques permettant d'apprécier la précision des résultats obtenus. Pour ce faire, nous avons développé un outil de validation permettant de générer la solution exhaustive à un problème d'ordonnancement donné. Nous pouvons alors comparer les résultats fournis par cet outil avec ceux obtenus grâce à l'approche par trajectoire.

1.5 CONTRIBUTIONS ET ASPECTS NOVATEURS

Les contributions et les aspects novateurs de cette thèse concernent principalement les points suivants :

- **Prise en compte de contraintes temporelles**

Nous proposons une solution permettant de prendre en compte, pour chaque flux généré, deux paramètres simples de QoS spécifiés par l'utilisateur : le degré d'importance du flux et un paramètre temporel. Le degré d'importance représente la criticité du flux du point de vue utilisateur et le paramètre temporel permet de départager les flux de même degré d'importance.

- **Ordonnements composites**

Notre solution se base sur un ordonnancement combinant priorités fixes et priorités dynamiques. Ainsi, la priorité fixe reflète le degré d'importance du flux et la priorité dynamique est calculée à partir de son paramètre temporel. La priorité fixe est le critère principal d'ordonnement, la priorité dynamique ne servant qu'à départager les flux de même priorité fixe. Nous considérons que les priorités fixes et dynamiques sont attribuées sur le nœud d'entrée du flux et notons un tel ordonnancement FP/DP* (voir section 2.2.3).

- **Garanties déterministes**

Avec la solution proposée, nous offrons des garanties déterministes sur les temps de réponse et les giges de bout-en-bout des flux. Le déterminisme des bornes proposées est indispensable pour certaines applications temps-réel (par exemple le contrôle de processus).

- **Approche par trajectoire**

Les résultats sur les temps de réponse pire cas sont obtenus à l'aide d'une approche par trajectoire, ne considérant que des scénarios pouvant se produire, à la différence de l'approche holistique (voir section 3.3). Dans cette thèse, nous montrons comment utiliser l'approche par trajectoire dans une analyse pire cas en environnement distribué.

- **Nouveaux résultats ou améliorations de résultats existants**

Nous obtenons les temps de réponse pire cas pour FP/DP*⁵. Nous appliquons ces résultats à deux ordonnancements FP/DP* particuliers : FP/FIFO* et FP/EDF*. Ainsi, nous montrons qu'en contexte monoprocesseur : (i) nous améliorons les résultats pour les ordonnancements FIFO et EDF, (ii) nous retrouvons les résultats existants pour l'ordonnement FP et (iii) nous établissons de nouveaux résultats pour les ordonnancements FP/FIFO, FP/EDF. Par ailleurs, nous établissons de nouveaux résultats en environnement distribué pour les ordonnancements FP/FIFO*, FP/EDF*, FP, FIFO* et EDF*.

⁵Dans le cas monoprocesseur, un ordonnancement FP/DP* équivaut à un ordonnancement FP/DP.

- **Comparaison de différentes approches (trajectoire, exhaustive, holistique)**

Nous montrons que les résultats établis en environnement distribué avec l'approche par trajectoire améliorent significativement les résultats obtenus avec l'approche holistique. Par ailleurs, pour chaque configuration étudiée, nous évaluons la précision de nos résultats par des exemples numériques en les comparant avec les valeurs exactes, fournies par un outil de validation que nous avons développé.

- **Dominance de FP/EDF sur FP/FIFO en monoprocesseur**

En monoprocesseur, deux ordonnancements FP/DP sont particulièrement intéressants :

- FP/FIFO, qui correspond à l'implémentation la plus utilisée de FP en raison de sa simplicité ;
- FP/EDF, qui tire bénéfice de l'optimalité d'EDF [18].

Nous démontrons qu'en monoprocesseur, FP/EDF domine FP/FIFO lorsque les flux partageant la même priorité fixe ont même temps de traitement maximum.

- **Contrôle d'admission**

A partir des résultats établis dans cette thèse, nous montrons comment dériver un contrôle d'admission permettant de garantir, de manière déterministe, des bornes sur les temps de réponse et les giges de bout-en-bout. Ce contrôle d'admission est établi dans une architecture de QoS combinant DiffServ et MPLS. Il est généralisable à d'autres architectures de QoS (voir chapitre 9).

- **Garanties quantitatives pour la classe EF du modèle DiffServ**

La classe de service la plus prioritaire du modèle DiffServ, proposée pour des applications ayant des contraintes temps-réel, souffre d'une définition purement qualitative. Dans cette thèse, nous montrons comment apporter des garanties quantitatives.

1.6 ORGANISATION DU DOCUMENT

Cette thèse est organisée en quatre parties, présentées ci-dessous.

- **Partie I Présentation du sujet et de son contexte**

Dans la première partie, nous introduisons tout d'abord le sujet étudié et montrons son intérêt (chapitre 1). Nous précisons ensuite la problématique considérée (chapitre 2), à savoir : fournir aux différents flux parcourant un réseau des garanties déterministes en termes de temps de réponse et de gigue de bout-en-bout.

Nous proposons alors un modèle d'ordonnancement, basé sur un ordonnancement FP (*Fixed Priority*) combiné à un ordonnancement DP (*Dynamic Priority*). Ainsi, un flux se voit assigner une priorité fixe en entrée du réseau et à chaque paquet de ce flux est attribuée une priorité dynamique. Dans les nœuds, les paquets sont ordonnancés en fonction de leurs priorités généralisées, c'est-à-dire suivant leurs priorités fixes puis leurs priorités dynamiques s'ils ont même priorité fixe. Nous notons un tel ordonnancement FP/DP* si la priorité dynamique d'un paquet est attribuée sur le premier nœud visité, FP/DP si un paquet se voit attribuer une priorité dynamique sur chacun des nœuds.

Afin de comparer les résultats obtenus à ceux existants, nous rappelons également dans la première partie (chapitre 3) un ensemble de résultats de l'état de l'art connus en ordonnancement monoprocesseur et distribué.

- **Partie II : Nouveaux résultats pour les ordonnancements de type FP/DP***

Dans la deuxième partie, nous considérons un ordonnancement de type FP/DP* non-préemptif, c'est-à-dire lorsque la priorité dynamique d'un paquet est attribuée sur le premier nœud visité. Après avoir précisé nos notations (chapitre 4), nous déterminons le temps de réponse pire cas d'un flux quelconque parmi un ensemble de flux. Trois cas sont étudiés : le cas monoprocesseur où les flux sont traités dans un seul et même nœud (chapitre 5), le cas d'une ligne de diffusion, où les flux suivent tous une même séquence de nœuds (chapitre 6) et le cas général, où les flux suivent des chemins différents dans le réseau (chapitre 7).

Pour la détermination du temps de réponse pire cas d'un flux, nous adoptons l'approche par trajectoire, consistant à examiner l'ordonnancement produit sur l'ensemble des nœuds visités par le flux. Avec cette approche, les seuls scénarios considérés sont des scénarios pouvant se produire, ce qui conduit à des résultats moins pessimistes que ceux basés sur l'approche holistique. Nous comparons les résultats obtenus avec l'approche par trajectoire aux résultats établis avec l'approche holistique puis ceux fournis par une approche exhaustive.

Nous appliquons enfin nos résultats à deux ordonnancements DP particuliers : FIFO (*First In First Out*) et EDF (*Earliest Deadline First*). Nous montrons ainsi que nous améliorons les résultats connus pour les ordonnancements FIFO et EDF en contexte monoprocesseur et apportons de nouveaux résultats en environnement distribué. De plus, les résultats théoriques obtenus sont confirmés par un outil de validation réalisé dans le cadre de cette thèse.

- **Partie III : Extensions et applications**

Dans la troisième partie, nous nous intéressons au cas où les flux sont remis en forme en entrée de chaque nœud (chapitre 8). Pour cela, nous considérons deux techniques de remise en forme, à savoir : l'annulation de gigue (*jitter cancellation*) et le seau à jetons (*token bucket*). Cette analyse nous permet de déterminer les cas où il est intéressant (i) de ne pas remettre en forme les flux, (ii) de remettre en forme les flux avec la technique de l'annulation de gigue et (iii) de remettre en forme les flux avec la technique du seau à jetons.

Nous proposons ensuite (chapitre 9) des exemples d'applications de nos résultats, notamment en proposant une architecture de qualité de service combinant DiffServ (*Differentiated Services*) et MPLS (*MultiProtocol Label Switching*). Nous montrons en effet comment mettre en place, dans une telle architecture, un contrôle d'admission garantissant aux flux acceptés le respect de leurs contraintes temporelles de bout-en-bout. Une généralisation à d'autres architectures de QoS est alors décrite.

- **Partie IV : Conclusion**

Dans la quatrième et dernière partie, nous concluons sur le travail réalisé dans cette thèse et présentons plusieurs perspectives intéressantes pour la suite de cette étude.

CHAPITRE 2

Problématique

2.1	Introduction	14
2.1.1	Inadéquation d'un traitement uniforme	14
2.1.2	Dimension temporelle de la qualité de service	15
2.1.3	Importance de l'ordonnancement	15
2.2	Modèles considérés	15
2.2.1	Modèle de réseau	15
2.2.2	Modèle de flux	16
2.2.3	Modèle d'ordonnancement	17
2.3	Cas étudiés	21
2.3.1	Cas monoprocesseur	21
2.3.2	Cas d'une ligne de diffusion	22
2.3.3	Cas général	22
2.4	Outil de validation	22
2.5	Problématique traitée	23
2.6	Conclusion	24

2.1 INTRODUCTION

Nous nous intéressons dans cette thèse à la dimension temporelle de la QoS dans un réseau. Pour cela, nous considérons un réseau dans lequel coexistent n flux différents. Un flux est une séquence de paquets ayant des caractéristiques communes. Dans un réseau de type Internet, par exemple, un flux est identifié par les cinq champs suivants de l'entête TCP/IP [19] : les adresses IP source et destination, les ports source et destination et le protocole de transport (TCP, UDP ou autre). Dans la suite, nous considérerons un flux comme une séquence de paquets ayant ces cinq champs IP identiques, une QoS égale et le même identificateur, défini par l'utilisateur. Dans ce contexte, nous souhaitons fournir à chacun des flux parcourant le réseau des garanties déterministes sur son temps de réponse de bout-en-bout et sa gigue de bout-en-bout¹. Pour cela, nous adoptons trois modèles, détaillés dans la section 2.2.

Mais avant de détailler les trois modèles adoptés, nous revenons sur l'inadéquation d'un traitement uniforme pour le problème posé. Nous présentons alors le type de réseau considéré, puis nous définissons précisément la dimension temporelle de la QoS. Enfin, nous soulignons l'importance de l'ordonnancement pour la QoS.

Puis, après avoir décrit les modèles de réseau, de flux et d'ordonnancement que nous utiliserons tout au long de cette thèse, nous présentons les trois cas dans lesquels nous apportons des résultats nouveaux par rapport à l'état de l'art², à savoir (i) en contexte monoprocesseur, (ii) pour une ligne de diffusion et (iii) dans le cas général. A chaque cas correspond un chapitre de cette thèse.

Une définition formelle de la problématique traitée dans cette thèse est alors donnée à la fin de ce chapitre.

2.1.1 Inadéquation d'un traitement uniforme

Nous considérons un réseau dans lequel coexistent des applications aux exigences variées, telles que des applications critiques d'entreprises, des applications multimédia et des applications de messagerie ou de transfert de fichiers. Pour satisfaire les différentes contraintes, un traitement uniforme serait inadéquat. En effet, même avec un surplus de bande passante, les flux importants (c'est-à-dire d'applications critiques ou temps-réel) ne sont pas privilégiés. Ils peuvent donc être pénalisés par des flux moins prioritaires. Par exemple, un paquet multimédia, très sensible au délai d'acheminement, peut être gêné plusieurs fois dans le réseau par un paquet volumineux lié au transfert d'un fichier. Le paquet multimédia subit alors des délais d'attente préjudiciables dans les nœuds du réseau, sans pour autant que les liens soient saturés. Par ailleurs, il n'est pas possible avec un traitement uniforme d'allouer de la bande passante à des applications spécifiques.

Nous supposons donc un réseau proposant des services différenciés en fonction des exigences des flux, tel IntServ ou DiffServ, afin de privilégier le traitement des applications sensibles sans surdimensionner le réseau. Cette différenciation consistera ici à prendre en compte deux critères de QoS spécifiés par chacun des flux, à savoir :

- un degré d'importance, représentant la criticité du flux ;
- un paramètre temporel, utilisé pour départager les flux de même degré d'importance.

Ces paramètres sont définis en détails dans la section 2.2.3, page 17.

¹La nécessité de telles garanties et l'importance de ces paramètres temporels ont été discutées dans le chapitre précédent.

²L'état de l'art fait l'objet du chapitre 3.

2.1.2 Dimension temporelle de la qualité de service

Nous nous focalisons sur le paramètre temporel de la qualité de service, à savoir le temps de réponse de bout-en-bout et éventuellement la gigue de bout-en-bout. Plus précisément, nous considérons un réseau tel que celui décrit dans la sous-section précédente et dans lequel coexistent n flux. Nous avons alors les deux définitions suivantes :

Définition 2.1.1 *Le temps de réponse pire cas de bout-en-bout d'un flux est le temps maximum mis par un de ses paquets pour traverser le réseau.*

Définition 2.1.2 *La gigue de bout-en-bout d'un flux est la différence entre les temps maximum et minimum mis par deux de ses paquets pour traverser le réseau.*

Afin de proposer des garanties déterministes sur le temps de réponse de bout-en-bout d'un flux quelconque parcourant le réseau, il convient de déterminer une borne sur le temps de réponse pire cas du flux. Pour cela, nous proposons une analyse pire cas basée sur une approche par trajectoire³. Concernant la gigue de bout-en-bout, deux techniques existent pour la limiter, voire l'annuler, à savoir :

- la technique de limitation de gigue, consistant à vérifier que la gigue d'un flux reste bornée par une valeur acceptable maximale avant que le flux ne soit considéré par l'ordonnanceur. Si ce n'est pas le cas, la gigue est réduite à sa valeur acceptable maximale par des mécanismes de remise en forme ;
- la technique de l'annulation de gigue, consistant à annuler sur chacun des nœuds visités la gigue du flux avant qu'il ne soit considéré par l'ordonnanceur [20]. Pour ce faire, les paquets sont retenus jusqu'à leurs instants d'arrivée au plus tard. Ainsi, les paquets arrivent sur chacun des nœuds avec une gigue égale à celle introduite par le nœud précédent et le lien parcouru.

Nous verrons en détail ces techniques dans le chapitre 8.

2.1.3 Importance de l'ordonnancement

Le temps mis par un paquet pour traverser le réseau peut se décomposer en (i) un délai fixe (temps de propagation et temps minimum de traitement) qui dépend du chemin suivi et (ii) un délai dans les files d'attente des nœuds [21]. Le temps passé dans les files d'attente est fonction de la gestion de ces files d'attente, c'est-à-dire de l'ordonnancement. En effet, un algorithme d'ordonnancement [22] détermine, parmi les paquets en attente de traitement, le prochain paquet à transmettre. Il est donc important de s'interroger sur l'ordonnancement à appliquer dans chacun des nœuds du réseau lors de la mise en place d'une architecture de QoS. Dans cette thèse, nous proposons un modèle d'ordonnancement (voir section 2.2.3) permettant de fournir des garanties déterministes sur le temps de réponse de bout-en-bout et la gigue de bout-en-bout de tout flux parcourant le réseau.

2.2 MODÈLES CONSIDÉRÉS

2.2.1 Modèle de réseau

Nous considérons un réseau fiable, c'est-à-dire sans défaillances des processeurs ou du réseau et sans pertes de paquets. Par ailleurs, les liens reliant les différents nœuds sont supposés FIFO : les paquets arrivent sur le nœud $h + 1$ dans l'ordre où ils ont été émis sur le nœud h . De plus, le délai réseau entre deux nœuds est borné inférieurement par L_{min} et supérieurement par L_{max} .

³Voir le chapitre 3 pour plus de détails.

Chacun des nœuds du réseau a la structure présentée à la figure 2.1, composée de :

- un *classificateur*, qui insère les paquets dans les files d'attente de l'ordonnanceur en fonction de leurs priorités ;
- un *ordonnanceur*, qui gère un ensemble ⁴ de files d'attente et sélectionne parmi les paquets en attente, selon un algorithme prédéfini, le paquet qui sera traité par le processeur ;
- un *processeur*, qui transmet les paquets sur le lien de sortie du nœud.

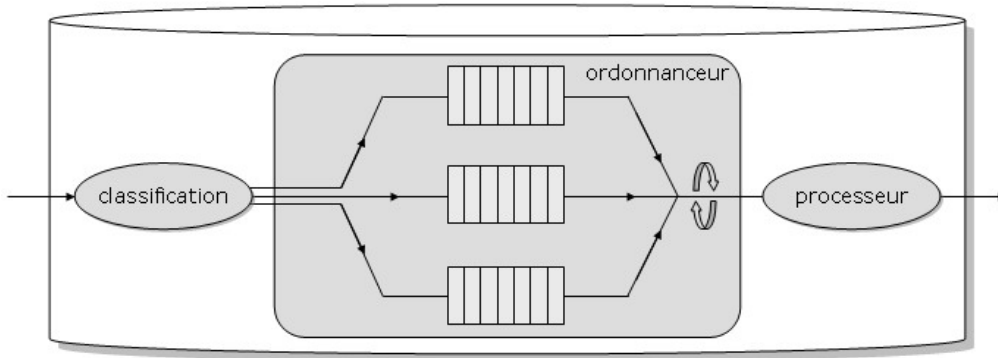


FIG. 2.1: Structure d'un nœud du réseau considéré

L'ordonnanceur peut également contenir un module de remise en forme, permettant de contrôler la conformité des flux ou de limiter, voire d'annuler la distorsion du trafic (dans le cadre d'un contrôle de la gigue). Nous verrons dans le chapitre 8 la mise en œuvre d'un tel mécanisme.

2.2.2 Modèle de flux

Nous considérons un ensemble $\tau = \{\tau_1, \dots, \tau_n\}$ de n flux sporadiques, chacun suivant une *ligne de diffusion*, c'est-à-dire une séquence de nœuds, le premier étant le nœud d'entrée du flux dans le réseau. Un flux sporadique τ_i est défini par :

- T_i , le délai minimum d'interarrivée⁵ entre deux paquets successifs de τ_i ;
- C_i^h , le temps de traitement⁶ maximum d'un paquet de τ_i dans le nœud h ;
- J_i^1 , la gigue d'activation du flux τ_i (voir définition ci-dessous) ;
- D_i , l'échéance de bout-en-bout du flux τ_i , précisant son temps de réponse maximal acceptable.

Nous appellerons *instant de génération* d'un paquet l'instant où est généré ce paquet sur son nœud source. Sur tout autre nœud visité par le paquet, nous parlerons d'*instant d'arrivée*.

Définition 2.2.1 *Un paquet subit une gigue d'activation lorsqu'il existe un temps non nul entre son instant de génération et l'instant de prise en compte du paquet par l'ordonnanceur.*

⁴Il peut n'y avoir qu'une seule file d'attente.

⁵Dans la suite, nous appellerons ce délai la *période* du flux τ_i .

⁶Ce temps inclus les temps de classification et de transmission.

Le modèle sporadique a été formellement introduit par [23], bien que déjà utilisé dans [24]. Cette caractérisation est particulièrement bien adaptée aux flux temps-réel qui ont très souvent un comportement pseudo-périodique. C'est le cas, par exemple, des flux multimedia dans Internet ou encore des applications de contrôle de processus industriels. Les irrégularités affectant la périodicité proviennent de la gigue d'activation qui perturbe les délais interarrivée des paquets.

Remarque 2.2.1 *En monoprocesseur, pour chaque flux τ_i , nous noterons C_i le temps de traitement maximum d'un de ses paquets et J_i sa gigue d'activation. Par ailleurs, D_i représentera son échéance de remise.*

Nous considérons donc un ensemble de flux sporadiques et nous adoptons l'hypothèse 2.2.1.

Hypothèse 2.2.1 *Les instants de génération des paquets de tout flux sporadique τ_i , $i \in [1, n]$, ne sont pas connus à priori.*

Cette hypothèse correspond bien à la nature imprévisible des instants de génération des paquets et permet d'établir des résultats valables quelque soient ces instants.

2.2.3 Modèle d'ordonnancement

Lorsque le lien de sortie est disponible, une politique d'ordonnancement détermine le paquet à transmettre parmi ceux en attente de traitement⁷. Un ordonnancement peut être préemptif ou non-préemptif (définition 2.2.2), oisif ou non-oisif (définition 2.2.3). L'hypothèse généralement admise en ordonnancement est que la transmission des paquets est non-préemptive. Nous supposons donc un ordonnancement non-préemptif. De plus, nous nous intéressons à un ordonnancement non-oisif. Nous présenterons dans le chapitre 8 l'étude d'un ordonnancement oisif.

Définition 2.2.2 *Un ordonnancement est préemptif lorsqu'il est possible d'interrompre le traitement d'un paquet au profit d'un autre paquet plus prioritaire.*

Définition 2.2.3 *Un ordonnancement est oisif lorsqu'il est possible de ne pas ordonnancer un paquet en attente, même si le processeur est disponible.*

Dans cette thèse, nous considérons que chacun des flux entrant dans le réseau spécifie deux paramètres de qualité de service, à savoir :

- un degré d'importance, associée à une priorité fixe. Ce paramètre peut être vu comme la criticité du flux [25]. Le degré d'importance est le critère principal d'ordonnancement ;
- un paramètre temporel, permettant d'attribuer une priorité dynamique à chaque paquet du flux considéré. L'idée est que la priorité dynamique d'un paquet augmente avec le temps. Ce paramètre temporel n'est utilisé que pour départager des flux de même degré d'importance.

Les paquets sont alors ordonnancés dans chacun des nœuds du réseau considéré dans l'ordre de leurs priorités fixes, puis en fonction de leurs priorités dynamiques pour ceux ayant le même degré d'importance.

⁷Nous considérons qu'un paquet est en attente de traitement dans un nœud dès qu'il entre dans la file d'attente de l'ordonnanceur.

Remarque 2.2.2 *L'intérêt d'un tel ordonnancement peut être illustré par l'exemple suivant. Supposons deux applications de type voix sur IP traversant un même réseau. Dans une architecture telle que DiffServ⁸ où le trafic est divisé en un nombre limité de classes, les flux de ces deux applications seront placés dans la même classe de service (typiquement la classe EF) puisqu'ils ont le même degré d'importance. En effet, leurs contraintes de QoS sont les mêmes, notamment leurs contraintes temporelles. Cela dit, si le flux de la première application doit traverser dix nœuds dans le réseau alors que celui de la seconde application ne doit en traverser que deux, il pourrait être intéressant de privilégier le flux "longue distance". La priorité dynamique apporte cette granularité supplémentaire.*

Dans la suite, nous parlerons d'ordonnancement à base de priorités généralisées, défini ci-dessous.

2.2.3.a Ordonnancement à base de priorités généralisées : FP/DP

Nous supposons que dans chacun des nœuds du réseau considéré, les paquets sont ordonnancés suivant leurs priorités généralisées. Une priorité généralisée est l'association d'une priorité fixe et d'une priorité dynamique, déterminées comme suit :

- à tout flux τ_i , $i \in [1, n]$, est assignée une priorité fixe, notée P_i ;
- à tout paquet de τ_i est assignée, sur chaque nœud h visité, une priorité dynamique, notée $P_i^h(t)$, avec t l'instant de génération du paquet considéré. Ainsi, les priorités dynamiques de deux paquets distincts de τ_i sur le nœud h ou d'un même paquet sur deux nœuds différents peuvent varier. En revanche, nous considérerons qu'une fois assignée, la priorité dynamique d'un paquet sur un nœud ne varie plus. Ainsi, l'appel de l'ordonnanceur ne se fait qu'à la fin de traitement d'un paquet ou à l'arrivée d'un nouveau paquet.

Nous notons $PG_i^h(t) = (P_i, P_i^h(t))$ la priorité généralisée sur le nœud h du paquet de τ_i généré à l'instant t . L'algorithme d'ordonnancement à base de priorités généralisées est alors le suivant.

Définition 2.2.4 *Si un nœud quelconque h traite les paquets dans l'ordre de leurs priorités généralisées, alors dès que le processeur devient disponible, l'algorithme d'ordonnancement choisit le paquet ayant la plus forte priorité fixe. Si plusieurs paquets partagent cette priorité fixe, alors l'algorithme d'ordonnancement choisit le paquet ayant la plus forte priorité dynamique.*

Ainsi, dans un nœud h visité par les flux τ_i et τ_j , la priorité généralisée du paquet de τ_i généré à l'instant t est supérieure ou égale à celle du paquet de τ_j généré à l'instant t' si :

$$\left\{ \begin{array}{l} P_i > P_j \\ \text{ou} \\ P_i = P_j \text{ et } P_i^h(t) \geq P_j^h(t'). \end{array} \right.$$

L'ordonnancement à base de priorités généralisés est donc un ordonnancement à base de priorités fixes (noté FP), combiné à un ordonnancement à base de priorités dynamiques (noté DP) afin d'offrir une meilleure QoS aux flux ayant la même priorité fixe. Nous notons un tel ordonnancement FP/DP. Un exemple du calcul de temps de réponse de bout-en-bout est donné dans [26].

⁸Le modèle DiffServ est présenté en détails dans l'état de l'art (chapitre 3).

2.2.3.b Type d'ordonnancement étudié : FP/DP*

Dans cette thèse, nous considérons que la priorité dynamique d'un paquet est attribuée sur son nœud d'entrée dans le réseau puis ne varie plus, ce qui correspond à la tendance actuelle d'effectuer un minimum de traitement dans le cœur du réseau. Par conséquent, pour tout paquet m d'un flux τ_i généré à l'instant t , sa priorité dynamique est égale à $P_i(t)$, quelque soit le nœud visité. Nous parlons alors d'ordonnancement FP/DP*. Puisque la priorité dynamique dépend du temps, cette solution n'exige une synchronisation des horloges que pour les nœuds d'entrée dans le réseau. Par ailleurs, puisque la priorité dynamique d'un paquet est la même sur chacun des nœuds visités, nous avons la propriété suivante.

Propriété 2.2.1 *Si les nœuds du réseau appliquent tous un ordonnancement de type FP/DP*, alors l'ordre des priorités généralisées des paquets est le même sur chacun des nœuds.*

Remarque 2.2.3 *Il est important de souligner que la propriété 2.2.1 n'implique pas que les paquets sont ordonnancés dans le même ordre dans chaque nœud. En effet, considérons le cas illustré par la figure 2.2. La priorité généralisée du paquet m' est supérieure à celle du paquet m . Lorsque m est généré sur le premier nœud, le processeur est libre. Le paquet m est donc immédiatement traité. Le paquet m' , généré après, doit attendre la fin de traitement du paquet m , bien qu'ayant une priorité généralisée supérieure. Sur le nœud suivant, un paquet est en cours de traitement lorsque m et m' arrivent. Le paquet m' , plus prioritaire, est alors traité avant le paquet m .*

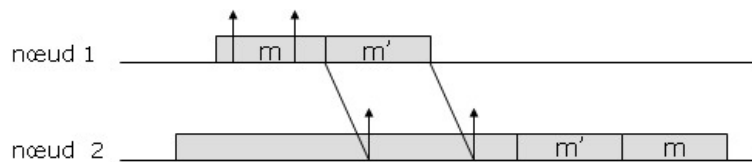


FIG. 2.2: L'ordre d'ordonnancement dépend du nœud

Nous présentons maintenant plus en détail les ordonnancements FP et DP.

2.2.3.b.1 Priorités fixes

Avec un ordonnancement de type FP/DP, les paquets sont donc traités en premier lieu avec un ordonnancement à base de priorités fixes (FP). Ce mécanisme permet d'attribuer de façon stricte une priorité à des flux importants puisque ceux de fortes priorités sont systématiquement privilégiés. Afin de ne pas empêcher les paquets de faibles priorités d'être traités, il est nécessaire de limiter le volume des paquets prioritaires à une petite portion du trafic total.

L'ordonnancement FP a été largement étudié ces dernières années [27, 28] et est utilisé dans la plupart des systèmes informatiques, car il présente les propriétés suivantes :

- il est facile à utiliser et simple à mettre en œuvre ;
- l'impact d'un nouveau flux est limité aux flux ayant une plus faible priorité ;
- il est très bien adapté pour les services différenciés : les flux de fortes priorités sont transmis plus vite.

Cependant, les travaux portant sur cet algorithme ne font aucune hypothèse sur l'ordonnancement des paquets de même priorité et considèrent donc qu'ils sont traités arbitrairement. Pourtant, il n'est pas rare que plusieurs flux partagent une même priorité, notamment dans les cas suivants :

- le nombre de priorités disponibles dans un nœud est inférieur au nombre de flux considérés ;
- les priorités sont déterminées par des contraintes externes et ne peuvent être choisies arbitrairement ;
- un ordonnancement par classe est utilisé ; les flux d'une même classe ont alors la même priorité.

Dans cette thèse, nous améliorons les résultats existants et en établissons de nouveaux pour l'ordonnancement FP en monoprocesseur et en distribué, en considérant que les paquets de même priorité fixe sont ordonnancés suivant leurs priorités dynamiques.

2.2.3.b.2 Priorités dynamiques

Afin d'offrir une meilleure QoS aux utilisateurs en prenant davantage en compte les exigences formulées, nous proposons de traiter les paquets de même priorité fixe avec un ordonnancement à base de priorités dynamiques (DP). La granularité des services proposés est ainsi plus fine puisque des flux ayant la même priorité fixe peuvent être traités différemment. La mise en place d'un algorithme d'ordonnancement dynamique pour des flux d'importance équivalente permet, par exemple, de satisfaire des exigences différentes en terme de temps de réponse ou de bande passante. Nous présentons ci-dessous deux ordonnancements DP particuliers répondant à des besoins spécifiques.

- L'ordonnancement *First In First Out*, ou FIFO [29], est un ordonnancement classique qui consiste à transmettre les paquets dans l'ordre où ils sont arrivés. La priorité dynamique implicite est donc son instant d'arrivée. Il est intéressant d'étudier son comportement temps-réel car il est bien adapté pour la gestion d'une classe de service dans laquelle les flux ont des caractéristiques semblables ou lorsque l'ordonnancement doit refléter l'ordre des événements. De plus, FIFO présente l'avantage (i) de minimiser le temps de réponse maximum d'un ensemble de flux en contexte monoprocesseur et (ii) d'offrir une grande simplicité de traitement.
- L'ordonnancement *Earliest Deadline First*, ou EDF [30], trie les paquets dans l'ordre de leurs échéances absolues. Plus précisément, chaque flux a une échéance relative, représentant sa contrainte sur le temps de réponse. Ainsi, lorsqu'un paquet arrive dans un nœud, l'algorithme d'ordonnancement EDF lui affecte une échéance absolue égale à son instant d'arrivée dans le nœud plus l'échéance relative du flux auquel il appartient. Le paquet ayant la plus petite échéance absolue sera transmis le premier. Il est important de souligner qu'avec EDF, le temps de réponse d'un flux n'est pas lié à la bande passante allouée : un flux ayant réservé une petite partie de la bande passante peut tout de même obtenir un faible temps de réponse. Par ailleurs, la mise en place de ce type d'ordonnancement dans une classe permet d'accroître la granularité du service proposé. Il est alors possible de faire coexister au sein d'une même classe des flux de même importance ayant des exigences temporelles différentes. C'est pourquoi l'étude de l'ordonnancement EDF est intéressante, d'autant que cet algorithme a été prouvé optimal⁹ en contexte monoprocesseur [18] lorsque les instants de demande d'activation ne sont pas connus à priori et qu'il peut être très performant en distribué, comme nous le verrons au chapitre 3.

Nous appliquerons les résultats obtenus pour un ordonnancement de type FP/DP* aux deux ordonnancements suivants : FP/FIFO* et FP/EDF*.

⁹Pour un problème d'ordonnancement donné, si EDF ne trouve pas une solution faisable, alors il n'y en a pas pour ce problème.

2.2.3.c Exemple d'implémentation d'un ordonnancement de type FP/DP

Nous présentons ici un exemple d'implémentation d'un ordonnancement de type FP/DP (ou FP/DP*) dans un nœud quelconque du réseau. En effet, comme le montre la figure 2.3, il est possible d'associer à chaque priorité fixe une file d'attente particulière.

Ainsi, un paquet arrivant dans l'ordonnanceur est inséré dans la file d'attente correspondant à la priorité fixe du flux auquel il appartient. Les paquets d'une file d'attente sont triés dans l'ordre de leurs priorités dynamiques. Lorsque le processeur devient disponible, il traite le premier paquet de la file la plus prioritaire non vide.

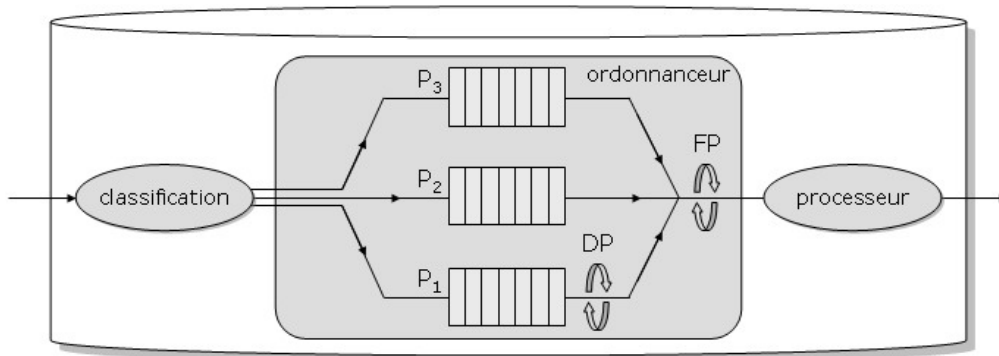


FIG. 2.3: Exemple d'implémentation d'un ordonnancement de type FP/DP

Remarque 2.2.4 *Puisqu'un ordonnancement à base de priorités dynamiques est appliqué à une file d'attente de priorité fixe donnée, il est possible d'implémenter pour chaque file d'attente un ordonnancement DP différent.*

2.3 CAS ÉTUDIÉS

Nous analysons dans ce document le temps de réponse d'un flux sporadique quelconque dans trois cas différents, à savoir (i) en contexte monoprocesseur, (ii) pour une ligne de diffusion et (iii) dans le cas général. Ces cas sont détaillés ci-dessous.

2.3.1 Cas monoprocesseur

En contexte monoprocesseur, les flux sporadiques considérés sont traités dans un seul et même nœud. Ce cas a été largement étudié et de nombreux résultats ont été proposés. Nous présentons dans le chapitre 3 un état de l'art des résultats connus en contexte monoprocesseur lorsque les flux sont ordonnancés FP, FIFO et EDF.

Dans le chapitre 5, nous améliorons les résultats existants pour les ordonnancements FIFO et EDF. Nous montrons également l'intérêt de combiner l'ordonnancement FP à un ordonnancement DP.

2.3.2 Cas d'une ligne de diffusion

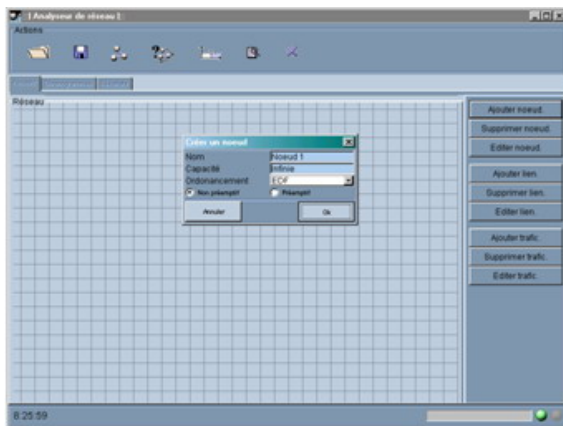
Pour la détermination du temps de réponse de bout-en-bout d'un flux sporadique quelconque dans un réseau, nous adoptons une approche par trajectoire (définie dans l'état de l'art). Afin de présenter clairement la démarche suivie, nous réduisons le cas général au cas où tous les flux suivent dans le réseau la même ligne de diffusion, c'est-à-dire une même séquence de nœuds. Cette hypothèse n'est pas irréaliste. Une ligne louée ou un réseau privé virtuel (ou VPN) peuvent être des exemples du cas considéré. Nous étudions ce cas dans le chapitre 6 et proposons de nouveaux résultats pour les ordonnancements FP, FIFO et EDF. Nous comparons nos résultats (i) à ceux obtenus avec l'approche holistique et (ii) ceux fournis par une approche exhaustive, obtenus avec un outil que nous avons développé et présenté dans la section 2.4.

2.3.3 Cas général

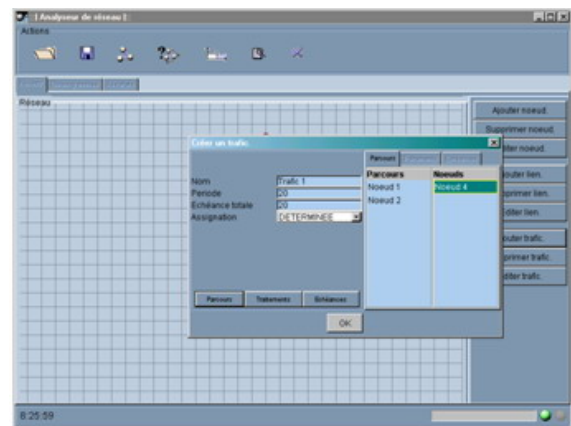
La généralisation du cas d'une seule ligne de diffusion est traitée dans le chapitre 8. Nous considérons dans le cas général que les flux suivent des lignes de diffusion différentes dans le réseau. Nous adoptons une approche par trajectoire pour la détermination du temps de réponse de bout-en-bout d'un flux sporadique quelconque et établissons de nouveaux résultats. Nous présentons alors une évaluation comparative de nos résultats avec ceux obtenus par l'approche holistique et ceux fournis par une approche exhaustive.

2.4 OUTIL DE VALIDATION

Nous avons développé dans le cadre de cette thèse un outil informatique afin de valider les résultats obtenus. Cet outil donne la solution exhaustive à un problème d'ordonnancement temps-réel dans un réseau. En effet, après avoir spécifié (i) le réseau considéré et la politique d'ordonnancement de chacun des nœuds du réseau (figure 2.4.a) et (ii) entré les paramètres des flux (figure 2.4.b), l'outil génère tous les scénarios d'activation possibles et teste pour chacun d'eux la faisabilité des flux. Le nombre de scénarios à tester est borné par le produit des délais interarrivée de chaque flux.



(a) Spécification du réseau considéré

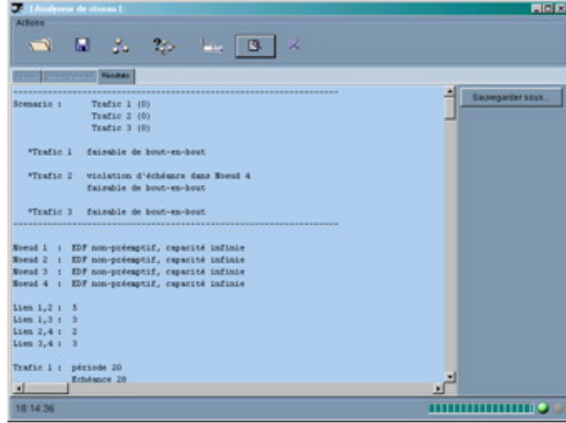


(b) Saisie des paramètres des flux

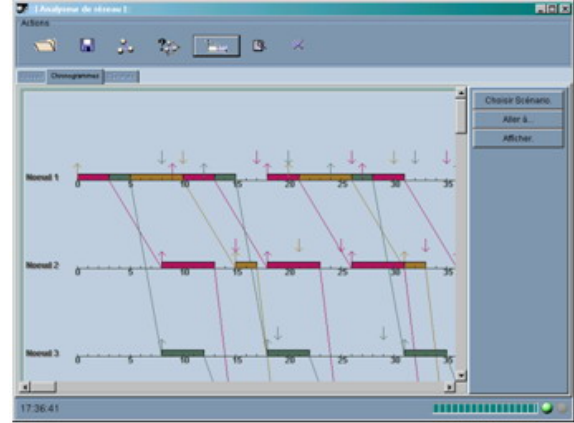
FIG. 2.4: Paramétrage du réseau et des flux

Le résultat de cette simulation est un fichier dans lequel est précisé, pour chaque flux, le temps de réponse maximum atteint, ainsi que le scénario correspondant, nœud par nœud, puis de bout-en-bout (figure 2.5.a).

L'outil permet également de visualiser le déroulement d'un scénario particulier (figure 2.5.b).



(a) Temps de réponse pire cas de chaque flux



(b) Visualisation d'un scénario particulier

FIG. 2.5: Résultats fournis par l'outil de validation

Initialement, cet outil de validation a été développé en Java. Pour des raisons de performances, il a été partiellement porté en C++.

2.5 PROBLÉMATIQUE TRAITÉE

La problématique traitée dans cette thèse peut se résumer comme suit :

Nous considérons un réseau dans lequel coexistent n flux sporadiques différents et supposons que les instants de génération des paquets ne sont pas connus a priori. Tout flux τ_i , $i \in [1, n]$, entrant dans le réseau spécifie deux paramètres de QoS, à savoir son degré d'importance et son paramètre temporel. A partir de ces deux paramètres est attribuée à chacun des paquets du flux τ_i une priorité généralisée, c'est-à-dire la priorité fixe de τ_i et la priorité dynamique du paquet. Les nœuds du réseau traitent alors les paquets selon leurs priorités généralisées, c'est-à-dire suivant un ordonnancement de type FP/DP*¹¹. Plus précisément, les paquets sont traités dans l'ordre de leurs priorités fixes, puis dans l'ordre de leurs priorités dynamiques pour ceux ayant la même priorité fixe. Par ailleurs, nous considérons un ordonnancement non-préemptif et non-oisif.

Dans cette thèse, nous souhaitons fournir à chacun des flux parcourant le réseau des garanties déterministes sur son temps de réponse de bout-en-bout et sa gigue de bout-en-bout. Pour cela, nous procédons à une analyse pire cas du temps de réponse de bout-en-bout d'un flux quelconque τ_i , $i \in [1, n]$. Cette analyse est réalisée dans trois cas distincts : en contexte monoprocesseur, pour une ligne de diffusion et dans le cas général. Puisque nous ne faisons aucune hypothèse sur les instants de génération, les garanties déterministes proposées doivent donc être vérifiées quelques soient les instants d'arrivée des paquets dans le réseau.

¹¹Nous parlons d'ordonnancement de type FP/DP* (et non FP/DP) car nous considérons que la priorité dynamique d'un paquet est attribuée sur le nœud d'entrée puis ne varie plus.

Par ailleurs, dans l'analyse pire cas que nous réalisons, nous supposons que le temps est discret, ce qui correspond à l'évolution du temps cadencé par une horloge. Dans [31], il est démontré que les résultats obtenus avec un ordonnancement discret sont aussi généraux que ceux obtenus avec un ordonnancement continu lorsque tous les paramètres des flux sont multiples du tic de l'horloge. Il s'en suit que, dans ces conditions, tout ensemble de flux est ordonnançable en temps discret si et seulement s'il est ordonnançable en temps continu.

2.6 CONCLUSION

Dans ce chapitre, nous avons présenté la problématique de cette thèse, portant sur un réel besoin de qualité de service dans les réseaux, spécialement en termes de temps de réponse et de gigue de bout-en-bout. En effet, les nouvelles applications telles que la téléphonie sur IP, la vidéo à la demande ou encore les jeux interactifs distribués nécessitent des garanties temporelles déterministes pour fonctionner correctement.

Au début de chapitre, nous avons précisé pourquoi un traitement uniforme des flux était inadéquat pour le problème posé. Nous avons ensuite défini la dimension temporelle de la QoS et souligné l'importance de l'ordonnancement pour la QoS. Puis les différents modèles adoptés dans cette thèse ont été présentés, à savoir les modèles (i) de réseau, (ii) de flux et (iii) d'ordonnancement. Le modèle d'ordonnancement que nous proposons associe priorités fixes et priorités dynamiques, permettant ainsi de traiter des flux ayant un même degré d'importance en fonction d'un second critère qui est temporel.

Après avoir décrit ces trois modèles, nous avons présenté les trois cas considérés dans cette thèse, c'est-à-dire les trois cas pour lesquels nous apportons des résultats nouveaux par rapport à l'état de l'art, à savoir (i) contexte monoprocesseur, (ii) ligne de diffusion et (iii) le cas général.

Enfin, nous avons défini plus formellement la problématique traitée dans cette thèse.

CHAPITRE 3

Etat de l'art

3.1	Introduction	26
3.2	Ordonnancement monoprocesseur	26
3.2.1	Incidence de la non-préemption	27
3.2.2	Ordonnancement non-préemptif	27
3.2.3	Ordonnancement <i>Fixed Priority</i>	28
3.2.4	Ordonnancement <i>First In First Out</i>	30
3.2.5	Ordonnancement <i>Earliest Deadline First</i>	31
3.2.6	Ordonnancements associant priorités fixes et priorités dynamiques	32
3.3	Ordonnancement distribué	33
3.3.1	Approche holistique	33
3.3.2	Network calculus	34
3.4	Architectures de qualité de service	37
3.4.1	Les services intégrés (IntServ)	37
3.4.2	Les services différenciés (DiffServ)	40
3.5	Contrôle d'admission	43
3.6	Conclusion	44

3.1 INTRODUCTION

Ce chapitre est organisé en trois parties.

- La section 3.2 présente des résultats sur l'ordonnancement non-préemptif et non-oisif de flux sporadiques en contexte monoprocesseur, lorsque les instants de génération des paquets ne sont pas connus à priori. Nous rappelons ainsi le temps de réponse pire cas d'un flux quelconque lorsque les flux sont ordonnancés FP (sous-section 3.2.3), FIFO (sous-section 3.2.4) ou EDF (sous-section 3.2.5). Nous utiliserons ces résultats dans le chapitre 5. Nous présentons ensuite trois algorithmes d'ordonnancement combinant priorités fixes et priorités dynamiques, à savoir : FP/MLF, FP/EDF et FP/FIFO.
- La section 3.3 rappelle les approches existantes pour la détermination du temps de réponse pire cas de bout-en-bout d'un flux en environnement distribué, à savoir l'approche holistique (sous-section 3.3.1) et le *network calculus* (sous-section 3.3.2). Dans les chapitres 6 et 7, nous appliquerons une approche dite par trajectoire pour établir de nouveaux résultats en distribué.
- La section 3.4 détaille les deux architectures de qualité de service développées par l'IETF : IntServ (sous-section 3.4.1) et DiffServ (sous-section 3.4.2). Nous verrons dans le chapitre 9 comment appliquer nos résultats à de telles architectures, notamment par la mise en place d'un contrôle d'admission dont le principe est présenté dans la section 3.5.

3.2 ORDONNANCEMENT MONOPROCESSEUR

Nous rappelons que tout flux sporadique τ_i , $i \in [1, n]$, est défini par :

- T_i , sa période, ou plus exactement le délai minimum d'interarrivée entre deux paquets successifs de τ_i ;
- C_i , le temps de traitement maximum d'un de ses paquets ;
- J_i , sa gigue d'activation ;
- D_i , son échéance précisant sa contrainte sur son temps de réponse.

Par ailleurs, nous notons :

- $W_i(t)$ l'instant de démarrage au plus tard du paquet de τ_i généré à l'instant t ;
- $\delta_i(t)$ le délai maximum subi par le paquet de τ_i généré à l'instant t dû à l'effet non-préemptif direct¹ ;
- R_i le temps de réponse pire cas du flux τ_i .

Enfin, nous rappelons la définition du facteur d'utilisation du processeur.

Définition 3.2.1 *Le facteur d'utilisation du processeur représente la fraction du temps processeur utilisée pour l'exécution de l'ensemble des paquets et est égal à : $U = \sum_{j \in [1, n]} C_j / T_j$.*

Remarque 3.2.1 *Une condition nécessaire évidente pour la faisabilité d'un ensemble de flux est que le facteur d'utilisation du processeur soit inférieur ou égal à 1, soit : $U \leq 1$.*

Nous précisons dans la sous-section suivante les conséquences de la non-préemption et décomposons l'effet non-préemptif en deux : direct et indirect. La sous-section 3.2.3 présente ensuite les résultats connus pour l'ordonnancement FP en contexte monoprocesseur.

¹L'effet non-préemptif direct est défini dans la section suivante.

Ces résultats ont été établis en supposant que les paquets de même priorité étaient ordonnancés arbitrairement. Nous améliorerons ces résultats dans le chapitre 5 en considérant que les paquets de même priorité sont ordonnancés soit FIFO (ordonnancement FP/FIFO), soit EDF (ordonnancement FP/EDF). Par ailleurs, nous verrons que, dans le cas particulier où tous les flux partagent la même priorité fixe, l'ordonnancement FP/FIFO (respectivement FP/EDF) revient à ordonnancer un ensemble de flux sporadiques suivant FIFO (respectivement EDF).

Afin de comparer nos résultats avec ceux existants, nous rappelons dans les sous-sections 3.2.4 et 3.2.5 les résultats connus pour ces deux ordonnancements en contexte monoprocesseur.

3.2.1 Incidence de la non-préemption

Nous nous intéressons au temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur, lorsque les flux sont ordonnancés FP, FIFO ou EDF. Ces ordonnancements sont supposés non-préemptifs. Par conséquent, un paquet peut être gêné par un paquet moins prioritaire. En effet, si un paquet m est généré au moment où un paquet m' , de priorité strictement inférieure, est en cours d'exécution, alors m doit attendre la fin de traitement du paquet m' .

Il est important de noter que la non-préemption peut entraîner un retard supplémentaire. En effet, si des paquets plus prioritaires que m sont générés après m mais avant son début d'exécution, alors ils seront traités avant m lorsque le processeur se libérera, comme le montre la figure 3.1.

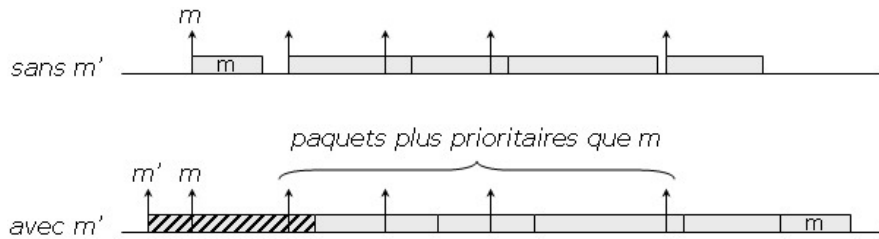


FIG. 3.1: Effet non-préemptif

Soit m un paquet appartenant à flux quelconque τ_i et généré à l'instant t . Si un paquet m' est généré avant t avec une priorité plus faible, alors m peut subir un délai dû à la non-préemption. Nous décomposons ce délai en deux, à savoir :

- le délai subi par m dû à l'effet non-préemptif direct, noté $\delta_i(t)$ et correspondant au temps d'attente de fin d'exécution du paquet m' ;
- le délai subi par m dû à l'effet non-préemptif indirect, correspondant au retard introduit par les paquets plus prioritaires que m qui n'auraient pas été traités avant lui si m' n'avait pas existé.

3.2.2 Ordonnancement non-préemptif

Dans cette thèse, nous considérons que le cas monoprocesseur correspond au cas particulier où le réseau ne comprend qu'un seul nœud. Ainsi, le terme *flux* est similaire au terme *tâche* utilisé classiquement dans la théorie de l'ordonnancement. De même, un *paquet* représente une *instance de tâche*.

Une extension naïve des résultats obtenus pour un ordonnancement préemptif à un contexte non-préemptif serait d'introduire le facteur de blocage comme facteur correctif dû à l'effet non-préemptif (c'est-à-dire la durée maximale de traitement d'une tâche de priorité plus faible que celle considérée, moins une unité de temps si le temps est supposé discret). En effet, cela conduirait à des bornes pessimistes : une tâche τ_j peut retarder une tâche τ_i seulement si elle est activée avant que τ_i ne commence son exécution. Or dans le cas préemptif, une tâche τ_j peut retarder une tâche τ_i seulement si elle est activée avant que τ_i ne termine son exécution. Par conséquent, ajouter le facteur de blocage associé à la non-préemption au temps de réponse obtenu en contexte préemptif serait surestimer le temps de réponse en contexte non-préemptif.

Une borne plus fine a été proposée dans [32], où l'instant de démarrage au plus tard d'une tâche est calculé lorsqu'un ordonnancement à base de priorités fixes est utilisé dans le cadre de messages CAN. L'instant de démarrage au plus tard de la $q^{i\text{ème}}$ instance de la tâche τ_i est donné par :

$$W_{i,q} = q \cdot C_i + \sum_{j \in hp_i} \left\lceil \frac{W_{i,q} + \tau_{res}}{T_j} \right\rceil \cdot C_j + B_i, \text{ où }^2 :$$

- B_i est le facteur de blocage de τ_i dû à la non-préemption et est égal au temps de transmission maximal d'un paquet de plus faible priorité ;
- τ_{res} est la résolution avec laquelle le temps est mesuré. Ce temps de blocage est pessimiste lorsque le temps est supposé discret. En effet, le retard maximum introduit par une tâche de plus faible priorité que τ_i est égal à sa durée de traitement maximale moins une unité de temps, puisqu'elle doit avoir commencé son exécution avant que τ_i arrive.

Il est montré dans [33] que la condition proposée dans [32] est suffisante mais pas nécessaire. En effet, lorsque le temps est supposé discret, pour toute tâche τ_j et pour tout instant t , nous avons : $\lceil \frac{t+1}{T_j} \rceil = 1 + \lfloor \frac{t}{T_j} \rfloor$. En normalisant tous les paramètres par τ_{res} , nous obtenons :

$$W_{i,q} = q \cdot C_i + \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_{i,q}}{T_j} \right\rfloor \right) \cdot C_j + B_i.$$

Nous montrons dans cette thèse qu'il est possible d'améliorer significativement ces résultats en considérant que les tâches de même priorité fixe ne sont pas traitées arbitrairement mais en fonction de leurs priorités dynamiques. D'ailleurs, une implémentation de l'ordonnancement FP applique implicitement une règle d'ordonnancement pour les tâches partageant la même priorité fixe (cette règle est généralement FIFO), mais cette information n'est pas prise en compte dans l'analyse du temps de réponse pire cas.

3.2.3 Ordonnancement *Fixed Priority*

Les résultats suivants sont issus de [34] et [27]. A chaque flux τ_i est associée une priorité fixe, notée P_i . Nous définissons alors :

- $hp_i = \{j \in [1, n], P_j > P_i\}$;
- $sp_i = \{j \in [1, n], j \neq i, P_j = P_i\}$;
- $lp_i = \{j \in [1, n], P_j < P_i\}$.

²Les notations C_i , T_i et hp_i sont présentées dans la sous-section 3.2.3.

Soit m le paquet de τ_i généré à l'instant t . Par définition, les paquets des flux τ_j , $j \in hp_i$, sont tous plus prioritaires que m . Ils ne peuvent cependant gêner m que s'ils sont générés avant son instant de début de traitement, l'ordonnancement étant non-préemptif. Par ailleurs, les paquets des flux τ_j , $j \in lp_i$, sont tous moins prioritaires que m . L'un d'eux peut néanmoins retarder le traitement de m puisque l'ordonnancement est non-préemptif. Enfin, les paquets des flux τ_j , $j \in sp_i$, ont la même priorité que m . Dans l'état de l'art, les paquets de même priorité sont traités arbitrairement.

Avant de présenter la propriété donnant le temps de réponse pire cas d'un flux quelconque τ_i , nous rappelons les trois lemmes suivants, établissant :

- le délai maximum subi par un paquet de τ_i dû à l'effet non-préemptif direct ;
- l'instant de démarrage au plus tard d'un paquet de τ_i généré à l'instant t ;
- les instants à tester pour le calcul du temps de réponse pire cas du flux τ_i .

Lemme 3.2.1 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP, le délai maximum subi par un paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est égal à : $\delta_i(t) = \max(0; \max_{j \in lp_i} \{C_j\} - 1)$, avec $\max_{j \in lp_i} \{C_j\} = 0$ si $lp_i = \emptyset$.*

Lemme 3.2.2 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP, l'instant de démarrage au plus tard du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est égal à :*

$$W_i(t) = \sum_{j \in hp_i \cup sp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max\left(0; \max_{j \in lp_i} \{C_j\} - 1\right).$$

Lemme 3.2.3 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP, le temps de réponse pire cas d'un flux quelconque τ_i est atteint par un paquet généré à un instant $t = -J_i + k \cdot T_i$ tel que $k \in \mathbb{N} \cap [0, K]$, où K est le plus petit entier vérifiant $-J_i + (K + 1) \cdot T_i \geq W_i(-J_i + K \cdot T_i) + C_i$.*

A partir des trois lemmes précédents, nous pouvons établir la propriété suivante, donnant le temps de réponse pire cas d'un flux quelconque τ_i pour l'ordonnancement FP.

Propriété 3.2.1 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP, si $U \leq 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$$R_i = \max_{t \in \mathcal{S}_i} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in hp_i \cup sp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t)$$

et \mathcal{S}_i l'ensemble des instants $t = -J_i + k \cdot T_i$ tels que $k \in \mathbb{N} \cap [0, K]$,

où K est le plus petit entier vérifiant $-J_i + (K + 1) \cdot T_i \geq W_i(-J_i + K \cdot T_i) + C_i$.

Dans le prochain chapitre, nous améliorons ces résultats en associant à l'ordonnancement FP un ordonnancement à base de priorités dynamiques pour les flux de même priorité fixe.

3.2.4 Ordonnancement *First In First Out*

Les résultats suivants sont issus de [35]. Lorsque les flux sont ordonnancés FIFO, il ne peut y avoir d'effet non-préemptif. En effet, un paquet généré à un instant t ne peut être gêné par un paquet moins prioritaire, c'est-à-dire généré après t .

Avant de présenter la propriété donnant le temps de réponse pire cas d'un flux quelconque τ_i , nous rappelons les deux lemmes suivants, établissant (i) l'instant de démarrage au plus tard d'un paquet de τ_i généré à l'instant t et (ii) les instants à tester pour le calcul du temps de réponse pire cas du flux τ_i .

Lemme 3.2.4 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FIFO, l'instant de démarrage au plus tard du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est égal à :*

$$W_i(t) = \sum_{j \neq i, -J_j \leq t} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i.$$

Lemme 3.2.5 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FIFO, le temps de réponse pire cas d'un flux quelconque τ_i est atteint par un paquet généré à un instant $t = t_i^0 + k \cdot T_i < \min(\mathcal{B}_i(t_i^0); \mathcal{P})$, avec $t_i^0 \in [-J_i, -J_i + T_i[$, $k \in \mathbb{N}$, $\mathcal{P} = PPCM_{j \in [1,n]} \{T_j\}$ et :*

$$\mathcal{B}_i(t_i^0) = \sum_{j \neq i, -J_j \leq t} \left\lceil \frac{\mathcal{B}_i(t_i^0) + J_j}{T_j} \right\rceil \cdot C_j + \left\lceil \frac{\mathcal{B}_i(t_i^0) - t_i^0}{T_i} \right\rceil \cdot T_i.$$

Nous pouvons alors définir le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur pour l'ordonnancement FIFO.

Propriété 3.2.2 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FIFO, si $U \leq 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$$R_i = \max_{t \in \mathcal{S}_i} \left\{ \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor\right) \cdot C_j - t \right\}, \text{ avec :}$$

$$\mathcal{S}_i = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i} \mathcal{S}_i(t_i^0) \text{ et } \mathcal{S}_i(t_i^0) \text{ l'ensemble des instants } t = t_i^0 + k \cdot T_i, k \in \mathbb{N}, \text{ tels que } t < \min(\mathcal{B}_i(t_i^0); \mathcal{P}).$$

Nous verrons dans le chapitre 5 que nous améliorons ces résultats, non sur la borne qui est exacte mais sur le nombre d'instant à tester pour la détermination du temps de réponse pire cas.

3.2.5 Ordonnancement *Earliest Deadline First*

Les résultats suivants sont issus de [33]. Nous définissons les deux ensembles suivants :

- $sp_i(t) = \{j \in [1, n], j \neq i, D_j - J_j \leq t + D_i\}$;
- $\overline{sp}_i(t) = \{j \in [1, n], D_j - J_j > t + D_i\}$.

Par construction, si un flux quelconque τ_j appartient à $sp_i(t)$, alors il existe au moins un paquet de τ_j (le paquet généré en $-J_j$) de priorité supérieure ou égale à celle de m , le paquet de τ_i généré à l'instant t . En revanche, si τ_j appartient à $\overline{sp}_i(t)$, alors tous les paquets de τ_j ont une priorité strictement inférieure à celle de m , quelque soient leurs instants de génération.

Avant de présenter la propriété donnant le temps de réponse pire cas d'un flux quelconque τ_i , nous rappelons les trois lemmes suivants, établissant :

- le délai maximum subi par un paquet de τ_i dû à l'effet non-préemptif direct ;
- l'instant de démarrage au plus tard d'un paquet de τ_i généré à l'instant t ;
- les instants à tester pour le calcul du temps de réponse pire cas du flux τ_i .

Lemme 3.2.6 *En contexte monoprocesseur, lorsque les flux sont ordonnancés EDF, le délai maximum subi par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est égal à :*
 $\delta_i(t) = \max(0; \max_{j \in \overline{sp}_i(t)} \{C_j\} - 1)$.

Lemme 3.2.7 *En contexte monoprocesseur, lorsque les flux sont ordonnancés EDF, l'instant de démarrage au plus tard du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est égal à :*

$$W_i(t) = \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t+D_i-D_j; W_i(t)+J_j)}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max\left(0; \max_{j \in \overline{sp}_i(t)} \{C_j\} - 1\right).$$

Lemme 3.2.8 *En monoprocesseur, lorsque les flux sont ordonnancés EDF, le temps de réponse pire cas d'un flux τ_i est atteint par un paquet généré à un instant $t = t_i^0 + k \cdot T_i \leq \max(0; \max_{k \neq i} \{D_k - J_k\} - D_i) + \mathcal{P}$, avec $t_i^0 \in [-J_i, -J_i + T_i]$, $k \in \mathbb{N}$ et $\mathcal{P} = PPCM_{j \in [1, n]} \{T_j\}$.*

A partir des trois lemmes précédents, nous pouvons établir la propriété suivante, donnant le temps de réponse pire cas d'un flux quelconque τ_i pour l'ordonnancement FP.

Propriété 3.2.3 *En contexte monoprocesseur, lorsque les flux sont ordonnancés EDF, si $U \leq 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$R_i = \max_{t \in \mathcal{S}_i} \{W_i(t) - t\} + C_i$, avec :

$$W_i(t) = \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t+D_i-D_j; W_i(t)+J_j)}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max\left(0; \max_{j \in \overline{sp}_i(t)} \{C_j\} - 1\right)$$

et $\mathcal{S}_i = \bigcup_{t_i^0 = -J_i}^{-J_i+T_i} \mathcal{S}_i(t_i^0)$ et $\mathcal{S}_i(t_i^0)$ l'ensemble des instants $t = t_i^0 + k \cdot T_i$, $k \in \mathbb{N}$, tels que $t < \max(0; \max_{k \neq i} \{D_k - J_k\} - D_i) + \mathcal{P}$.

3.2.6 Ordonnancements associant priorités fixes et priorités dynamiques

Nous présentons maintenant trois exemples d'ordonnancements de type FP/DP, à savoir : FP/MLF, FP/EDF et FP/FIFO. Chacun de ces ordonnancements considère le degré d'importance comme le paramètre principal d'ordonnement et la priorité dynamique comme le paramètre secondaire.

3.2.6.a FP/MLF (*Fixed Priority / Maximum Laxity First*)

Cet ordonnancement, où la priorité dynamique est basée sur la laxité, est similaire à MUF (*Maximum Urgency First*) [25, 36]. Un paquet m d'un flux quelconque τ_i , généré à l'instant t' avec un échéance relative D_i , a une laxité égale à : $t' + D_i - C_i(t)$, avec $C_i(t)$ le temps de traitement du paquet m restant à réaliser à l'instant t .

L'avantage de l'ordonnement MLF est sa capacité à détecter le non-respect d'une échéance avant la fin d'exécution d'un paquet (ou avant le début d'exécution d'un paquet en contexte non-préemptif). Cependant, [33] montre que MLF n'est pas optimal en non-préemptif, contrairement à EDF qui reste optimal lorsque les instants d'arrivée des paquets ne sont pas connus à priori.

3.2.6.b FP/EDF (*Fixed Priority / Earliest Deadline First*)

En monoprocesseur, l'ordonnement EDF a été prouvé optimal en contexte préemptif et non-préemptif, lorsque les instants d'activation des paquets ne sont pas connus à priori [28, 18]. En d'autres termes, si EDF ne trouve pas de solution faisable pour un problème d'ordonnement donné, alors il n'existe pas de solution pour ce problème. Un paquet m d'un flux quelconque τ_i , généré à l'instant t' avec une échéance relative D_i , a une échéance absolue égale à : $t' + D_i$. Avec l'ordonnement FP/EDF, la priorité dynamique est basée sur l'échéance absolue.

[37] montre comment calculer les temps de réponse pire cas d'un ensemble de tâches sporadiques ordonnées suivant l'ordonnement FP/EDF en contexte préemptif. Il est également montré comment calculer le temps de blocage maximum, c'est-à-dire le temps maximum perdu à cause de l'exclusion mutuelle. Mais ces résultats ne peuvent pas être appliqués directement au contexte non-préemptif en ajoutant simplement le facteur de blocage associé à la non-préemption. En effet, cela conduirait à un temps de réponse pessimiste (voir la sous-section précédente).

3.2.6.c FP/FIFO (*Fixed Priority / First In First Out*)

FP/FIFO est l'implémentation la plus utilisée de FP. A notre présente connaissance, l'utilisation de FIFO pour arbitrer les flux de même priorité fixe n'est pas pris en compte dans l'analyse pire cas du temps de réponse [27, 28].

Par ailleurs, [38] présente un nouvel algorithme d'ordonnement, nommé RPQ (*Rotating Priority Queues*), où les flux sont ordonnés FP, les paquets de même priorité fixe étant insérés dans la même file d'attente et traités FIFO. L'idée est d'émuler l'ordonnement EDF en échangeant périodiquement les priorités des files d'attente FIFO. Contrairement à EDF, cet ordonnancement ne nécessite pas d'opération de tri dès l'arrivée d'un nouveau paquet dans l'ordonneur. Ici, nous ne souhaitons pas émuler l'ordonnement EDF avec des priorités fixes mais fournir des garanties déterministes sur le temps de réponse et la gigue de chacun des flux lorsque les paquets sont ordonnés FP/DP, la priorité fixe d'un flux n'étant jamais modifiée. En effet, nous considérons le degré d'importance d'un flux comme critère principal, et non son échéance. Nous nous intéressons plus particulièrement aux ordonnancements FP/FIFO et FP/EDF. En effet, FP/FIFO est largement utilisé et FP/EDF domine FP/FIFO sous certaines hypothèses (voir section 5.8, page 75).

3.3 ORDONNANCEMENT DISTRIBUÉ

Deux types d'approche existent pour la détermination du temps de réponse maximum de bout-en-bout d'un flux dans le cas distribué, à savoir :

- *l'approche stochastique*, consistant à étudier le comportement général du réseau. Cette approche conduit à des temps de réponse de bout-en-bout probabilistes ou en moyenne [39, 40] ;
- *l'approche déterministe*, basée sur une analyse pire cas du comportement du réseau. Cette approche conduit à des temps de réponse pire cas de bout-en-bout [41, 20].

Nous nous intéressons ici à la seconde approche puisque nous souhaitons fournir des garanties déterministes sur le temps de réponse de bout-en-bout et la gigue de bout-en-bout de tout flux dans le réseau. Dans ce contexte, la détermination du temps de réponse de bout-en-bout d'un flux peut se faire suivant deux techniques différentes : *l'approche holistique* et le *network calculus*, définies ci-dessous.

3.3.1 Approche holistique

L'approche holistique [42] considère le scénario pire cas sur chacun des nœuds visités par un flux, en considérant la gigue maximale introduite par les nœuds précédents. Si aucun contrôle de la gigue n'est réalisé, la gigue maximale augmente de nœud en nœud. Dans ce cas, le temps de réponse minimum et maximum sur le nœud h induisent une gigue sur le nœud $h + 1$, qui conduit à des temps de réponse minimum et maximum sur ce nœud et donc à une gigue sur le nœud suivant, et ainsi de suite.

Cette approche peut aboutir à une borne très, voire trop pessimiste sur le temps de réponse de bout-en-bout dans la mesure où elle considère des scénarios pire cas sur chacun des nœuds visités, conduisant éventuellement à des scénarios impossibles. En effet, le scénario constitué par la juxtaposition des scénarios pire cas sur chacun des nœuds est généralement irréalisable : un scénario pire cas se déroulant sur un nœud n'engendre pas nécessairement un scénario pire cas sur le nœud suivant.

Plus précisément, considérons un flux quelconque $\tau_i, i \in [1, n]$, suivant une ligne de diffusion composée de q nœuds, numérotés de 1 à q . Nous notons respectivement $R_{max_i^h}$ et $R_{min_i^h}$ les temps de réponse maximum et minimum du flux τ_i sur le nœud h et J_i^h la gigue maximale de τ_i en entrée du nœud h . L'approche holistique procède de manière itérative, en commençant sur le nœud 1.

- Connaissant la valeur de J_i^1 , la gigue d'activation de τ_i , nous calculons $R_{max_i^1}$ en supposant être en contexte monoprocesseur. Par ailleurs, $R_{min_i^1} = C_i^1$.
- Pour tout nœud $h \in]1, q]$, $J_i^h = \sum_{k=1..h-1} (R_{max_i^k} - R_{min_i^k}) + (h - 1) \cdot (L_{max} - L_{min})$. Nous calculons alors $R_{max_i^h}$ en supposant être en contexte monoprocesseur. Par ailleurs, $R_{min_i^h} = C_i^h$;

Une borne sur le temps de réponse de bout-en-bout est donnée par : $\sum_{h=1}^q R_{max_i^h} - \sum_{h=2}^q J_i^h + (q - 1) \cdot L_{max}$. Nous verrons plusieurs exemples d'application de cette approche dans le chapitre 6, lorsque nous comparerons nos résultats avec ceux obtenus par l'approche holistique.

3.3.2 Network calculus

Network Calculus [43] a été développé récemment pour fournir un cadre méthodologique aux problèmes rencontrés dans les réseaux pour les flux. Cet outil se base particulièrement sur la théorie mathématique de l'algèbre *Min-Plus* et permet de comprendre certaines propriétés fondamentales du modèle IntServ (*Integrated Services*), du contrôle de flux par fenêtre, de l'ordonnancement et du dimensionnement des éléments actifs du réseau (délai et *buffer*). En effet, nous avons la propriété suivante.

Propriété 3.3.1 *Considérant (i) un système \mathcal{S} , caractérisé par une courbe de service et (ii) les flux visitant \mathcal{S} , caractérisés par leurs courbes d'arrivée, il est possible de calculer des bornes sur la durée de séjour maximale de chaque flux, la taille maximale de la file d'attente et les courbes de départ des flux.*

Ces résultats sont déterministes si les courbes d'arrivée et de service sont déterministes. Nous présentons dans cette section les définitions et résultats de base du *Network Calculus*.

3.3.2.a Modèles pour les flux

Un flux peut être décrit par une fonction cumulative croissante $R(t)$, définie comme le nombre de bits générés par le flux dans l'intervalle $[0, t]$. Par convention, $R(0) = 0$. Il est possible d'adopter un modèle de temps discret ou continu. Dans les systèmes réels, il existe toujours une granularité minimum (le bit, l'octet, la cellule ou encore le paquet). C'est pourquoi un temps discret, avec un ensemble fini de valeurs pour $R(t)$, devrait toujours être considéré. Cependant, il est souvent plus facile de considérer le temps continu, avec une fonction R continue (dans ce cas, nous parlons de modèle fluide). Par ailleurs, il est toujours possible de transformer un modèle de temps continu $R(t)$ en un modèle de temps discret $S(n)$, $n \in \mathbb{N}$, en choisissant un *slot* de temps δ et en posant : $S(n) = R(n\delta)$. Ainsi, tout résultat obtenu pour un modèle de temps continu peut être appliqué à un modèle de temps discret.

Supposons un système \mathcal{S} (par exemple une liaison, un nœud ou un réseau) recevant en entrée des données décrites sous la forme d'une fonction cumulative $R(t)$ et délivrant en sortie les données après un temps variable. Soit $R^*(t)$ la fonction de sortie, c'est-à-dire la fonction cumulative en sortie du système. D'après les fonctions d'entrée et de sortie, les deux propriétés suivantes peuvent être établies.

Propriété 3.3.2 *La quantité de données $x(t)$ se trouvant dans le système à l'instant t satisfait : $x(t) = R(t) - R^*(t)$.*

Si le système est une file d'attente, cette quantité correspond à la longueur de la file.

Propriété 3.3.3 *Le délai $d(t)$ qui devrait être obtenu par un bit arrivant à l'instant t , si tous les bits reçus avant lui sont servis avant lui, satisfait : $d(t) = \inf \{ \tau \geq 0 : R(t) \leq R^*(t + \tau) \}$.*

Afin de fournir des garanties aux flux, il est nécessaire de limiter le trafic émis par les différentes sources. Dans les réseaux à services intégrés (ATM ou IntServ), ceci est réalisé en utilisant le concept de courbe d'arrivée, défini ci-dessous.

3.3.2.b Courbes d'arrivée et de service

Deux types de courbes d'arrivée sont définis³ : les courbes affines [19] et les courbes en escalier [45]. Dans tous les cas, nous avons la définition suivante.

Définition 3.3.1 *Soit α une fonction croissante définie pour tout $t \geq 0$. Un flux R est contraint par α si et seulement si : $\forall s \leq t, R(t) - R(s) \leq \alpha(t - s)$.*

³Une approche duale est présentée dans [44].

Si le flux $R(t)$ est contraint par α , alors $R(t)$ admet α comme courbe d'arrivée. Les courbes en escalier sont principalement utilisées dans le cadre d'ATM. Ces courbes peuvent également être utilisées pour le modèle sporadique. Sinon, il est généralement considéré qu'un flux est contraint par une courbe d'arrivée affine, souvent représentée sous la forme d'un seau percé (*leaky bucket*). Un seau percé (r, b) est un receptacle contenant un fluide. Le seau est initialement vide et sa capacité est égale à b . Lorsque le seau n'est pas vide, le fluide s'échappe par un trou à un taux de r unités par seconde. Pour contraindre un flux $R(t)$ par un tel mécanisme, il suffit de considérer que le flux remplit le seau d'un volume équivalent à celui des données émises. Les données provoquant le débordement du fluide (c'est-à-dire rendant le volume contenu dans le seau supérieur à b) sont alors déclarées non-conformes. Ainsi, le flux $R(t)$ admet $\gamma_{r,b}$ comme courbe d'arrivée, avec : $\gamma_{r,b} = rt + b$ si $t > 0$ et 0 sinon.

Les architectures IntServ et ATM utilisent une même famille de courbes d'arrivée, correspondant à deux seaux percés combinés : $\alpha(t) = \min(M + pt; rt + b)$, où M est la taille maximale d'un paquet, p est le débit maximum, r est le taux moyen et b la taille de rafale maximale⁴. Dans la terminologie IntServ, le quadruplet (p, M, r, b) est appelé T-SPEC (*Traffic SPECification*).

Il convient donc, dans les réseaux à services intégrés, de déterminer les courbes d'arrivée des flux. Mais pour permettre les réservations de ressources nécessaires, les nœuds du réseau doivent en retour fournir des garanties de service aux flux. Cela est réalisé par les ordonnanceurs [46]. La notion de courbe de service est alors utilisée, afin de ne pas considérer les détails de l'ordonnancement des paquets.

Définition 3.3.2 *Considérons un système S et un flux le traversant, avec des fonctions d'entrée et de sortie R et R^* . Le système S offre à ce flux une courbe de service β si et seulement si $\beta(0) = 0$ et $R^* \geq R \otimes \beta$, où $R \otimes \beta$ représente la convolution Min-Plus définie par : $\forall t \geq 0, (R \otimes \beta)(t) = \inf_{s \leq t} \{R(s) + \beta(t - s)\}$.*

L'utilisation combinée des courbes d'arrivée et des courbes de service permettent d'obtenir les résultats présentés ci-dessous.

3.3.2.c Résultats de base

Connaissant la courbe de service du système et la courbe d'arrivée d'un flux traversant ce système, il est possible d'obtenir les résultats suivants :

- une borne sur la quantité de données se trouvant dans le système à un instant donné ;
- le délai introduit par le système ;
- la courbe du flux de sortie du système.

Propriété 3.3.4 *Soit un flux contraint par une courbe d'arrivée α et traversant un système offrant une courbe de service β . La quantité de données du flux se trouvant dans le système à l'instant t satisfait : $R(t) - R^*(t) \leq \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}$.*

Propriété 3.3.5 *Soit un flux contraint par une courbe d'arrivée α et traversant un système offrant une courbe de service β . Le délai qui devrait être obtenu par un bit arrivant à l'instant t , si tous les bits reçus avant lui sont servis avant lui, satisfait : $d(t) \leq h(\alpha, \beta)$, où $h(\alpha, \beta) = \sup_{s \geq 0} \{\inf\{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}\}$.*

Propriété 3.3.6 *Soit un flux contraint par une courbe d'arrivée α et traversant un système offrant une courbe de service β . Le flux de sortie est contraint par la courbe d'arrivée $\alpha^* = \alpha \oslash \beta$, où $\alpha \oslash \beta$ représente la déconvolution Min-Plus définie par : $(\alpha \oslash \beta)(t) = \sup_{u \geq 0} \{\alpha(t + u) - \beta(u)\}$.*

⁴Nous revenons en détail sur ces différents paramètres dans le chapitre 8.

Il est également possible de déterminer la courbe de service offerte par la succession de plusieurs systèmes, comme le montre la propriété suivante.

Propriété 3.3.7 *Soit un flux traversant les systèmes \mathcal{S}_1 et \mathcal{S}_2 en séquence, offrant respectivement les courbes de service β_1 et β_2 . La concaténation des deux systèmes offre au flux la courbe de service $\beta_1 \otimes \beta_2$, définie par : $(\beta_1 \otimes \beta_2)(t) = \inf_{0 \leq u < t} \{\beta_1(u) + \beta_2(t - u)\}$.*

3.3.2.d Comparaison avec notre approche

Le Network Calculus fournit un ensemble de propriétés formelles permettant de déterminer des bornes pour un flux ou un ensemble de flux contraint par une courbe d'arrivée, notamment sur l'espace mémoire nécessaire dans un nœud quelconque du réseau et la durée de séjour maximale du flux dans le nœud. Ces bornes, jugées pessimistes dans un premier temps, ont été améliorées dans [45]. Une courbe d'arrivée fréquemment utilisée dans l'approche du Network Calculus est celle du seau à jetons, ou *token bucket* [47], représentée par deux paramètres⁵ : σ , la taille de la rafale maximale et ρ , le débit moyen. Un seau à jetons (σ, ρ) fonctionne de la manière suivante. Le seau se remplit de ρ jetons toutes les secondes. Si σ jetons sont déjà présents dans le seau, les jetons supplémentaires sont jetés. Chaque jeton permet de transmettre une certaine quantité d'information (par exemple, un bit). Les paquets sont retenus jusqu'à ce que le nombre nécessaire de jetons soit atteint. Ainsi, dans tout intervalle de longueur t , le nombre de bits injectés dans le réseau est inférieur ou égal à $(\sigma + \rho \cdot t)$.

Dans cette thèse, nous procédons à une analyse pire cas du temps de réponse pour un ensemble de flux sporadiques. Un flux sporadique est caractérisé par un délai minimum d'interarrivée, un temps de traitement maximum et une gigue d'activation (voir section 2.2.2).

Une étude comparative a été menée dans [48] entre ces deux approches pour l'évaluation d'une borne supérieure sur le temps de réponse pour un ensemble de flux périodiques perturbés par des giges et ordonnancés selon l'algorithme des priorités fixes non-préemptif. Afin de comparer ces deux approches, une méthode de transposition du modèle classique vers le modèle (σ, ρ) -borné a été développée. Ainsi, il est possible de contraindre tout flux sporadique $\tau_i = (T_i, C_i, J_i)$ par une courbe d'arrivée $(\sigma_i + \rho_i \cdot t)$. En effet, si $W_i = c \cdot C_i$ représente la longueur d'un paquet de τ_i , avec c la vitesse de traitement du processeur, nous avons :

- $\rho_i = \frac{W_i}{T_i}$;
- $\sigma_i = \frac{W_i}{T_i} \cdot (T_i + J_i)$.

L'étude montre que l'analyse pire cas est plus précise que l'approche du Network Calculus et fournit des bornes plus réalistes, mais est plus coûteuse au niveau du calcul. Par ailleurs, pour un nombre important de sauts dans le réseau, l'approche du Network Calculus s'est avérée peu efficace pour estimer précisément les temps de réponse pire cas. Cette approche reste cependant intéressante pour spécifier le besoin en mémoire ou la taille de la rafale maximale en sortie d'un nœud utilisant le mécanisme des priorités fixes.

Par ailleurs, dans le cadre de réseaux agrégeant le trafic en un nombre limité de classes de service, tel Diff-Serv, se pose le problème de déterminer les conditions nécessaires sur les flux individuels pour que leurs agrégats soient conformes aux courbes d'arrivée. Il est également difficile d'obtenir des bornes pour les flux individuels à partir des caractéristiques obtenues pour leurs agrégats [49, 50].

⁵Le seau à jetons est défini en détail dans le chapitre 8.

Dans cette thèse, nous déterminons des bornes déterministes sur les temps de réponse et les giges de bout-en-bout des flux coexistant dans un réseau. Pour cela, nous réalisons une analyse pire cas en adoptant l'approche dite "par trajectoire" (chapitres 5, 6 et 7). Nous verrons qu'avec cette approche nous obtenons des résultats exacts ou très proches des résultats réels et soulignerons le pessimisme de l'approche holistique. Enfin, dans le chapitre 8, nous présenterons un exemple d'application de nos résultats et montrerons que nos bornes restent valides lorsque la charge du réseau est importante, contrairement aux résultats établis avec l'approche du Network Calculus.

3.4 ARCHITECTURES DE QUALITÉ DE SERVICE

Bien que les problèmes du modèle *best-effort* étaient connus depuis longtemps, le développement d'architectures de qualité de service dans Internet ne commença qu'au début des années 1990. L'IETF proposa ainsi en 1994 un nouveau modèle de service : IntServ (*Integrated Services*) [5, 51]. Dans cette architecture, basée sur la réservation de ressources par flux, une application peut obtenir des garanties en termes de ressources, à condition de les réserver avant de transmettre des paquets dans le réseau. La philosophie du modèle IntServ implique donc que (i) l'application caractérise son flux et ses besoins de QoS et (ii) un protocole réserve les ressources dans chacun des nœuds qui seront visités. Les nœuds sont alors capables d'offrir des garanties de QoS, à condition qu'ils réalisent une classification des flux et qu'ils utilisent des mécanismes d'ordonnancement spécifiques. IntServ propose, en plus du *best-effort*, deux services : CL (*Controlled Load Service*) [52], fournissant à un flux une qualité de service très proche de celle que le flux recevrait avec le service *best-effort* dans un réseau peu chargé et GS (*Guaranteed Service*) [21], offrant des garanties déterministes sur le temps de réponse de bout-en-bout. L'architecture IntServ, le protocole de réservation de ressources RSVP, ainsi que les services proposés sont présentés à la section 3.4.1.

En 1998, l'IETF proposa un autre modèle de service : DiffServ (*Differentiated Services*) [7, 53]. Le modèle IntServ ne pouvant être déployé à grande échelle, cette nouvelle architecture se base sur une approche plus simple, l'agrégation des flux dans un nombre limité de classes de service. Ainsi, un paquet entrant dans le réseau est inclus dans une classe, en fonction du contrat, ou SLA (*Service Level Agreement*), passé entre l'utilisateur et le fournisseur de services. Les nœuds appliquent alors un traitement prédéfini pour transmettre ce paquet. L'avantage de ce modèle est qu'il résiste au facteur d'échelle et que les nœuds dans le cœur du réseau transmettent rapidement les paquets puisque les fonctions complexes telle que la classification sont réalisées par les nœuds en bordure. Deux types de service sont proposés en plus du *best-effort* : AF (*Assured Forwarding*) [54], assurant aux paquets d'être délivrés avec une plus grande probabilité que ceux recevant le service *best-effort* (surtout en cas de congestion) et EF (*Expedited Forwarding*) [55], garantissant des temps de réponse, une gigue et un taux de pertes faibles. La section 3.4.2 détaille l'architecture DiffServ et les services associés.

3.4.1 Les services intégrés (IntServ)

La première avancée majeure pour offrir de la qualité de service dans Internet fut l'architecture des services intégrés, ou IntServ (*Integrated Services*), après qu'il ait été montré que des modifications significatives de l'architecture d'Internet étaient nécessaires pour supporter des applications temps-réel, notamment des conférences audio et vidéo. L'architecture IntServ est basée sur la réservation de ressources, afin d'offrir aux applications temps-réel des garanties en termes de bande passante et de délai dans les files d'attente des différents nœuds du réseau.

3.4.1.a L'architecture IntServ

Dans l'architecture IntServ, un ensemble de mécanismes et de protocoles est utilisé pour faire des réservations explicites de ressources. Ainsi, une application ayant besoin d'un certain niveau de qualité de service doit réserver des ressources tout au long du chemin suivi dans le réseau pour obtenir des garanties de performance. Cette phase de réservation doit être réalisée avant que la transmission des paquets ne commence. Elle nécessite que l'application décrive les caractéristiques du flux qu'elle injectera dans le réseau et ses besoins en termes de ressources. Le réseau peut alors savoir s'il est possible d'accepter cette nouvelle application, en vérifiant les ressources disponibles dans chacun des nœuds qui seront visités.

La qualité de service est donc gérée au niveau du flux⁶. Chaque flux ayant des besoins de qualité de service doit effectuer une réservation de ressources dans les nœuds intermédiaires. Ces nœuds doivent alors maintenir un état par flux. La figure 3.2 présente les composants nécessaires pour qu'un nœud puisse appliquer les mécanismes du modèle IntServ.

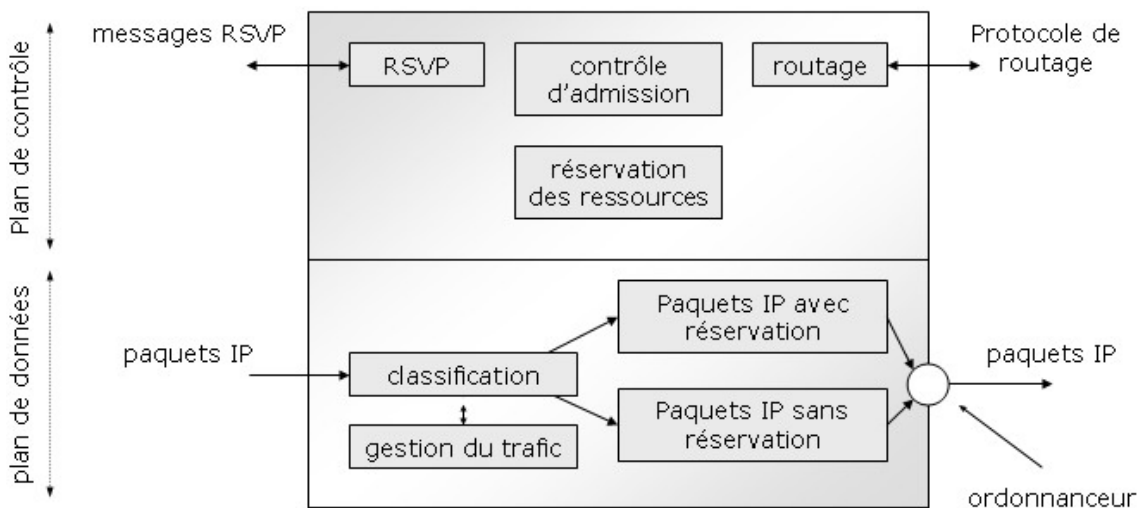


FIG. 3.2: Composants nécessaires dans le modèle IntServ

Cette structure est divisée en deux parties, à savoir :

- le plan de contrôle, permettant de réserver les ressources nécessaires. Pour cela, le routeur comprend un agent RSVP afin d'initialiser et de maintenir les réservations, un contrôle d'admission déterminant si le routeur a les ressources suffisantes pour répondre aux besoins de l'application et un contrôle de règles vérifiant si la demande de réservation est légitime par rapport aux règles fixées par l'administrateur du réseau (par exemple, si la requête est issue d'un utilisateur autorisé) ;
- le plan de données, transmettant les paquets en fonction de l'état des réservations. Lorsque des paquets arrivent, le module de classification sélectionne les paquets appartenant aux flux réservés et les insère dans les files d'attente appropriées. L'ordonnanceur alloue les ressources aux flux en fonction des informations contenues dans la table des réservations. Par ailleurs, le module de gestion du trafic vérifie que les flux restent conformes à leurs spécifications. Les paquets non conformes sont marqués ou éliminés.

⁶Un flux correspond à une séquence de paquets possédant les mêmes source, destination(s) et qualité de service.

3.4.1.b Le protocole RSVP

Dans le modèle IntServ, une application doit établir une réservation de ressources avant de commencer à transmettre des paquets. Pour cela, un nouveau protocole de réservation des ressources était nécessaire. *ReSerVation Protocol*, ou RSVP, fut développé par l'IETF à cet effet [56, 57]. Le principe de réservation, illustré par la figure 3.3, fut influencé par les besoins des applications de vidéoconférence multicast.

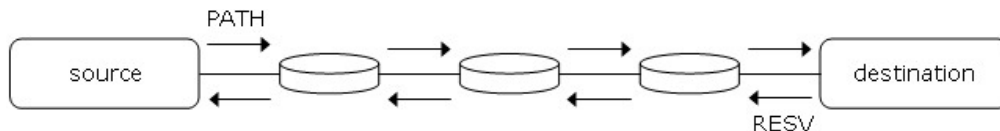


FIG. 3.3: Fonctionnement du protocole RSVP

En effet, le fonctionnement du protocole RSVP est le suivant⁷. La source envoie au destinataire un message PATH (encapsulé dans un datagramme IP et acheminé en *best-effort*), contenant la spécification du flux. Chaque routeur rencontré sur la route vers la destination insère ou modifie dans le message PATH des informations relatives au chemin suivi, ainsi qu'aux ressources et services disponibles.

Grâce à la spécification du flux et aux informations indiquant les possibilités réelles du réseau, le récepteur est capable de déterminer la qualité de service qui peut être demandée. Pour effectuer la réservation, le récepteur renvoie un message RESV parcourant le chemin inverse du message PATH et contenant le type de service désiré (voir la section suivante) et le filtre, à savoir protocole de transport et numéro de port, qui caractérisent les paquets pour lesquels la réservation doit être établie.

La demande de réservation de ressources dans chacun des routeurs recevant le message RESV est soumise au contrôle d'admission et au contrôle des règles (voir figure 3.2). Si la réservation ne peut être satisfaite (par manque de ressources ou échec d'authentification), le routeur retourne une erreur au récepteur. Si la réservation est acceptée, le routeur construit et maintient dynamiquement un état temporaire (*soft state*) pour le flux correspondant, configure la table des réservations de ressources et le module d'identification des flux. Cet état devra être renouvelé périodiquement par des messages PATH et RESV (environ toutes les 30 secondes), sinon l'état sera détruit et les ressources correspondantes libérées.

3.4.1.c Les services proposés

IntServ propose aux applications, en plus du service *best-effort*, les deux services suivants :

- *le service contrôlé*, ou CL (*Controlled Load*) [52]. Ce service fournit à un flux une qualité de service très proche de celle que le flux recevrait avec le service *best-effort* dans un réseau peu chargé. Cela permet l'utilisation d'applications temps-réel adaptatives, qui fonctionnent correctement quand le réseau est peu chargé, mais qui se dégradent très vite lorsque le réseau sature. En supposant que le réseau fonctionne correctement, ces applications ont un très grand pourcentage de paquets délivrés avec succès, avec un délai d'acheminement de bout-en-bout ne dépassant pas de beaucoup le délai minimum mis par n'importe quel paquet délivré avec succès. La définition de CL est volontairement vague.

⁷Le fonctionnement est présenté pour une connexion unicast, c'est-à-dire impliquant une source et une destination. Le principe est le même dans le cadre de connexions multicast.

Un flux recevant ce service dans un élément réseau peut s'attendre à avoir un faible délai moyen dans les files d'attente et peu de pertes dûes aux congestions. Le service contrôlé est donc plus élaboré que le service *best-effort*, mais sans garantie. En effet, ce service ne garantit pas que le temps de traversée des paquets sera borné ou que des paquets ne seront pas perdus à cause de saturations temporaires dans les nœuds du réseau ;

- *le service garanti*, ou GS (*Guaranteed Service*) [21]. Le temps mis par un paquet pour traverser le réseau peut se décomposer en : (i) un délai fixe (temps de propagation) qui dépend du chemin suivi et (ii) un délai dans les files d'attente des routeurs. Le service garanti émule au maximum un circuit virtuel dédié en garantissant une certaine bande passante et en bornant le délai dans les files d'attente. En d'autres termes, les paquets d'un flux bénéficiant de ce service n'auront pas un délai d'acheminement supérieur à une borne mathématiquement calculable. Il est important de noter que GS n'essaie pas de minimiser la gigue, mais contrôle simplement le délai maximum dans les files d'attente. Ce service répond donc aux besoins des applications temps-réel non adaptatives, à savoir débit garanti et délai limité.

3.4.1.d Limites du modèle

Le modèle des services intégrés est confronté au problème d'extensibilité : IntServ résiste mal au facteur d'échelle. En effet, la granularité des services proposés permet de réserver des ressources au niveau du flux (voire du microflux). Les nœuds au cœur du réseau doivent donc traiter un nombre très important de flux simultanément et le coût introduit par la gestion des états peut entraîner une réduction considérable de leurs performances. Par ailleurs, le coût de la signalisation des flux et de la maintenance des états, réalisées par le protocole RSVP, croît proportionnellement au nombre de réservations concurrentes. La perte de performances des routeurs et la charge additionnelle induite par le mécanisme de réservation font que l'architecture IntServ ne peut être déployée dans le cœur du réseau Internet.

L'idée est donc d'assurer la qualité de service dans le cœur du réseau par des mécanismes de contrôle à l'échelle d'agrégat de flux.

3.4.2 Les services différenciés (DiffServ)

Le modèle des services différenciés, ou DiffServ (*Differentiated Services*), fut développé à la fin des années 1990 comme une alternative au modèle IntServ pour les fournisseurs de service. L'approche DiffServ est significativement différente puisqu'elle est basée sur l'agrégation. En effet, DiffServ effectue une différenciation de services en divisant le trafic des utilisateurs en un petit nombre de classes, afin d'éviter le problème d'extensibilité de l'architecture IntServ. Une autre motivation de la définition du modèle DiffServ était la possibilité d'une facturation simple des services proposés, qui devait être compréhensible par les utilisateurs. Ce partitionnement des flux en agrégats permet cela. Par contre, la granularité des services offerts est moins fine que dans le modèle IntServ.

3.4.2.a L'architecture DiffServ

Le principe du modèle DiffServ consiste à séparer les flux dans des classes identifiées par une valeur codée sur six bits, appelée DSCP (*Differentiated Services CodePoint*). Cette agrégation permet d'inclure directement dans l'en-tête IP⁸ d'un paquet la classe de service à laquelle il appartient.

⁸Le DSCP est placé dans le champ *Type of Service* (IPv4) ou dans le champ *Traffic Class* (IPv6).

Ainsi, les opérations complexes (classification, contrôle et marquage des paquets) sont réalisées à l'entrée du réseau sur les nœuds en bordure du domaine DiffServ. Les nœuds dans le cœur du domaine se contentent alors de transmettre les paquets en fonction de leurs DSCP, selon un comportement prédéfini, appelé PHB (*Per Hop Behavior*).

Un domaine DiffServ est une zone administrative dans laquelle l'ensemble des nœuds possède une même définition des services et des PHB. Les nœuds en bordure du domaine sont des nœuds d'entrée (*ingress nodes*), permettant d'insérer les flux dans les classes de service appropriées, ou des nœuds de sortie (*egress nodes*), exécutant un certain nombre de contrôle. La figure 3.4 présente les mécanismes DiffServ utilisés par les nœuds en bordure de domaine.

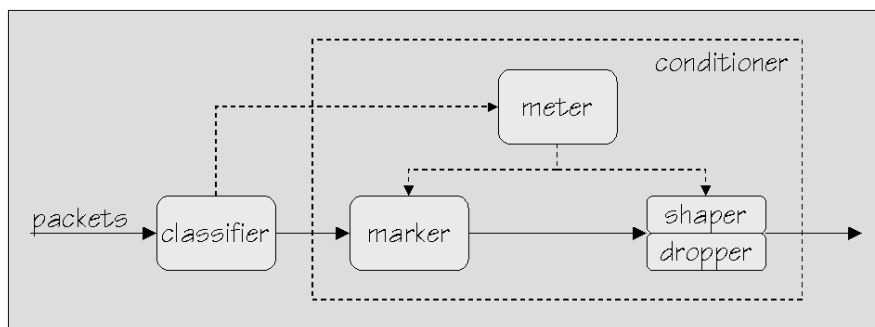


FIG. 3.4: Composants nécessaires dans le modèle DiffServ

La classification des paquets est réalisée en fonction de certaines portions de leurs en-têtes IP. Deux types de classification sont possibles :

- la classification BA (*Behavior Aggregate*), ne se basant que sur le DSCP du paquet ;
- la classification MF (*Multi-Field*), se basant sur les valeurs de plusieurs champs de l'en-tête (tels que l'adresse source, l'adresse destination, le port source, le port destination, etc).

Le module de conditionnement du trafic permet de réaliser des fonctions de mesure, de marquage et de sanction (au cas où le flux serait hors profil). En effet, dans l'architecture DiffServ, les services sont définis sous la forme d'un SLA (*Service Level Agreement*) entre un client et un fournisseur de service. Ce contrat précise les engagements de l'opérateur en ce qui concerne la qualité de service proposée, les métriques associées et les pénalités applicables en cas de non-respect des engagements. Un des éléments importants du SLA, en termes de services différenciés, est le TCA (*Traffic Conditioning Agreement*), détaillant les profils de flux (par exemple, les paramètres d'un *token bucket* pour chaque classe) et les actions à entreprendre pour les paquets non conformes. Le module de conditionnement applique donc les termes du TCA. Pour cela, il peut contenir jusqu'à quatre composants, à savoir :

- un composant de mesure (*meter*), vérifiant que le flux est conforme au profil déterminé ;
- un composant de marquage (*marker*), écrivant ou réécrivant le DSCP ;
- un composant de remise en forme (*shaper*), retardant les paquets non conformes ;
- un composant de rejet (*dropper*), rejetant les paquets non conformes.

Les routeurs dans le cœur du domaine DiffServ classent les paquets en fonction de leurs DSCP (classification BA) puis les transmettent en appliquant le PHB correspondant. Le module de conditionnement n'est donc pas nécessaire dans ce type de nœuds.

3.4.2.b Les services proposés

Le DSCP étant codé sur six bits, il peut y avoir soixante quatre agrégats de flux différents. Le DSCP indique au routeur, dans une perspective de qualité de service, la manière de traiter le paquet. En d'autres termes, un nœud appliquant les mécanismes DiffServ réalise une association entre un ou plusieurs DSCP et un PHB, à l'aide d'une table configurable (dans laquelle le DSCP sert d'index). Il est possible de définir des PHB locaux dans un réseau, cependant, plusieurs PHB ont été standardisés par l'IETF pour trois services suivants :

- *le service par défaut*. Le comportement *best-effort* des réseaux actuels est défini comme le PHB par défaut. Une implémentation de ce service est d'inclure les paquets correspondants dans une file d'attente particulière et d'envoyer des paquets de cette file dès que le lien de sortie n'est pas utilisé par d'autres files d'attente gérant d'autres services ;
- *le service AF (Assured Forwarding)*. Quatre classes sont définies pour ce service, chacune fournissant plus ou moins de ressources (taille des files d'attente et bande passante). La bande passante n'est pas garantie, mais les paquets appartenant à une classe AF ont une priorité plus forte que ceux ne recevant qu'un service par défaut, ils ont donc une plus grande probabilité d'être transmis (surtout en cas de congestion). D'autre part, chaque classe AF gère trois niveaux de rejet différents (faible, moyen et élevé) qui déterminent l'importance relative d'un paquet dans la classe. Ainsi, douze PHB, notés AF_{xy} , peuvent être définis, où x désigne la classe AF et y le niveau de rejet. La spécification du service AF requiert que les paquets provenant d'une même application (d'un même flux) ne soient pas déséquilibrés si seuls leurs niveaux de rejet diffèrent. Une implémentation de ce service peut donc être d'associer à chaque classe AF une file d'attente particulière. Par ailleurs, un algorithme de gestion de la congestion de type RED (*Random Early Detection*) [58, 59] est fréquemment utilisé dans chacune des quatre files d'attente. Ce mécanisme d'élimination de paquets permet de prévenir les congestions en rejetant des paquets si le taux de remplissage de la file atteint un certain seuil, en commençant par les paquets ayant un niveau de rejet élevé ;
- *le service EF (Expedited Forwarding)*. La classe EF fournit un service assimilé à une ligne louée virtuelle, en assurant une garantie de bande passante, ainsi que des temps de réponse, une gigue et un taux de pertes faibles. Pour cela, les points suivants doivent être respectés :
 - à chaque nœud, le taux de départ des paquets doit être supérieur à un taux configurable ;
 - la bufferisation doit être limitée dans les nœuds ;
 - le nœud doit disposer d'une forme d'ordonnancement par priorités statiques ;
 - le trafic sortant doit être lissé pour maintenir le contrat vers un autre domaine ;
 - ce type de trafic doit être limité à une faible portion du trafic total (environ 10%).

En effet, le temps passé dans les files d'attente des routeurs représente la raison principale du retard, de la gigue et de la perte des paquets. Afin d'assurer de meilleures performances, le trafic EF doit être conditionné de telle sorte que le taux d'arrivée maximal des paquets sur chaque nœud soit inférieur au taux de service minimal. Les nœuds en bordure du domaine doivent donc négocier ce taux avec les domaines adjacents, contrôler strictement l'ensemble des flux appartenant à la classe EF et rejeter tous les paquets ne respectant pas cette condition. Cette politique garantit une taille des files d'attente au-dessous d'une certaine limite. Par ailleurs, en offrant à la classe EF la plus forte priorité avec un algorithme de type *priority queuing*, la présence ou non d'autres classes n'a pas d'influence sur la qualité de service offerte.

3.5 CONTRÔLE D'ADMISSION

Nous avons vu que pour des applications temps-réel, comme par exemple les applications multimédia (incluant de la vidéo et/ou du son), les contraintes temporelles doivent être satisfaites ; mais maintenir ou assurer une qualité de service lors de l'arrivée d'un nouveau flux désirant traverser le réseau nécessite d'effectuer un contrôle d'admission [60], afin de vérifier les deux points suivants :

- 1) l'acceptation du nouveau flux ne remet pas en cause les garanties accordées aux flux déjà acceptés ;
- 2) les contraintes de QoS du nouveau flux seront satisfaites.

Le schéma 3.5 illustre le principe de contrôle d'admission.

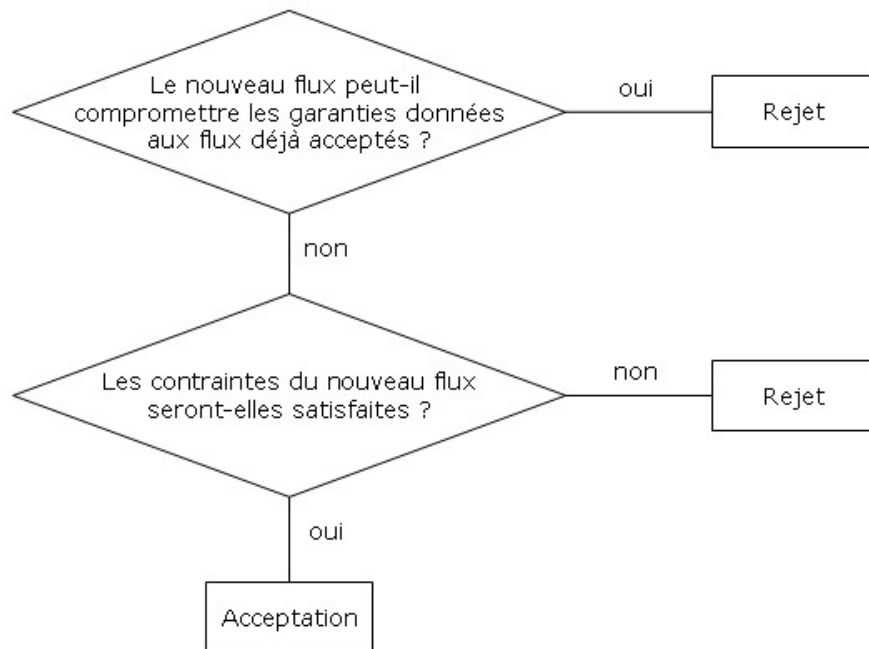


FIG. 3.5: Principe du contrôle d'admission

Le contrôle d'admission permet donc de s'assurer du bon respect des garanties contractées. Il doit être réalisé dans l'optique de maximiser l'utilisation des infrastructures. Le problème d'acceptation de nouveaux flux est critique pour la gestion d'un réseau et fait l'objet de nombreuses recherches. Récemment, la notion de gestionnaire de bande passante (*Bandwidth Broker*) a été introduite [7, 61]. Il s'agit, dans une architecture telle que DiffServ, d'introduire un agent capable de :

- 1) authentifier la demande d'une requête ;
- 2) réaliser un contrôle d'admission, c'est-à-dire vérifier que le niveau de service demandé est compatible avec l'état du domaine ;
- 3) configurer les routeurs se trouvant en bordure du domaine.

Le contrôle d'admission est l'élément crucial de ce type d'agent.

Il existe trois approches différentes pour un contrôle d'admission, à savoir :

- une approche déterministe [62], qui se base sur une analyse pire cas et empêche ainsi toute violation de QoS ;
- une approche statistique [63], qui permet une plus grande utilisation des ressources en contrepartie d'une possibilité non nulle de violation de QoS ;
- une approche basée sur la mesure [64], où les décisions d'admission sont prises en fonction de certains paramètres mesurés dans le réseau.

Par ailleurs, un contrôle d'admission ne pourra être exécuté en-ligne que dans la mesure où il est d'une complexité raisonnable (complexité pseudo-polynômiale). Par conséquent, l'arrivée d'un nouveau flux ne doit pas nécessiter de recalculer les temps de réponse de tous les flux déjà acceptés.

Nous verrons dans le chapitre 9 certaines applications possibles de nos résultats théoriques, notamment la mise en place d'un contrôle d'admission déterministe dans une architecture de QoS pouvant être exécuté en-ligne. Cette architecture est basée sur DiffServ et MPLS (*MultiProtocol Label Switching*). Elle permet d'offrir des garanties déterministes de bout-en-bout à des applications temps-réel.

3.6 CONCLUSION

Dans ce chapitre, nous avons présenté les résultats connus sur l'ordonnancement non-préemptif en mono-processeur. Ainsi, nous avons rappelé le calcul du temps de réponse pire cas d'un flux quelconque lorsque les flux sont ordonnancés FP (sous-section 3.2.3), FIFO (sous-section 3.2.4) ou EDF (sous-section 3.2.5). Dans le chapitre 5, nous comparerons nos résultats à ceux présentés dans l'état de l'art.

Ensuite, nous avons présenté trois algorithmes d'ordonnancement combinant priorités fixes et priorités dynamiques, à savoir : FP/MLF, FP/EDF et FP/FIFO.

Puis nous avons rappelé dans la section 3.3 les approches existantes pour la détermination du temps de réponse pire cas de bout-en-bout d'un flux en environnement distribué, à savoir l'approche holistique (sous-section 3.3.1) et le *network calculus* (sous-section 3.3.2). Nous verrons dans les chapitres 6 et 7, l'application d'une approche dite par trajectoire. Cette approche nous permettra d'établir de nouveaux résultats en distribué.

Enfin, nous avons détaillé dans la section 3.4 les deux architectures de qualité de service développées par l'IETF : IntServ (sous-section 3.4.1) et DiffServ (sous-section 3.4.2). Nous verrons dans le chapitre 9 comment appliquer nos résultats à de telles architectures, notamment par la mise en place d'un contrôle d'admission, dont le principe a été présenté dans la section 3.5.

Deuxième partie

*Nouveaux résultats pour les
ordonnancements de type FP/DP**

CHAPITRE 4

Notations et définitions

4.1	Introduction	48
4.2	En contexte monoprocesseur	49
4.3	En environnement distribué	50
4.3.1	Ligne de diffusion	50
4.3.2	Cas général	51
4.4	Conclusion	51

4.1 INTRODUCTION

Nous présentons dans ce chapitre les différentes notations et définitions que nous utiliserons tout au long de cette deuxième partie, à savoir en contexte monoprocesseur (chapitre 5) et en environnement distribué (chapitres 6 et 7).

Dans tous les cas, nous notons :

- τ_i un des n flux considérés ;
- m paquet du flux τ_i généré à l'instant t ;
- P_i priorité fixe du flux τ_i , $i \in [1, n]$;
- $P_i(t)$ priorité dynamique du paquet m ;
- $PG_i(t)$ priorité généralisée du paquet m .

Ainsi, le paquet m est dit de priorité supérieure ou égale à celle d'un paquet m' d'un flux quelconque τ_j généré à l'instant t' si et seulement si $PG_i(t) \geq PG_j(t')$, c'est-à-dire si :

$$\left\{ \begin{array}{l} P_i > P_j \\ \text{ou} \\ P_i = P_j \text{ et } P_i(t) \geq P_j(t'). \end{array} \right.$$

Nous définissons alors les trois ensembles suivants :

- $hp_i = \{j \in [1, n], P_j > P_i\}$;
- $sp_i = \{j \in [1, n], j \neq i, P_j = P_i\}$;
- $lp_i = \{j \in [1, n], P_j < P_i\}$.

Par conséquent, si $j \in hp_i$ (respectivement lp_i), alors le flux τ_j a une priorité fixe plus grande (respectivement plus petite) que celle du flux τ_i . Si $j \in sp_i$, les flux τ_j et τ_i ont la même priorité fixe. Dans ce cas, les paquets de ces flux sont ordonnancés selon leurs priorités dynamiques. Il est alors intéressant de distinguer (i) les flux susceptibles de générer des paquets de priorité dynamique supérieure ou égale à celle de m et (ii) les flux susceptibles de générer un paquet moins prioritaire que m mais pouvant le gêner dû à l'effet non-préemptif. Ainsi, pour tout instant $t \geq -J_i$, nous définissons :

- $sp_i(t) = \{j \in sp_i, P_j(-J_j) \geq P_i(t)\}$;
- $\overline{sp}_i(t) = \{j \in sp_i, P_j(-J_j) < P_i(t)\}$.

Pour tout flux τ_j , $j \in sp_i(t) \cup \{i\}$, il est intéressant de distinguer l'instant au-delà duquel τ_j ne peut plus générer de paquets plus prioritaires que m . Nous adoptons alors la notation suivante :

- $G_{j,i}(t)$ instant de génération au plus tard d'un paquet de τ_j , $j \in sp_i(t) \cup \{i\}$, tel que :
pour tout instant $t' \in [-J_j, G_{j,i}(t)]$, $P_j(t') \geq P_i(t)$.

Ainsi, tout paquet du flux τ_j généré entre $-J_j$ et $G_{j,i}(t)$ aura une priorité dynamique supérieure ou égale à celle de m . Autrement dit, tout paquet de τ_j généré après l'instant $G_{j,i}(t)$ aura une priorité dynamique strictement inférieure à celle de m . Par ailleurs, nous avons $G_{i,i}(t) = t$.

A titre d'exemple, notons que :

- si l'ordonnancement est FP/FIFO, alors : $\forall j \in sp_i(t) \cup \{i\}, G_{j,i}(t) = t$;
- si l'ordonnancement est FP/EDF, alors : $\forall j \in sp_i(t) \cup \{i\}, G_{j,i}(t) = t + D_i - D_j$.

Nous définissons ci-dessous un instant oisif de niveau p et une période active de niveau p .

Définition 4.1.1 *Un instant oisif t de niveau p est un instant tel que tous les paquets générés avant t avec une priorité généralisée supérieure ou égale à p ont été traités à l'instant t .*

Définition 4.1.2 *Une période active de niveau p est définie par un intervalle $[t, t']$ où t et t' sont deux instants oisifs de niveau p et dans lequel il n'existe aucun autre instant oisif de niveau p .*

4.2 EN CONTEXTE MONOPROCESSEUR

En plus des notations et définitions présentées dans la section 4.1, nous utiliserons dans le chapitre 5 (traitant le cas monoprocesseur) les notations et définitions présentées ci-dessous.

Nous rappelons que tout flux sporadique τ_i , $i \in [1, n]$, est défini par :

- T_i , sa période, ou plus exactement le délai minimum d'interarrivée entre deux paquets successifs de τ_i ;
- C_i , le temps de traitement maximum d'un de ses paquets ;
- J_i , sa gigue d'activation ;
- D_i , son échéance représentant son temps de réponse maximal acceptable.

Par ailleurs, nous noterons :

- τ_i un des n flux sporadiques considérés ;
- R_i temps de réponse pire cas du flux τ_i ;
- m paquet du flux τ_i généré à l'instant t ;
- $\tau(g)$ indice du flux auquel le paquet g appartient ;
- $W_i(t)$ instant de démarrage au plus tard du paquet m ;
- $\delta_i(t)$ délai maximum subi par m dû à l'effet non-préemptif direct¹.

Enfin, nous utilisons les définitions suivantes.

Définition 4.2.1 *Le facteur d'utilisation du processeur pour un flux τ_j représente la fraction du temps processeur utilisée pour l'exécution des paquets de τ_j et est égal à : $U_j = C_j/T_j$.*

Remarque 4.2.1 *Comme cela a été précisé dans l'état de l'art, une condition nécessaire évidente pour la faisabilité d'un ensemble de flux est que le facteur d'utilisation du processeur, noté U et représentant la fraction du temps processeur utilisée pour l'exécution des paquets de l'ensemble des flux τ_i , $i \in [1, n]$, soit inférieur ou égal à 1, avec $U = \sum_{j=1}^n U_j = \sum_{j=1}^n C_j/T_j$.*

¹L'effet non-préemptif direct est défini dans la sous-section 3.2.1, page 27.

4.3 EN ENVIRONNEMENT DISTRIBUÉ

Pour des raisons de clarté, nous décomposons dans cette thèse le cas distribué en deux, à savoir :

- tous les flux suivent la même ligne de diffusion, c'est-à-dire la même séquence de nœuds ;
- les flux suivent des lignes de diffusion quelconques.

Ainsi, après avoir présenté les notations et définitions communes en environnement distribué, nous distinguerons celles utilisées dans le cas d'une ligne de diffusion (chapitre 6) de celles utilisées dans le cas général (chapitre 7). Ces notations et définitions viennent en complément de celles présentées dans la section 4.1.

Nous rappelons que tout flux sporadique τ_i , $i \in [1, n]$, est défini par :

- T_i , sa période, ou plus exactement le délai minimum d'interarrivée entre deux paquets successifs de τ_i ;
- C_i^h , le temps de traitement maximum d'un de ses paquets dans le nœud h ;
- J_i , sa gigue d'activation sur le premier nœud visité ;
- D_i , son échéance de bout-en-bout représentant son temps de réponse maximal acceptable.

Par ailleurs, nous utiliserons la définition suivante.

Définition 4.3.1 *Le facteur d'utilisation du processeur d'un nœud quelconque h pour un flux τ_j représente la fraction du temps processeur de h utilisée pour l'exécution des paquets de τ_j et est égal à : $U_j^h = C_j^h / T_j$.*

Notation Pour tout $x \in \mathbb{R}$, nous noterons : $\max(0; \lfloor x \rfloor) = \lfloor x \rfloor^+$.

4.3.1 Ligne de diffusion

Nous présentons ci-dessous les notations utilisées dans le chapitre 6.

- \mathcal{L} ligne de diffusion suivie par les n flux, composée de q nœuds numérotés de 1 à q ;
- $slow$ nœud le plus lent de la ligne \mathcal{L} , c'est-à-dire tel que : $\forall j \in [1, n], \forall h \in \mathcal{L}, C_j^h \leq C_j^{slow}$;
- L_{min} délai réseau minimum entre deux nœuds ;
- L_{max} délai réseau maximum entre deux nœuds ;
- τ_i un des n flux sporadiques considérés ;
- $R_i^{1,q}$ temps de réponse pire cas de bout-en-bout du flux τ_i ;
- m paquet du flux τ_i généré à l'instant t ;
- $\tau(g)$ indice du flux auquel le paquet g appartient ;
- $W_i^q(t)$ instant de démarrage au plus tard du paquet m sur le nœud q ;
- $\delta_i^{1,q}(t)$ délai maximum subi par m dans les nœuds 1 à q dû à l'effet non-préemptif direct ;
- B_i^{slow} quantité satisfaisant $B_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil B_i^{slow} / T_j \rceil \cdot C_j^{slow}$;
- \bar{t}_i^\emptyset plus petit instant tel que $\overline{sp}_i(t) = \emptyset$;
- $S_{min_j}^q$ temps minimum mis par un paquet du flux τ_j depuis son instant de génération pour arriver sur le nœud q . Ce temps prend en compte le temps de traitement minimum sur les nœuds visités et le délai réseau, soit : $S_{min_j}^q = \sum_{h=1}^{q-1} C_j^h + (q-1) \cdot L_{min}$.

Par ailleurs, nous pouvons noter la remarque suivante concernant le facteur d'utilisation des processeurs.

Remarque 4.3.1 Une condition nécessaire évidente pour la faisabilité d'un ensemble de flux dans le cas d'une ligne de diffusion est que le facteur d'utilisation du processeur de tout nœud h , $h \in [1, q]$, noté U^h et représentant la fraction du temps processeur du nœud h utilisée pour l'exécution des paquets de l'ensemble des flux τ_i , $i \in [1, n]$, soit inférieur ou égal à 1. Notons que cette condition est vérifiée si $U^{slow} \leq 1$.

4.3.2 Cas général

Nous présentons ci-dessous les notations utilisées dans le chapitre 7.

- \mathcal{L}_j ligne de diffusion suivie par un flux quelconque τ_j , $j \in [1, n]$;
- τ_i un des n flux sporadiques considérés ;
- \mathcal{L}_i ligne de diffusion suivie par τ_i , composée de q nœuds numérotés de 1 à q ;
- $slow$ nœud le plus lent de la ligne de diffusion \mathcal{L}_i ;
- $slow_j$ nœud le plus lent de la ligne de diffusion \mathcal{L}_i visité par τ_j , c'est-à-dire tel que :
 $\forall h \in \mathcal{L}_i \cap \mathcal{L}_j, C_j^h \leq C_j^{slow_j}$;
- $R_i^{1,q}$ temps de réponse pire cas de bout-en-bout du flux τ_i ;
- m paquet du flux τ_i généré à l'instant t ;
- $\tau(g)$ indice du flux auquel le paquet g appartient ;
- $W_i^h(t)$ instant de démarrage au plus tard du paquet m sur le nœud h , avec $h \in [1, q]$;
- $\delta_i^{1,q}(t)$ délai maximum subi par m dans les nœuds 1 à q dû à l'effet non-préemptif direct ;
- B_i^{slow} quantité satisfaisant $B_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil B_i^{slow} / T_j \rceil \cdot C_j^{slow}$;
- \bar{t}_i^\emptyset plus petit instant tel que $\overline{sp}_i(t) = \emptyset$;
- $S_{min_j}^h$ temps minimum mis par un paquet du flux τ_j depuis son instant de génération pour arriver sur le nœud h ;
- $S_{max_j}^h$ temps maximum mis par un paquet du flux τ_j depuis son instant de génération pour arriver sur le nœud h ;
- $M_i^h(t)$ quantité égale à : $\sum_{k=1}^{h-1} \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^k\} + (h-1) \cdot L_{min}$.

4.4 CONCLUSION

Nous avons présenté dans ce chapitre les notations et définitions que nous utiliserons tout au long de cette deuxième partie, à savoir :

- (chapitre 5) contexte monoprocesseur ;
- (chapitre 6) ligne de diffusion ;
- (chapitre 7) environnement distribué.

Ce chapitre permettra ainsi la lecture des différents résultats établis dans les chapitres suivants.

CHAPITRE 5

Contexte monoprocesseur

5.1	Introduction	54
5.2	Démarche suivie	54
5.3	Analyse pire cas	55
5.3.1	Instant de démarrage au plus tard	55
5.3.2	Temps de réponse pire cas	59
5.4	Avantage de combiner priorités fixes et priorités dynamiques	62
5.5	Cas particuliers	64
5.5.1	Les flux ont tous une priorité fixe différente	64
5.5.2	Les flux ont tous la même priorité fixe	65
5.6	Ordonnancement FP/FIFO	66
5.6.1	Temps de réponse pire cas	67
5.6.2	Ordonnancement FIFO	68
5.6.3	Exemples	69
5.7	Ordonnancement FP/EDF	71
5.7.1	Temps de réponse pire cas	71
5.7.2	Ordonnancement EDF	72
5.7.3	Exemples	73
5.7.4	Région d'ordonnançabilité	74
5.8	Dominance de FP/EDF sur FP/FIFO	75
5.8.1	Propriétés	75
5.8.2	Exemples	78
5.9	Conclusion	79

5.1 INTRODUCTION

Dans le chapitre 3, nous avons présenté les résultats connus en monoprocesseur pour l'ordonnancement FP non oisif et non-préemptif de flux sporadiques dont les instants d'arrivée ne sont pas connus à priori. Ces résultats ont été établis en supposant que les paquets de même priorité étaient ordonnancés arbitrairement. Nous améliorons ici les résultats existants en considérant que les paquets de même priorité fixe sont ordonnancés suivant leurs priorités dynamiques. L'ordonnancement étudié est donc de type FP/DP : un ordonnancement à base de priorités fixes combiné à un ordonnancement à base de priorités dynamiques. Nous rappelons que la priorité fixe est le critère principal d'ordonnancement, la priorité dynamique n'est que le critère secondaire.

Dans ce chapitre :

- nous déterminons le temps de réponse pire cas d'un flux sporadique quelconque en contexte monoprocesseur lorsque les flux sont traités suivant un ordonnancement de type FP/DP ;
- nous montrons l'intérêt de combiner un ordonnancement à base de priorités fixes à un ordonnancement à base de priorités dynamiques ;
- nous étudions deux cas particuliers concernant les priorités fixes des flux et obtenons des résultats pour l'ordonnancement FP seul, puis pour un ordonnancement DP seul ;
- nous appliquons les résultats obtenus à deux ordonnancements DP particuliers, à savoir FIFO et EDF, puis présentons des exemples numériques ;
- nous comparons les résultats obtenus avec ceux existants et prouvons que nos résultats améliorent ceux établis dans l'état de l'art ;
- nous démontrons la dominance de FP/EDF sur FP/FIFO sous certaines hypothèses.

Dans le chapitre suivant, nous verrons comment généraliser ces résultats au cas d'une ligne de diffusion.

5.2 DÉMARCHE SUIVIE

Afin de déterminer le temps de réponse pire cas en contexte monoprocesseur de tout flux τ_i , $i \in [1, n]$, lorsque les paquets sont ordonnancés suivant un algorithme du type FP/DP, nous nous intéressons à un paquet quelconque m de τ_i , généré à l'instant t et procédons comme suit.

- 1) Nous étudions la période active dans laquelle m est traité. Cela nous conduit à la détermination de l'instant de démarrage au plus tard du paquet m qui, en contexte non-préemptif, est l'instant à considérer (et non l'instant de terminaison au plus tard). En effet, les autres paquets ne peuvent plus retarder le traitement de m s'ils sont générés après que ce dernier ait commencé son exécution.
- 2) L'expression de l'instant de démarrage au plus tard du paquet m est une formule récursive que nous analysons, afin de prouver l'existence d'une solution. Nous obtenons alors le temps de réponse pire cas d'un paquet quelconque du flux τ_i . En calculant le maximum des temps de réponse pire cas des paquets de τ_i , nous obtenons le temps de réponse pire cas du flux.
- 3) Afin de ne pas calculer le temps de réponse pire cas de chacun des paquets de τ_i , nous analysons de nouveau l'expression de l'instant de démarrage au plus tard d'un paquet et montrons que seuls certains instants de génération sont pertinents pour le calcul du temps de réponse pire cas de τ_i .
- 4) Enfin, nous établissons l'expression du temps de réponse pire cas du flux τ_i .

5.3 ANALYSE PIRE CAS

Nous déterminons dans cette section le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur, lorsque les flux sont ordonnancés suivant un algorithme du type FP/DP. Pour cela, nous commençons par étudier la période active dans laquelle le paquet m (le paquet de τ_i généré à l'instant t) est traité. Cela nous conduira à la détermination de l'instant de démarrage au plus tard du paquet m . Par ailleurs, nous rappelons que les notations et définitions utilisées dans ce chapitre sont définies page 49.

Remarque 5.3.1 Dans cette analyse, tout paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, généré à un instant $t' < 0$ subit une gigue d'activation et est activé à l'instant 0.

5.3.1 Instant de démarrage au plus tard

Pour déterminer l'instant de démarrage au plus tard du paquet m considéré, nous identifions la période active de niveau $PG_i(t)$ dans laquelle m est traité. Soit f le premier paquet de cette période active avec une priorité généralisée plus grande ou égale à $PG_i(t)$ (voir figure 5.1). Pour des raisons de clarté, nous numérotions successivement les paquets de la période active de niveau $PG_i(t)$ considérée. Ainsi, nous notons $m'-1$ (respectivement $m'+1$) le paquet traité avant (respectivement après) m' . De plus, nous considérons l'instant d'activation du paquet f comme origine des temps.

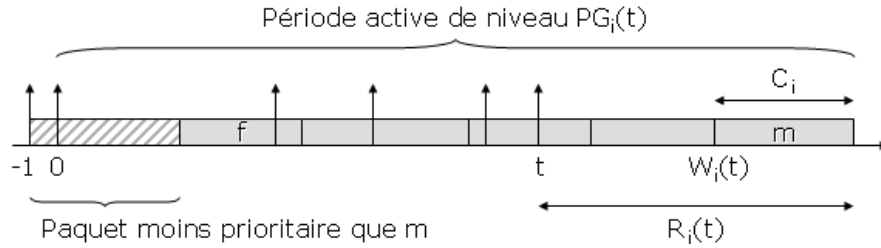


FIG. 5.1: Période active de niveau $PG_i(t)$ considéré

Remarque 5.3.2 Un paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, généré avant $-J_j$ n'appartient pas à la période active de niveau $PG_i(t)$ considérée.

Puisque l'ordonnancement est non-préemptif, le traitement des paquets f à m peut être retardé par au plus un paquet moins prioritaire. Le lemme 5.3.1 précise le délai maximum que le paquet m peut subir à cause de l'effet non-préemptif direct.

Lemme 5.3.1 En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le délai maximum subi par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est égal à : $\delta_i(t) = \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1)$, avec $\max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} = 0$ si $lp_i \cup \overline{sp}_i(t) = \emptyset$.

Démonstration Par définition, les paquets f à m ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$. Leur traitement ne peut donc être retardé que par un seul paquet moins prioritaire, c'est-à-dire appartenant à un flux τ_j tel que $j \in lp_i \cup \overline{sp}_i(t)$. Si un tel paquet est activé en même temps ou après f , il ne sera traité qu'après la période active de niveau $PG_i(t)$ considérée. Le pire est donc que l'exécution du paquet ayant le plus grand temps de traitement parmi les paquets moins prioritaires commence une unité de temps avant f (le temps étant supposé discret [31]). Ainsi, le délai subi par m dû à l'effet non-préemptif direct est au plus égal à : $\max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1$ si $lp_i \cup \overline{sp}_i(t) \neq \emptyset$, 0 sinon. ■

L'instant de démarrage au plus tard du paquet m est donc égal au temps de traitement des paquets f à $m-1$, plus $\delta_i(t)$, le délai maximum introduit par l'effet non-préemptif direct, soit :

$$W_i(t) = \sum_{g=f}^{m-1} C_{\tau(g)} + \delta_i(t). \quad (5.1)$$

Les deux lemmes suivants permettent (i) de déterminer les paquets à considérer dans le terme $\sum_{g=f}^{m-1} C_{\tau(g)}$ et (ii) de maximiser ce terme.

Lemme 5.3.2 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, hormis l'effet non-préemptif direct, le traitement du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est retardé par :*

- les paquets des flux τ_j , $j \in hp_i$, générés dans l'intervalle $[-J_j, W_i(t)]$;
- les paquets des flux τ_j , $j \in sp_i(t)$, générés dans l'intervalle $[-J_j, \min(G_{j,i}(t); W_i(t))]$;
- les paquets du flux τ_i générés dans l'intervalle $[-J_i, t]$.

Démonstration Puisque l'instant d'activation du paquet f (l'instant 0) est un instant oisif de niveau $PG_i(t)$, aucun paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, généré avant $-J_j$ ne peut gêner m , même avec une gigue d'activation maximale. De plus, si $j \in hp_i$, aucun paquet de τ_j ne peut retarder l'exécution de m s'il est généré après $W_i(t)$, l'ordonnancement étant non-préemptif. Par ailleurs, aucun paquet du flux τ_i ne peut gêner m s'il est généré après t .

D'autre part, un paquet d'un flux quelconque τ_j , $j \in sp_i(t)$, a une priorité dynamique supérieure ou égale à celle de m s'il est généré avant $G_{j,i}(t)$. Mais pour gêner m , avoir une priorité généralisée supérieure ou égale à celle de m ne suffit pas ; il faut également que le paquet soit généré avant $W_i(t)$. Par conséquent, seuls les paquets de τ_j générés avant ou en l'instant $\min(G_{j,i}(t); W_i(t))$ peuvent retarder l'exécution de m . ■

Lemme 5.3.3 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le temps de réponse pire cas d'un paquet appartenant à un flux quelconque τ_i et généré à l'instant $t \geq -J_i$ est atteint dans la première période active de niveau $PG_i(t)$ du scénario défini ci-dessous :*

- tout flux τ_j , $j \in hp_i \cup sp_i(t)$, génère son premier paquet à l'instant $-J_j$ puis est périodique ;
- le flux τ_i génère son premier paquet à l'instant $t_i^0 = t - \lfloor (t + J_i)/T_i \rfloor T_i$ puis est périodique ;
- un paquet du flux τ_k , $k \in lp_i \cup \overline{sp}_i(t)$, tel que $C_k = \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\}$ est activé à l'instant -1 .

Démonstration Considérons la période active de niveau $PG_i(t)$ d'un scénario dans lequel le paquet m est traité. L'exécution de m peut être retardé par des paquets de priorités généralisées supérieures ou égales à $PG_i(t)$ et un paquet de priorité généralisée strictement inférieure à $PG_i(t)$. Pour augmenter le temps de réponse du paquet m , il faut donc modifier le scénario considéré afin de maximiser (i) la charge générée par les paquets des flux τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, et (ii) le délai dû à l'effet non-préemptif direct.

Le premier point est obtenu lorsque chacun des flux τ_j , $j \in hp_i \cup sp_i(t)$, génère son premier paquet en $-J_j$, puis est périodique et le flux τ_i génère son premier paquet en $t_i^0 = t - \lfloor (t + J_i)/T_i \rfloor T_i$, puis est périodique. Le second point est obtenu lorsqu'un paquet du flux ayant le plus grand temps de traitement parmi les flux τ_j , $j \in lp_i \cup \overline{sp}_i(t)$, est activé à l'instant -1 (voir lemme 5.3.1).

Dans ce nouveau scénario, le temps de réponse du paquet m reste inchangé ou augmente. ■

Les deux lemmes précédents permettent de maximiser le terme $\sum_{g=f}^{m-1} C_{\tau(g)}$ de l'équation 5.1, c'est-à-dire le retard subi par m dû à des paquets de priorités généralisées supérieures ou égales.

Lemme 5.3.4 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le délai maximum subi par le paquet d'un flux quelconque τ_i généré à l'instant t dû à des paquets de priorités généralisées supérieures ou égales à $PG_i(t)$ est égal à :*

$$\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i.$$

Démonstration D'après le lemme 5.3.3, le terme $\sum_{g=f}^{m-1} C_{\tau(g)}$ est borné par la quantité maximale de travail¹ générée par les paquets définis dans le lemme 5.3.2, soit :

- pour les flux $\tau_j, j \in hp_i$: $\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j$;
- pour les flux $\tau_j, j \in sp_i(t)$: $\sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j$;
- pour le flux τ_i : $\left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i - C_{\tau(m)} = \left\lfloor \frac{t - t + \lfloor (t + J_i)/T_i \rfloor \cdot T_i}{T_i} \right\rfloor \cdot C_i = \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i.$ ■

RÉCAPITULATIF : L'étude de la période active de niveau $PG_i(t)$ dans laquelle le paquet m est traité a permis d'obtenir l'équation 5.1, à savoir : $W_i(t) = \sum_{g=f}^{m-1} C_{\tau(g)} + \delta_i(t)$. Le terme $\delta_i(t)$, représentant le délai maximum subi par m dû à l'effet non-préemptif direct, est maximisé dans le lemme 5.3.1. Le terme $\sum_{g=f}^{m-1} C_{\tau(g)}$, représentant le retard maximum introduit par les paquets de priorités généralisées supérieures ou égales à celle de m , est maximisé dans le lemme 5.3.4.

Nous pouvons alors établir la propriété suivante, précisant l'instant de démarrage au plus tard du paquet m .

Propriété 5.3.1 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, l'instant de démarrage au plus tard du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est égal à :*

$$W_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t),$$

$$\text{avec } \delta_i(t) = \max(0; \max_{j \in lp_i \cup \overline{sp_i}(t)} \{C_j\} - 1).$$

Démonstration D'après l'équation 5.1, $W_i(t) = \sum_{g=f}^{m-1} C_{\tau(g)} + \delta_i(t)$. En appliquant les lemmes 5.3.1 et 5.3.4, nous obtenons la propriété. ■

¹Définie dans [66], la quantité maximale de travail générée par un flux quelconque τ_j dans l'intervalle $[t_1, t_2]$ est le temps nécessaire au processeur pour traiter les paquets de τ_j générés dans cet intervalle, soit : $(1 + \lfloor (t_2 - t_1)/T_j \rfloor) \cdot C_j$.

L'expression de l'instant de démarrage au plus tard du paquet m est donc une formule récursive. Nous montrons la convergence de la suite $\mathcal{W}_i(t)$ définie ci-dessous, afin de prouver l'existence de sa limite : $W_i(t)$.

$$\begin{cases} \mathcal{W}_i^{(0)}(t) = \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t) \\ \mathcal{W}_i^{(k+1)}(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{(k)}(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t). \end{cases}$$

La condition suivante montre que l'équation donnant l'instant de démarrage au plus tard du paquet m a une solution si le facteur d'utilisation du processeur pour les flux τ_j , $j \in hp_i \cup sp_i(t)$, est inférieur à 1.

Condition 5.3.1 Pour tout $t \geq -J_i$, si $U_{j \in hp_i \cup sp_i(t)} < 1$, c'est-à-dire si le facteur d'utilisation des flux τ_j tels que $j \in hp_i \cup sp_i(t)$ est inférieur à 1, alors la suite $\mathcal{W}_i(t)$ est convergente.

Démonstration La suite $\mathcal{W}_i(t)$ est une suite non-décroissante car la fonction *partie entière* ($\lfloor \cdot \rfloor$) est non-décroissante. De plus, si $U_{j \in hp_i \cup sp_i(t)} < 1$, $\mathcal{W}_i(t)$ est bornée supérieurement par $X/(1 - U_{j \in hp_i \cup sp_i(t)})$, où :

$$X = \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t).$$

En effet, par récurrence, nous avons : $\mathcal{W}_i^{(0)}(t) \leq X \leq X/(1 - U_{j \in hp_i \cup sp_i(t)})$ si $U_{j \in hp_i \cup sp_i(t)} < 1$. Nous montrons maintenant que si la récurrence est vraie au rang k , alors elle est vraie au rang $k+1$.

$$\begin{aligned} \mathcal{W}_i^{(k+1)}(t) &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{(k)}(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t) \\ &\leq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t). \end{aligned}$$

Puisque pour tout $x \in \mathbb{R}^+$, $\lfloor x \rfloor \leq x$, il vient :

$$\begin{aligned} \mathcal{W}_i^{(k+1)}(t) &\leq \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{\mathcal{W}_i^{(k)}(t) + J_j}{T_j}\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t) \\ &\leq \mathcal{W}_i^{(k)}(t) \cdot \sum_{j \in hp_i \cup sp_i(t)} \frac{C_j}{T_j} + \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t) \\ &\leq \mathcal{W}_i^{(k)}(t) \cdot U_{j \in hp_i \cup sp_i(t)} + X \leq \frac{X}{1 - U_{j \in hp_i \cup sp_i(t)}} \cdot U_{j \in hp_i \cup sp_i(t)} + X = \frac{X}{1 - U_{j \in hp_i \cup sp_i(t)}}. \end{aligned}$$

La suite $\mathcal{W}_i(t)$ est non-décroissante et bornée supérieurement, elle est donc convergente et la limite vérifie :

$$\mathcal{W}_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t). \blacksquare$$

Remarque 5.3.3 *Il est important de noter que cette condition n'est pas contraignante puisqu'une condition nécessaire pour la faisabilité d'un ensemble de flux est que le facteur d'utilisation du processeur soit inférieur ou égal à 1. Par conséquent, si $U = U_{j=1..n} \leq 1$, alors $U_{j \in hp_i \cup sp_i(t)} < 1$.*

D'après la condition précédente, si $U_{j \in hp_i \cup sp_i(t)} < 1$, alors $W_i(t)$ existe. Nous pouvons alors calculer le temps de réponse pire cas du paquet m , égal à son instant de démarrage au plus tard plus son temps de traitement maximum moins son instant de génération, soit : $W_i(t) + C_i - t$. Nous nous intéressons maintenant au temps de réponse pire cas du flux τ_i , obtenu en calculant le maximum des temps de réponse pire cas de ses paquets, soit : $R_i = \max_{t \geq -J_i} \{W_i(t) - t\} + C_i$.

5.3.2 Temps de réponse pire cas

Le temps de réponse pire cas du flux τ_i est : $R_i = \max_{t \geq -J_i} \{W_i(t) - t\} + C_i$. Nous commençons par déterminer les instants à considérer parmi les instants $t \geq -J_i$. Les trois lemmes suivants permettent ainsi de réduire l'ensemble des instants à tester pour le calcul du temps de réponse pire cas de τ_i .

Pour un paquet de τ_i généré à un instant quelconque $t \geq -J_i$, nous considérons le scénario décrit dans le lemme 5.3.3 pour lequel le temps de réponse pire cas du paquet est atteint dans la première période active de niveau $PG_i(t)$. Dans ce scénario, le flux τ_i génère son premier paquet à l'instant $t_i^0 = t - \lfloor (t + J_i)/T_i \rfloor T_i$ puis est périodique. Nous définissons alors les ensembles suivants :

- $\mathcal{S}_i(t_i^0) = \{t, t = t_i^0 + k \cdot T_i, t_i^0 \in [-J_i, -J_i + T_i[\text{ et } k \in \mathbb{N}\}$;
- $\mathcal{S}_i = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}_i(t_i^0)$.

Nous pouvons maintenant exprimer le temps de réponse pire cas de τ_i de la manière suivante :

$$R_i = \max_{t \in \mathcal{S}_i} \{W_i(t) - t\} + C_i.$$

Afin de limiter le nombre d'instants t à tester pour le calcul de R_i , nous établissons les lemmes 5.3.5 et 5.3.6.

Lemme 5.3.5 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le temps de réponse pire cas d'un flux quelconque τ_i est atteint par un paquet généré à un instant $t = t_i^0 + k \cdot T_i$ tel que $t_i^0 \in [-J_i, -J_i + T_i[$ et $k \in \mathbb{N} \cap [0, K]$, avec K le plus petit entier vérifiant $t_i^0 + (K + 1) \cdot T_i \geq \mathcal{B}_i(t_i^0)$ et $\mathcal{B}_i(t_i^0)$ la longueur de la première période active de niveau $PG_i(t)$ du scénario défini ci-dessous :*

- tout flux τ_j , $j \in hp_i \cup sp_i(t)$, génère son premier paquet à l'instant $-J_j$ puis est périodique ;
- le flux τ_i génère son premier paquet à l'instant $t_i^0 = t - \lfloor (t + J_i)/T_i \rfloor T_i$ puis est périodique ;
- un paquet du flux τ_k , $k \in lp_i \cup \overline{sp}_i(t)$, tel que $C_k = \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\}$ est activé à l'instant -1 .

Par définition, $\mathcal{B}_i(t_i^0) = \sum_{j \in hp_i \cup sp_i(t)} \left\lceil \frac{\mathcal{B}_i(t_i^0) + J_j}{T_j} \right\rceil \cdot C_j + \left\lceil \frac{\mathcal{B}_i(t_i^0) - t_i^0}{T_i} \right\rceil \cdot C_i + \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1)$.

Démonstration D'après le lemme 5.3.3 (page 56), le temps de réponse pire cas du paquet de τ_i généré à l'instant $t = t_i^0 + k \cdot T_i$, avec $t_i^0 \in [-J_i, -J_i + T_i[$ et $k \in \mathbb{N}$, est atteint dans la première période active de niveau $PG_i(t)$ du scénario considéré. Par conséquent, si $\mathcal{B}_i(t_i^0)$ désigne la longueur de cette période active, alors le temps de réponse pire cas du flux τ_i est atteint par un paquet généré à un instant $t \in [-J_i, \mathcal{B}_i(t_i^0)[$. ■

Lemme 5.3.6 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le temps de réponse pire cas d'un flux quelconque τ_i est atteint par un paquet généré à un instant $t = t_i^0 + k \cdot T_i$ tel que $t_i^0 \in [-J_i, -J_i + T_i[$ et $k \in \mathbb{N} \cap [0, K]$, où K est le plus petit entier vérifiant $t_i^0 + (K+1) \cdot T_i \geq W_i(t_i^0 + K \cdot T_i) + C_i$.*

Démonstration Nous calculons $W_i(t) + C_i - t$, le temps de réponse pire cas du paquet de τ_i généré à l'instant t , puis distinguons deux cas :

- si $W_i(t) + C_i > t + T_i$, la période active de niveau $PG_i(t)$ dans laquelle se trouve le paquet de τ_i généré à l'instant t n'est pas terminée à l'instant $t + T_i$. Le temps de réponse du paquet de τ_i généré à l'instant $t + T_i$ doit donc être calculé ;
- si $W_i(t) + C_i \leq t + T_i$, la période active de niveau $PG_i(t)$ dans laquelle se trouve le paquet de τ_i généré à l'instant t est terminée. Le paquet de τ_i généré à l'instant $t + T_i$ appartient à une autre période active. Puisque le pire cas est atteint dans la première période active, il est inutile de calculer le temps de réponse pire cas de ce paquet.

Seuls les instants $t = t_i^0 + k \cdot T_i$ tels que $t_i^0 + k \cdot T_i + T_i < W_i(t_i^0 + k \cdot T_i) + C_i$ doivent donc être testés pour le calcul du temps de réponse pire cas du flux τ_i , avec $t_i^0 \in [-J_i, -J_i + T_i[$ et $k \in \mathbb{N}$. ■

D'après les deux lemmes précédents, nous définissons les ensembles ci-dessous :

- $\mathcal{S}'_i(t_i^0) = \{t, t = t_i^0 + k \cdot T_i, t_i^0 \in [-J_i, -J_i + T_i[, k \in \mathbb{N} \cap [0, K] \text{ et } K \text{ le plus petit entier tel que } t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0))\}$;
- $\mathcal{S}'_i = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}'_i(t_i^0)$.

Le temps de réponse pire cas du flux τ_i est alors égal à :

$$R_i = \max_{t \in \mathcal{S}'_i} \{W_i(t) - t\} + C_i.$$

Le lemme suivant permet de réduire encore le nombre d'instants à tester parmi ceux appartenant à \mathcal{S}'_i .

Lemme 5.3.7 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, le temps de réponse pire cas d'un flux quelconque τ_i est obtenu par un paquet généré à un instant t tel que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.*

Démonstration Nous considérons deux instants, t_1 et t_2 , tels que $G_{j,i}(t_1)$ et $G_{j',i}(t_2)$ soient deux instants consécutifs parmi ceux définis dans l'énoncé du lemme. Nous prouvons que si un paquet de τ_i est généré à un instant $t' \in]t_1, t_2[$, alors son instant de démarrage au plus tard est le même que s'il avait été généré en t_1 . Pour cela, nous procédons par récurrence sur la suite $\mathcal{W}_i(t)$, présentée page 58.

- Par définition, nous avons : $\forall j \in sp_i(t') \cup \{i\}, \lfloor (G_{j,i}(t') + J_j)/T_j \rfloor = \lfloor (G_{j,i}(t_1) + J_j)/T_j \rfloor$. De plus, puisque $G_{i,i}(t') = t'$, il vient $\lfloor (t' + J_i)/T_i \rfloor = \lfloor (t_1 + J_i)/T_i \rfloor$.

- D'autre part, un paquet d'un flux τ_j , $j \in sp_i(t')$, ne peut gêner m que s'il est généré dans l'intervalle $[-J_j, G_{j,i}(t')]$. Puisque nous considérons le scénario décrit dans le lemme 5.3.3, c'est-à-dire dans lequel τ_j est périodique depuis $-J_j$, aucun paquet de τ_j ne peut être généré dans l'intervalle $]G_{j,i}(t_1), G_{j,i}(t_2)[$, deux instants de la forme $-J_l + k_l \cdot T_l$. Nous obtenons donc : $sp_i(t') = sp_i(t_1)$.
- Enfin, puisque $sp_i(t') \cup \overline{sp}_i(t') = sp_i$ pour tout $t' \geq -J_i$, il vient : $\overline{sp}_i(t') = \overline{sp}_i(t_1)$. Par conséquent, nous avons : $\delta_i(t') = \delta_i(t_1)$.

D'après les trois points précédents, $\mathcal{W}_i^{(0)}(t') = \sum_{j \in hp_i \cup sp_i(t')} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t' + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t') = \mathcal{W}_i^{(0)}(t_1)$.

Nous montrons maintenant que si la récurrence est vraie au rang k , alors elle est vraie au rang $k + 1$.

$$\begin{aligned}
\mathcal{W}_i^{(k+1)}(t') &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{(k)}(t') + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t')} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t'), W_i^{(k)}(t')) + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
&\quad + \left\lfloor \frac{t' + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t') \\
&= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{(k)}(t_1) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t_1)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t_1), W_i^{(k)}(t_1)) + J_j}{T_j} \right\rfloor\right) \cdot C_j \\
&\quad + \left\lfloor \frac{t_1 + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t_1) \\
&= \mathcal{W}_i^{(k+1)}(t_1).
\end{aligned}$$

Les deux séries étant convergentes, leurs limites satisfont $\mathcal{W}_i(t') = \mathcal{W}_i(t_1)$. ■

RÉCAPITULATIF : Les trois lemmes précédents ont permis de limiter, puis de réduire le nombre d'instants à tester pour le calcul du temps de réponse pire cas du flux τ_i . En effet, au début de cette section, la définition du temps de réponse pire cas du flux τ_i était la suivante : $R_i = \max_{t \geq -J_i} \{W_i(t) - t\} + C_i$. Les lemmes 5.3.5, 5.3.6 et 5.3.7 ont montré qu'il n'était pas nécessaire de tester tous les instants $t \geq -J_i$ mais seulement ceux de la forme $t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K est le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0))$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$.

Nous établissons alors la propriété suivante, précisant le temps de réponse pire cas du flux τ_i .

Propriété 5.3.2 En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/DP, si $U_{j \in hp_i \cup sp_i(t)} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :

$$R_i = \max_{t \in \mathcal{S}_i''} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i(t)) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t),$$

$$\delta_i(t) = \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1)$$

et $\mathcal{S}_i'' = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}_i''(t_i^0)$, où $\mathcal{S}_i''(t_i^0)$ est l'ensemble des instants $t = t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; B_i(t_i^0))$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$.

Démonstration L'expression de $W_i(t)$ est donnée par la propriété 5.3.1. Son expression est une formule récursive pour laquelle il existe une solution si $U_{j \in hp_i \cup sp_i(t)} < 1$ (voir condition 5.3.1, page 58). Par ailleurs, le temps de réponse pire cas du flux τ_i est égal à $R_i = \max_{t \geq -J_i} \{W_i(t) - t\} + C_i$. D'après les lemmes 5.3.5, 5.3.6 et 5.3.7, seuls les instants t appartenant à l'ensemble \mathcal{S}_i'' sont à considérer (voir récapitulatif page 61). ■

Les résultats établis dans cette section sont valides pour tout ordonnancement FP/DP considéré. Dans la section suivante, nous mettons en évidence les avantages d'un ordonnancement FP/DP par rapport à l'ordonnancement FP.

5.4 AVANTAGE DE COMBINER PRIORITÉS FIXES ET PRIORITÉS DYNAMIQUES

La mise en place d'un ordonnancement à base de priorités dynamiques pour les paquets partageant la même priorité fixe permet d'offrir une meilleure QoS que la seule prise en compte de l'importance d'un flux. Nous verrons dans les deux sections suivantes comment appliquer nos résultats aux ordonnancements FP/FIFO et FP/EDF. Mais avant, nous démontrons que :

- un ensemble de flux faisable avec FP l'est avec FP/DP (propriété 5.4.1) ;
- un ensemble de flux faisable avec FP/DP ne l'est pas nécessairement avec FP (propriété 5.4.2).

Propriété 5.4.1 *En contexte monoprocesseur, tout ensemble de flux faisable avec l'ordonnancement FP est faisable avec un ordonnancement FP/DP.*

Démonstration Avec l'ordonnancement FP, les paquets de même priorité sont traités arbitrairement (voir section 3.2.3, page 28). Par conséquent, tout ensemble de flux faisable avec FP est faisable quelque soit le traitement appliqué aux paquets de même priorité, notamment un ordonnancement de type DP. ■

Propriété 5.4.2 *En contexte monoprocesseur, un ensemble de flux faisable avec un ordonnancement FP/DP n'est pas nécessairement faisable avec l'ordonnancement FP.*

Démonstration Nous nous intéressons aux instants de démarrage au plus tard du paquet m obtenus respectivement avec l'ordonnancement FP et un ordonnancement de type FP/DP. Nous les notons respectivement $W_i^{FP}(t)$ et $W_i^{FP/DP}(t)$. Pour commencer, nous montrons que $W_i^{FP}(t) \geq W_i^{FP/DP}(t)$. Pour cela, nous considérons les suites associées et procédons par récurrence. Au rang 0, nous avons :

$$\mathcal{W}_i^{(0)FP}(t) = \sum_{j \in hp_i \cup sp_i} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in lp_i} \{C_j\} - 1)$$

$$\mathcal{W}_i^{(0)FP/DP}(t) = \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1).$$

Puisque $sp_i(t) \cup \overline{sp}_i(t) = sp_i$, nous obtenons :

$$\sum_{j \in hp_i \cup sp_i} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \max(0; \max_{j \in lp_i} \{C_j\} - 1) \geq \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \left\lfloor \frac{J_j}{T_j} \right\rfloor\right) \cdot C_j + \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1).$$

Par conséquent, $\mathcal{W}_i^{(0)FP}(t) \geq \mathcal{W}_i^{(0)FP/DP}(t)$. Nous montrons maintenant que si la récurrence est vraie au rang k , alors elle est vraie au rang $k+1$. En effet :

$$\begin{aligned} \mathcal{W}_i^{(k+1)FP}(t) &= \sum_{j \in hp_i \cup sp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)FP}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in lp_i} \{C_j\} - 1) \\ &\geq \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)FP}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1) \\ &\geq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)FP}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k)FP}(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t) \end{aligned}$$

Puisque $\mathcal{W}_i^{(k)FP}(t) \geq \mathcal{W}_i^{(k)FP/DP^*}(t) \geq \min(G_{j,i}(t); \mathcal{W}_i^{(k)FP/DP^*}(t))$, il vient :

$$\begin{aligned}
& \sum_{j \in h_{p_i}} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k) \text{ FP}}(t) + J_j}{T_j} \right\rfloor \right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k) \text{ FP}}(t) + J_j}{T_j} \right\rfloor \right) \cdot C_j \\
& \geq \sum_{j \in h_{p_i}} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(k) \text{ FP/DP}^*}(t) + J_j}{T_j} \right\rfloor \right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{(k) \text{ FP/DP}^*}(t)) + J_j}{T_j} \right\rfloor \right) \cdot C_j.
\end{aligned}$$

Nous obtenons donc : $\mathcal{W}_i^{(k+1) \text{ FP}}(t) \geq \mathcal{W}_i^{(k+1) \text{ FP/DP}}(t)$. Les deux séries étant convergentes, leurs limites satisfont $\mathcal{W}_i^{\text{FP}}(t) \geq \mathcal{W}_i^{\text{FP/DP}}(t)$. Le temps de réponse pire cas d'un flux quelconque τ_i obtenu avec l'ordonnancement FP est donc supérieur ou égal à celui obtenu avec un ordonnancement FP/DP.

Si, par exemple, le temps de réponse pire cas de τ_i est égal à son échéance ($R_i = D_i$) avec un ordonnancement FP/DP, alors il est possible que τ_i ne respecte pas son échéance avec l'ordonnancement FP. ■

RÉCAPITULATIF : Nous avons souligné dans cette section l'intérêt d'appliquer aux paquets de même priorité fixe un ordonnancement à base de priorités dynamiques. En effet :

- puisqu'un ordonnancement de type FP/DP considère, pour des paquets de même importance (c'est-à-dire de même priorité fixe) un critère supplémentaire (leurs priorités dynamiques) pouvant représenter un besoin spécifique tel que le temps de réponse maximum acceptable, le traitement appliqué répond davantage aux exigences spécifiques des différents flux ;
- les propriétés 5.4.1 et 5.4.2 ont montré qu'un ordonnancement de type FP/DP offre une meilleure faisabilité que l'ordonnancement FP.

Après avoir déterminé le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur avec un ordonnancement de type FP/DP, puis montré l'intérêt de combiner priorités fixes et priorités dynamiques, nous étudions dans la section suivante deux hypothèses concernant les priorités fixes des flux. Nous obtenons ainsi des résultats en contexte monoprocesseur pour l'ordonnancement FP seul, puis pour un ordonnancement DP seul.

5.5 CAS PARTICULIERS

Nous nous intéressons dans cette section à deux cas particuliers, à savoir :

- 1) les flux ont tous une priorité fixe différente ;
- 2) les flux ont tous la même priorité fixe.

Ces deux cas permettent d'obtenir des résultats en contexte monoprocesseur lorsque les flux sont ordonnancés FP (cas 1) ou DP (cas 2).

5.5.1 Les flux ont tous une priorité fixe différente

Lorsque tous les flux ont une priorité fixe différente, le traitement DP n'intervient pas dans l'ordonnancement FP/DP, puisque DP permet de départager des paquets ayant la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme FP.

Si tous les flux ont une priorité fixe différente, alors pour tout flux $\tau_i : sp_i = \emptyset$. En appliquant ce résultat à la propriété 5.3.2, nous obtenons la propriété suivante.

Propriété 5.5.1 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP, si $U_{j \in hp_i} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$$R_i = \max_{t \in S'_i(-J_i)} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in lp_i} \{C_j\} - 1)$$

et $S'_i(-J_i)$ l'ensemble des instants t de la forme $-J_i + k \cdot T_i$ tels que $k \in \mathbb{N} \cap [0, K]$,
 K étant le plus petit entier tel que $-J_i + (K+1) \cdot T_i \geq \min(W_i(-J_i + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0))$.

Démonstration Nous obtenons l'expression de $W_i(t)$ ci-dessus en appliquant $sp_i = \emptyset$ à la propriété 5.3.2. Par ailleurs, les instants t à tester doivent être de la forme $t_i^0 + k \cdot T_i$ et vérifier : $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$. Puisque $sp_i = \emptyset$ et $G_{i,i}(t) = t$, seuls les instants de la forme $-J_i + k \cdot T_i$ doivent être considérés, avec $k \in \mathbb{N} \cap [0, K]$, K étant le plus petit entier tel que $-J_i + (K+1) \cdot T_i \geq \min(W_i(-J_i + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0))$. ■

Remarque 5.5.1 *Nous pouvons remarquer que, dans le cas particulier où tous les flux ont une priorité fixe différente (cas où l'ordonnancement FP/DP est équivalent à l'ordonnancement FP), nous retrouvons les résultats existants pour l'ordonnancement FP en contexte monoprocesseur (voir section 3.2.3, page 28).*

5.5.2 Les flux ont tous la même priorité fixe

Lorsque tous les flux partagent la même priorité fixe, le traitement FP n'intervient pas dans l'ordonnancement FP/DP. En effet, seul l'ordonnancement permettant de départager les paquets ayant la même priorité fixe est sollicité. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant un ordonnancement DP.

Si tous les flux ont la même priorité fixe, alors pour tout flux $\tau_i : hp_i = lp_i = \emptyset$. Par ailleurs, nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m . De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i ne pouvant pas générer de paquets avec une priorité dynamique supérieure ou égale à celle de m , mais susceptibles de générer un paquet pouvant gêner m à cause de la non-préemption.

En appliquant ces résultats à la propriété 5.3.2, nous obtenons la propriété suivante.

Propriété 5.5.2 En contexte monoprocesseur, lorsque les flux sont ordonnancés DP, si $U_{sp_i(t)} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :

$$R_i = \max_{t \in S''_i} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i(t)) + J_j}{T_j} \right\rfloor \right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max(0; \max_{j \in \overline{sp}_i(t)} \{C_j\} - 1)$$

et $S''_i = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} S''_i(t_i^0)$, où $S''_i(t_i^0)$ est l'ensemble des instants $t = t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; B_i(t_i^0))$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$.

Démonstration D'après la propriété 5.3.2, avec $hp_i = lp_i = \emptyset$. ■

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas d'un flux quelconque τ_i lorsque les flux sont ordonnancés FP/DP (section 5.3), nous avons considéré deux cas, à savoir (i) les flux ont tous une priorité fixe différente et (ii) les flux ont tous la même priorité fixe. Ces deux cas ont permis de déterminer le temps de réponse pire cas d'un flux quelconque τ_i lorsque les flux sont ordonnancés respectivement FP et DP. Il a alors été montré que nos résultats sont identiques à ceux établis dans l'état de l'art pour l'ordonnancement FP.

Nous nous intéressons maintenant à deux ordonnancements DP particuliers, à savoir FIFO (section 5.6) et EDF (section 5.7). Nous montrons alors que nos résultats améliorent ceux existants pour ces deux ordonnancements.

5.6 ORDONNANCEMENT FP/FIFO

Dans cette section, nous montrons comment calculer, en contexte monoprocesseur, le temps de réponse pire cas d'un flux quelconque τ_i lorsque les flux sont ordonnancés FP et les paquets de même priorité sont traités FIFO. Nous utilisons pour cela les résultats établis dans la section 5.3 et les appliquons à l'ordonnancement FP/FIFO. Nous retrouvons les résultats établis par ailleurs [67].

Nous considérons un flux quelconque τ_i . Soit m le paquet de τ_i généré à l'instant t . Nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m . De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité que τ_i susceptibles de générer un paquet de priorité dynamique strictement inférieure à celle de m mais pouvant gêner m à cause de la non-préemption.

Avec l'ordonnancement FIFO, la priorité dynamique d'un paquet est égale à son instant de génération. Par conséquent, un paquet d'un flux quelconque $\tau_j, j \in sp_i$, a une priorité dynamique supérieure ou égale à celle de m s'il est généré avant ou en même temps que m . Nous avons donc pour tout instant $t \geq -J_i$:

- $sp_i(t) = \{j \in sp_i, -J_j \leq t\}$.

En revanche, un paquet généré après t ne peut gêner m (même avec un ordonnancement non-préemptif) puisque les paquets sont activés dans l'ordre de leurs instants de génération. Par conséquent, l'exécution de m ne peut être retardée par un paquet d'un flux $\tau_j, j \in sp_i$, ayant une gigue d'activation telle que $-J_j > t$. Nous avons donc pour tout instant $t \geq -J_i$:

- $\overline{sp}_i(t) = \emptyset$.

D'autre part, pour tout flux $\tau_j, j \in sp_i(t)$, l'instant de génération au plus tard d'un paquet de τ_j pouvant gêner m est égal à l'instant de génération de m , soit :

- $\forall j \in sp_i(t), G_{j,i}(t) = t$.

Nous avons établi page 62 la propriété 5.3.2 donnant le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/DP. En appliquant les résultats ci-dessus à cette propriété, nous définissons le temps de réponse pire cas de τ_i lorsque les flux sont traités FP/FIFO.

5.6.1 Temps de réponse pire cas

La propriété suivante établit le temps de réponse pire cas du flux τ_i en contexte monoprocesseur lorsque l'ordonnancement utilisé est FP/FIFO, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs instants de génération.

Propriété 5.6.1 *En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/FIFO, si $U_{j \in hp_i \cup sp_i(t)} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$$R_i = \max_{t \in \mathcal{S}_i''} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{t + J_j}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max \left(0; \max_{j \in lp_i} \{C_j\} - 1\right)$$

et $\mathcal{S}_i'' = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}_i''(t_i^0)$, où $\mathcal{S}_i''(t_i^0)$ est l'ensemble des instants t de la forme $t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; B_i(t_i^0))$;
- $\exists j \in sp_i(t) \cup \{i\}$ tel que $t = -J_j + k_j \cdot T_j, k_j \in \mathbb{N}$.

Démonstration D'après la propriété 5.3.2, avec $sp_i(t) = \{j \in sp_i, -J_j \leq t\}$, $\overline{sp}_i(t) = \emptyset$ et pour tout flux τ_j tel que $j \in sp_i(t)$, $G_{j,i}(t) = t$. ■

5.6.2 Ordonnancement FIFO

Nous avons vu dans la section 5.5 que le traitement FP n'intervient pas dans l'ordonnancement FP/FIFO lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme FIFO.

Si tous les flux ont la même priorité fixe, alors pour tout flux τ_i : $hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 5.6.1, nous obtenons la propriété suivante.

Propriété 5.6.2 *En contexte monoprocesseur, lorsque les paquets sont ordonnancés FIFO, si $U_{j \neq i, -J_j \leq t} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :*

$$R_i = \max_{t \in S_i''} \left\{ \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor \right) \cdot C_j - t \right\}, \text{ avec :}$$

$S_i'' = \bigcup_{t_i^0 = -J_i}^{-J_i+T_i-1} S_i''(t_i^0)$, où $S_i''(t_i^0)$ est l'ensemble des instants t de la forme $t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; B_i(t_i^0))$;
- $\exists j$ tel que $-J_j \leq t$ et $t = -J_j + k_j \cdot T_j$, $k_j \in \mathbb{N}$.

Démonstration Puisque les flux ont tous la même priorité fixe, $hp_i = lp_i = \emptyset$. D'après la propriété 5.6.1, l'instant de démarrage au plus tard du paquet de τ_i généré à l'instant t est donc égal à :

$$W_i(t) = \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor \right) \cdot C_j + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor \cdot C_i.$$

Par ailleurs, $R_i = \max_{t \in S_i''} \{W_i(t) + C_i - t\}$. En remplaçant $W_i(t)$ par son expression, nous obtenons :

$$R_i = \max_{t \in S_i''} \left\{ \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t+J_j}{T_j} \right\rfloor \right) \cdot C_j - t \right\}. \quad \blacksquare$$

Remarque 5.6.1 *Il est important de souligner que dans le cas particulier où les flux partagent tous la même priorité fixe (cas où l'ordonnancement FP/FIFO est équivalent à l'ordonnancement FIFO), nous améliorons les résultats existants pour l'ordonnancement FIFO en contexte monoprocesseur. En effet, pour un flux quelconque τ_i , nous obtenons le même résultat (c'est-à-dire le même temps de réponse pire cas) mais en testant un plus petit nombre d'instants. Plus précisément, pour un t_i^0 donné :*

- les résultats existants² testent les instants $t = t_i^0 + k \cdot T_i$, $k \in \mathbb{N}$ tels que $t < \mathcal{B}_i(t_i^0)$;
- avec nos résultats, nous testons les instants $t = t_i^0 + k \cdot T_i$, $k \in \mathbb{N}$ tels que $t < \mathcal{B}_i(t_i^0)$, à l'exception :
 - des instants $t + \alpha \cdot T_i$, $\alpha \in \mathbb{N}^*$, si $W_i(t) + C_i \leq t + T_i$;
 - des instants t pour lesquels : $\nexists j$ tel que $-J_j \leq t$ et $t = -J_j + k_j \cdot T_j$, $k_j \in \mathbb{N}$.

5.6.3 Exemples

Nous présentons deux exemples montrant les améliorations importantes que nous pouvons obtenir en considérant un ordonnancement FP/FIFO plutôt qu'un ordonnancement FP en termes de (a) temps de réponse pire cas pour les flux partageant la même priorité fixe et (b) région d'ordonnançabilité.

5.6.3.a Temps de réponse pire cas

Soient cinq flux sporadiques, avec des giges d'activation nulles et dont les caractéristiques sont données dans le tableau 5.1. Trois flux partagent la plus basse priorité. La charge est égale à 100%.

TAB. 5.1: Améliorations obtenues sur les temps de réponse pire cas avec FP/FIFO

Flow	P_i	C_i	T_i	D_i	Résultats nouveaux		Résultats classiques	
					$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	30	24	28	32	36
τ_2	1	4	20	30	24	28	32	36
τ_3	1	4	20	30	24	28	32	36
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

Pour les flux de priorité 1, nous obtenons avec l'ordonnancement FP/FIFO un temps de réponse pire cas égal à 28, alors que le temps de réponse pire cas "classique", c'est-à-dire obtenu avec l'ordonnancement FP, est égal à 36. L'amélioration est donc supérieure à 22% dans cet exemple. De plus, toutes les échéances sont respectées avec l'ordonnancement FP/FIFO, ce qui n'est pas le cas avec l'approche classique pour laquelle aucune des échéances des flux de priorité 1 n'est satisfaite. Cet exemple montre qu'un ensemble de flux sporadiques faisable avec FP/FIFO peut ne pas être faisable avec FP.

5.6.3.b Région d'ordonnançabilité

Nous considérons maintenant trois flux dont les caractéristiques sont données dans le tableau 5.2.

La figure 5.2.a représente la région d'ordonnançabilité obtenue en ordonnant les trois flux considérés selon FP. Pour calculer cette région, nous testons pour chaque triplet (C_1, C_2, C_3) la condition suivante : l'échéance de chacun des flux est respectée, soit $\forall i \in [1, 3]$, $R_i \leq D_i$. Si cette condition est satisfaite, le triplet (C_1, C_2, C_3) appartient à la région d'ordonnançabilité.

²Les résultats existants sont présentés dans l'état de l'art. Ceux étudiés ici sont issus de la propriété 3.2.2, page 30.

TAB. 5.2: Caractéristiques des flux considérés

Flow	P_i	T_i	D_i
τ_1	2	50	100
τ_2	1	10	25
τ_3	1	10	35

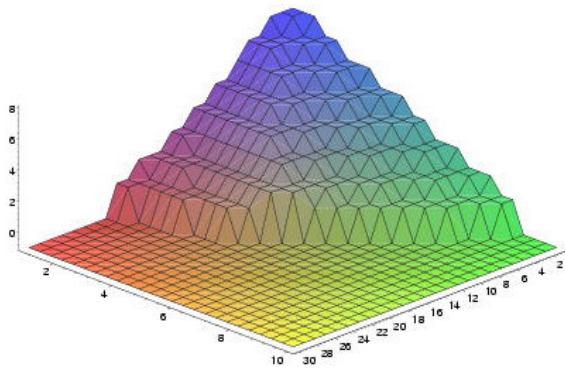
Les axes de la figure 5.2.a sont les suivants :

- axe de droite : C_1 ;
- axe de gauche : C_2 ;
- axe vertical : C_3 .

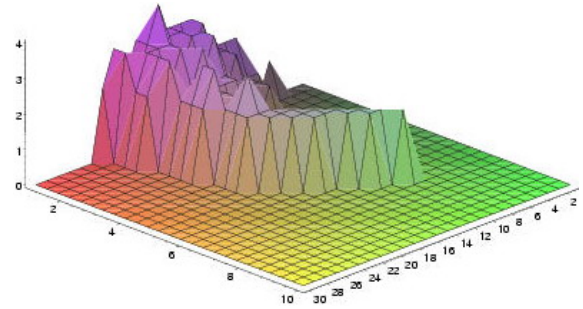
La figure 5.2.b montre l'amélioration qu'apporte l'ordonnancement FP/FIFO par rapport à l'ordonnancement FP. En effet, cette figure représente la différence entre les régions d'ordonnançabilité obtenues respectivement avec FP/FIFO et FP. Ainsi, les axes sont les suivants :

- axe de droite : C_1 ;
- axe de gauche : C_2 ;
- axe vertical : différence entre la valeur maximale de C_3 ordonnançable avec FP/FIFO et la valeur maximale de C_3 ordonnançable avec FP.

Remarque 5.6.2 Cette figure corrobore la propriété 5.4.2.



(a) Région d'ordonnançabilité avec FP



(b) Améliorations obtenues avec FP/FIFO

FIG. 5.2: Comparaison des régions d'ordonnançabilité obtenues avec FP et FP/FIFO

5.7 ORDONNANCEMENT FP/EDF

Dans cette section, nous montrons comment calculer, en contexte monoprocesseur, le temps de réponse pire cas d'un flux quelconque τ_i lorsque les flux sont ordonnancés FP et les paquets de même priorité sont traités EDF. Nous utilisons pour cela les résultats établis dans la section 5.3 et les appliquons à l'ordonnancement FP/EDF. Nous verrons ensuite le cas particulier où les flux ont tous la même priorité fixe, permettant d'obtenir des résultats pour l'ordonnancement EDF seul. Nous retrouvons les résultats établis par ailleurs [68].

Nous considérons un flux quelconque τ_i . Soit m le paquet de τ_i généré à l'instant t . Nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m . De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité que τ_i susceptibles de générer un paquet de priorité dynamique strictement inférieure à celle de m mais pouvant gêner m dû à la non-préemption.

Avec l'ordonnancement EDF, la priorité dynamique d'un paquet est égale à son échéance absolue, c'est-à-dire son instant de génération plus l'échéance relative de son flux. Ainsi, le paquet le plus prioritaire parmi ceux ayant la même priorité fixe est celui qui a la plus petite échéance absolue. Par conséquent, un paquet d'un flux quelconque τ_j , $j \in sp_i$, a une priorité dynamique supérieure ou égale à celle de m si son échéance absolue est plus petite que celle de m , soit pour tout instant $t \geq -J_i$:

- $sp_i(t) = \{j \in sp_i, P_j(-J_j) \geq P_i(t)\} = \{j \in sp_i, -J_j + D_j \leq t + D_i\}$;
- $\overline{sp}_i(t) = \{j \in sp_i, P_j(-J_j) < P_i(t)\} = \{j \in sp_i, -J_j + D_j > t + D_i\}$.

Un paquet d'un flux quelconque τ_j , $j \in sp_i(t)$, généré à l'instant t' est plus prioritaire que le paquet m si $t' + D_j \leq t + D_i$. Par conséquent, un paquet de τ_j doit être généré au plus tard en $t + D_i - D_j$ pour avoir une priorité généralisée supérieure ou égale à m , soit :

- $\forall j \in sp_i(t), G_{j,i}(t) = t + D_i - D_j$.

Nous avons établi page 62 la propriété 5.3.2 donnant le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/DP. En appliquant les résultats obtenus ci-dessus à cette propriété, nous définissons le temps de réponse pire cas du flux τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/EDF.

5.7.1 Temps de réponse pire cas

La propriété suivante établit le temps de réponse pire cas du flux τ_i en contexte monoprocesseur lorsque l'ordonnancement utilisé est FP/EDF, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs échéances absolues.

Propriété 5.7.1 En contexte monoprocesseur, lorsque les flux sont ordonnancés FP/EDF, si $U_{j \in hp_i \cup sp_i(t)} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :

$$R_i = \max_{t \in \mathcal{S}_i''} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i(t) + J_j}{T_j} \right\rfloor\right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t + D_i - D_j; W_i(t) + J_j)}{T_j} \right\rfloor\right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \delta_i(t),$$

$$\delta_i(t) = \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j\} - 1),$$

$$\mathcal{S}_i'' = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}_i''(t_i^0), \text{ où } \mathcal{S}_i''(t_i^0) \text{ est l'ensemble des instants } t \text{ de la forme } t_i^0 + k \cdot T_i \text{ tels que :}$$

- $t_i^0 \in [-J_i, -J_i + T_i[;$
- $k \in \mathbb{N} \cap [0, K],$
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0));$
- $\exists j \text{ et } l \text{ appartenant à } sp_i(t) \cup \{i\} \text{ tels que } t = -J_l + k_l \cdot T_l - D_i + D_j, k_l \in \mathbb{N}.$

Démonstration D'après la propriété 5.3.2, en appliquant : $sp_i(t) = \{j \in sp_i, -J_j + D_j \leq t + D_i\}$, $\overline{sp}_i(t) = \{j \in sp_i, -J_j + D_j > t + D_i\}$ et $\forall j \in sp_i(t), G_{j,i}(t) = t + D_i - D_j$. ■

5.7.2 Ordonnancement EDF

Nous avons vu dans la section 5.5 que le traitement FP n'intervient pas dans l'ordonnancement FP/EDF lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme EDF. Si tous les flux ont la même priorité fixe, alors pour tout flux τ_i : $hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 5.7.1, nous obtenons la propriété suivante.

Remarque 5.7.1 Il est important de souligner que dans le cas particulier où les flux partagent tous la même priorité fixe (cas où l'ordonnancement FP/EDF est équivalent à l'ordonnancement EDF), nous améliorons les résultats existants pour l'ordonnancement EDF en contexte monoprocesseur. En effet, pour un flux quelconque τ_i , nous obtenons le même résultat (c'est-à-dire le même temps de réponse pire cas) mais en testant un plus petit nombre d'instants. Plus précisément, pour un t_i^0 donné :

- les résultats existants³ testent les instants $t = t_i^0 + k \cdot T_i, k \in \mathbb{N}$ tels que :
 $t < \max(0; \max_{k \neq i} \{D_k - J_k\} - D_i) + \text{PPCM}_{j \in [1, n]} \{T_j\};$
- avec nos résultats, nous testons les instants $t = t_i^0 + k \cdot T_i, k \in \mathbb{N}$, tels que $t < \mathcal{B}_i(t_i^0)$, à l'exception :
 - des instants $t + \alpha \cdot T_i, \alpha \in \mathbb{N}^*, \text{ si } W_i(t) + C_i \leq t + T_i;$
 - des instants t pour lesquels : $\nexists (j, l) \in sp_i \cup \{i\} \text{ tels que } t = -J_l + k_l \cdot T_l + D_j - D_i, k_l \in \mathbb{N}.$

³Les résultats existants sont présentés dans l'état de l'art. Ceux étudiés ici sont issus de la propriété 3.2.3, page 31.

Propriété 5.7.2 En contexte monoprocesseur, lorsque les flux sont ordonnancés EDF, si $U_{j \in sp_i(t)} < 1$, alors le temps de réponse pire cas d'un flux quelconque τ_i est égal à :

$$R_i = \max_{t \in \mathcal{S}_i''} \{W_i(t) - t\} + C_i, \text{ avec :}$$

$$W_i(t) = \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t + D_i - D_j; W_i(t) + J_j)}{T_j} \right\rfloor \right) \cdot C_j + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \cdot C_i + \max \left(0; \max_{j \in \overline{sp}_i(t)} \{C_j\} - 1 \right),$$

$\mathcal{S}_i'' = \bigcup_{t_i^0 = -J_i}^{-J_i + T_i - 1} \mathcal{S}_i''(t_i^0)$, où $\mathcal{S}_i''(t_i^0)$ est l'ensemble des instants t de la forme $t_i^0 + k \cdot T_i$ tels que :

- $t_i^0 \in [-J_i, -J_i + T_i[$;
- $k \in \mathbb{N} \cap [0, K]$,
- K étant le plus petit entier tel que $t_i^0 + (K+1) \cdot T_i \geq \min(W_i(t_i^0 + K \cdot T_i) + C_i; \mathcal{B}_i(t_i^0))$;
- $\exists (j, l) \in sp_i(t) \cup \{i\}$ tels que $t = -J_l + k_l \cdot T_l - D_i + D_j$, $k_l \in \mathbb{N}$.

Démonstration D'après la propriété 5.7.1, avec $hp_i = lp_i = \emptyset$. ■

5.7.3 Exemples

Nous proposons ici deux exemples montrant les améliorations importantes que nous pouvons obtenir en considérant un ordonnancement FP/EDF plutôt qu'un ordonnancement FP sur :

- a) les temps de réponse pire cas des flux partageant la même priorité fixe ;
- b) la région d'ordonnancabilité.

5.7.3.a Temps de réponse pire cas

Soient cinq flux sporadiques, avec des giggers d'activation nulles et dont les caractéristiques sont données dans le tableau 5.3. Trois flux partagent la plus basse priorité. La charge est égale à 100%.

TAB. 5.3: Améliorations obtenues sur les temps de réponse pire cas avec FP/EDF

Flow	P_i	C_i	T_i	D_i	Résultats nouveaux		Résultats classiques	
					$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	26	20	24	32	36
τ_2	1	4	20	28	22	26	32	36
τ_3	1	4	20	30	24	28	32	36
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

Pour les flux de priorité 1, nous obtenons avec l'ordonnancement FP/EDF un temps de réponse pire cas égal à 24, alors que le temps de réponse pire cas "classique", c'est-à-dire obtenu avec l'ordonnancement FP, est égal à 36. L'amélioration est donc supérieure à 33% dans cet exemple. De plus, toutes les échéances sont respectées avec l'ordonnancement FP/EDF, ce qui n'est pas le cas avec l'approche classique pour laquelle aucune des échéances des flux de priorité 1 n'est satisfaite. Cet exemple montre qu'un ensemble de flux sporadiques faisable avec FP/EDF peut ne pas être faisable avec FP.

5.7.4 Région d'ordonnançabilité

Nous considérons maintenant trois flux dont les caractéristiques sont données dans le tableau 5.4.

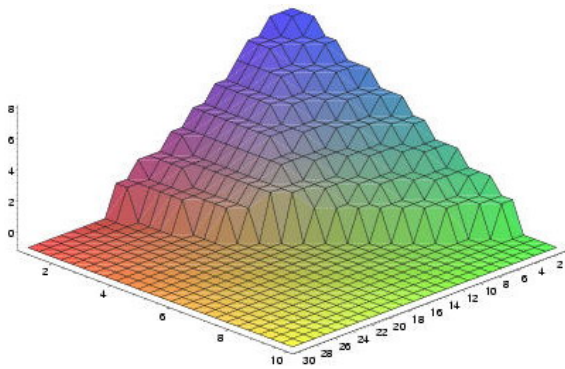
TAB. 5.4: Caractéristiques des flux considérés

Flow	P_i	T_i	D_i
τ_1	2	50	100
τ_2	1	10	25
τ_3	1	10	35

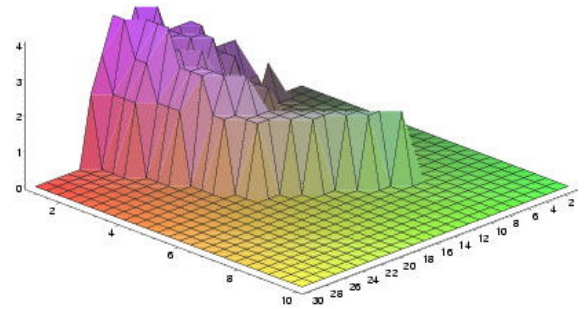
La figure 5.3.a représente la région d'ordonnançabilité obtenue en ordonnant les trois flux considérés selon FP. Pour calculer cette région, nous testons pour chaque triplet (C_1, C_2, C_3) la condition suivante : l'échéance de chacun des flux est respectée, soit $\forall i \in [1, 3], R_i \leq D_i$. Si cette condition est satisfaite, le triplet (C_1, C_2, C_3) appartient à la région d'ordonnançabilité. Sur cette figure, C_1 , C_2 et C_3 sont représentés respectivement sur les axes de droite, de gauche et vertical.

La figure 5.3.b montre l'amélioration qu'apporte l'ordonnancement FP/EDF par rapport à l'ordonnancement FP. En effet, cette figure représente la différence entre les régions d'ordonnançabilité obtenues respectivement avec FP/EDF et FP. Ainsi, C_1 et C_2 sont représentés respectivement sur l'axe de droite et l'axe de gauche. L'axe vertical représente la différence entre la valeur maximale de C_3 ordonnançable avec FP/EDF et la valeur maximale de C_3 ordonnançable avec FP.

Remarque 5.7.2 Cette figure corrobore la propriété 5.4.2.



(a) Région d'ordonnançabilité avec FP



(b) Améliorations obtenues avec FP/EDF

FIG. 5.3: Comparaison des régions d'ordonnançabilité obtenues avec FP et FP/EDF

5.8 DOMINANCE DE FP/EDF SUR FP/FIFO

5.8.1 Propriétés

Nous considérons un ensemble de flux sporadiques tel que les flux de même priorité fixe ont le même temps de traitement maximum. Cette hypothèse est réaliste car le temps de traitement d'un paquet dépend principalement de sa priorité fixe. Par exemple, dans une architecture DiffServ⁴, le temps de traitement d'un paquet dépend de sa classe de service, identifiée par son code DiffServ.

Propriété 5.8.1 *En contexte monoprocesseur, si un ensemble de flux tel que les flux de même priorité fixe ont le même temps de traitement maximum est faisable avec l'ordonnancement FP/FIFO, alors il est faisable avec l'ordonnancement FP/EDF.*

Démonstration Nous prouvons que si un ensemble de flux tel que les flux de même priorité fixe ont le même temps de traitement maximum n'est pas faisable avec FP/EDF, alors il n'est pas faisable avec FP/FIFO. Plus précisément, nous supposons que les flux sont ordonnancés FP/EDF et nous montrons que s'il existe un paquet m d'un flux τ_i qui dépasse son échéance, alors un paquet de même priorité fixe aurait dépassé son échéance si les flux avaient été ordonnancés FP/FIFO.

Soient m le paquet du flux τ_i généré à l'instant t et $W_i^{FP/EDF}(t)$ son instant de démarrage au plus tard lorsque les flux sont ordonnancés FP/EDF. Si m dépasse son échéance, nous avons alors : $W_i^{FP/EDF}(t) + C_i - t > D_i$, avec $W_i^{FP/EDF}(t) + C_i$ égal à :

$$\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{FP/EDF}(t) + J_j}{T_j} \right\rfloor \right) \cdot C_j + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(W_i^{FP/EDF}(t); t + D_i - D_j) + J_j}{T_j} \right\rfloor \right) \cdot C_j + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i + \delta_i(t).$$

Soit τ_k , $k \in sp_i \cup \{i\}$, le flux tel que $D_k = \min_{j \in sp_i \cup \{i\}} \{D_j\}$, c'est-à-dire le flux de priorité fixe P_i ayant l'échéance la plus petite parmi les flux τ_j vérifiant $t + D_i - D_j \geq -J_j$. Nous posons $\alpha = t + D_i - D_k$ et $\forall j \in sp_i \cup \{i\}$, $\Delta_j = D_j - D_k$. Ainsi, $W_i^{FP/EDF}(t) + C_i = W_i^{FP/EDF}(\alpha - \Delta_i) + C_i$.

Nous montrons que si les flux avaient été ordonnancés FP/FIFO, alors le paquet m' du flux τ_k , généré à l'instant α , aurait dépassé son échéance. Soit $W_k^{FP/FIFO}(\alpha)$ l'instant de démarrage au plus tard du paquet m' lorsque les flux sont ordonnancés FP/FIFO. Nous procédons en trois étapes :

- **Etape 1 :** Pour commencer, nous établissons le résultat suivant :

$$\begin{aligned} & \sum_{j \in sp_i(\alpha - \Delta_i)} \left(1 + \left\lfloor \frac{\alpha - \Delta_i + J_j}{T_j} \right\rfloor \right) \cdot C_j + \left(1 + \left\lfloor \frac{\alpha - \Delta_i + J_i}{T_i} \right\rfloor \right) \cdot C_i + \max \left(0; \max_{j \in lp_i \cup \overline{sp}_i(\alpha - \Delta_i)} \{C_j\} - 1 \right) \\ & \leq \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{\alpha + J_j}{T_j} \right\rfloor \right) \cdot C_j + \max(0; \max_{j \in lp_i} \{C_j\} - 1). \end{aligned}$$

⁴L'architecture de qualité de service DiffServ (*Differentiated Services*) est présentée en détails dans l'état de l'art, page 40.

Pour cela, nous montrons que tout flux compté dans la partie gauche de l'inéquation l'est également dans la partie droite. En effet, $\lfloor (\alpha - \Delta_j + J_j)/T_j \rfloor \leq \lfloor (\alpha + J_j)/T_j \rfloor$ et $sp_i(\alpha - \Delta_i) \subseteq sp_i$. De plus, si $lp_i \cup \overline{sp_i}(\alpha - \Delta_i) \neq \emptyset$, alors le flux $\tau_j, j \in lp_i \cup \overline{sp_i}(\alpha - \Delta_i)$, ayant le plus grand temps de traitement est compté dans la partie gauche de l'inéquation et :

- dans le premier terme de la partie droite si $j \in \overline{sp_i}(\alpha - \Delta_i)$, puisque $\overline{sp_i}(\alpha - \Delta_i) \subseteq sp_i$;
- dans le dernier terme de la partie droite si $j \in lp_i$.

- **Etape 2 :** Nous montrons maintenant que : $W_i^{FP/EDF}(\alpha - \Delta_i) + C_i \leq W_i^{FP/FIFO}(\alpha) + C_i$.
Pour cela, nous considérons les suites associées et procédons par récurrence. Au rang 0, nous avons $\mathcal{W}_i^{(0)FP/EDF}(\alpha - \Delta_i) + C_i$ égal à :

$$\begin{aligned} & \sum_{j \in hp_i \cup sp_i(\alpha - \Delta_i)} C_j + \left\lfloor \frac{\alpha - \Delta_i + J_i}{T_i} \right\rfloor \cdot C_i + \max \left(0 ; \max_{j \in lp_i \cup \overline{sp_i}(\alpha - \Delta_i)} \{C_j\} - 1 \right) + C_i \\ & \leq \sum_{j \in hp_i \cup sp_i} C_j + \left\lfloor \frac{\alpha + J_i}{T_i} \right\rfloor \cdot C_i + \max \left(0 ; \max_{j \in lp_i} \{C_j\} - 1 \right) + C_i \\ & \leq \mathcal{W}_i^{(0)FP/FIFO}(\alpha) + C_i. \end{aligned}$$

Nous supposons la récurrence vraie au rang k et montrons qu'elle est vraie au rang $k + 1$. En effet, $\mathcal{W}_i^{(k+1)FP/EDF^*}(\alpha - \Delta_i) + C_i$ est égal à :

$$\begin{aligned} & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(p)FP/EDF}(\alpha - \Delta_i) + J_i}{T_j} \right\rfloor \right) \cdot C_j + \sum_{j \in sp_i(\alpha - \Delta_i)} \left(1 + \left\lfloor \frac{\alpha - \Delta_i + J_i}{T_j} \right\rfloor \right) \cdot C_j \\ & + \left(1 + \left\lfloor \frac{\alpha - \Delta_i + J_i}{T_i} \right\rfloor \right) \cdot C_i + \max(0 ; \max_{j \in lp_i \cup \overline{sp_i}(\alpha - \Delta_i)} \{C_j\} - 1) \\ & \leq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(p)FP/FIFO}(\alpha) + J_i}{T_j} \right\rfloor \right) \cdot C_j + \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{\alpha + J_i}{T_j} \right\rfloor \right) \cdot C_j \\ & + \max(0 ; \max_{j \in lp_i} \{C_j\} - 1) \\ & \leq \mathcal{W}_i^{(p+1)FP/FIFO}(\alpha) + C_i. \end{aligned}$$

Les deux séries étant convergentes, leurs limites satisfont : $\mathcal{W}_i^{FP/EDF}(\alpha - \Delta_i) + C_i \leq \mathcal{W}_i^{FP/FIFO}(\alpha) + C_i$.
Par conséquent, $W_i^{FP/FIFO}(\alpha) + C_i > t + D_i = \alpha + D_k$.

- **Etape 3 :** Enfin, nous montrons par récurrence que si les flux de même priorité fixe ont le même temps de traitement, alors les temps de réponse pire cas des paquets des flux τ_i et τ_k générés à l'instant α sont les mêmes avec FP/FIFO. Notons que $\alpha \geq -J_i$ puisque, par construction, $D_i \geq D_k$ et $\alpha \geq -J_k$.

$$\forall \tau_i \in \tau, \forall \tau_k \in sp_i, W_i^{FP/FIFO}(\alpha) + C_i = W_k^{FP/FIFO}(\alpha) + C_k.$$

En effet, puisque $hp_i = hp_k, lp_i = lp_k$ et $sp_i \cup \{i\} = sp_k \cup \{k\}$, nous avons :

$$\mathcal{W}_i^{(0)FP/FIFO}(\alpha) + C_i = \sum_{j \in hp_i} C_j + \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{\alpha + J_i}{T_j} \right\rfloor \right) \cdot C_j + \max(0 ; \max_{j \in lp_i} \{C_j\} - 1) = \mathcal{W}_k^{(0)FP/FIFO}(\alpha) + C_k.$$

En supposant la récurrence vraie au rang p , nous montrons qu'elle est vraie au rang $p + 1$:

$$\begin{aligned}
 \mathcal{W}_i^{(p+1)FP/FIFO}(\alpha) + C_i &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{(p)FP/FIFO}(\alpha) + J_j}{T_j} \right\rfloor \right) \cdot C_j \\
 &\quad + \sum_{j \in sp_i \cup \{i\}} \left(1 + \left\lfloor \frac{\alpha + J_j}{T_j} \right\rfloor \right) \cdot C_j + \max(0; \max_{j \in lp_i} \{C_j\} - 1) \\
 &= \sum_{j \in hp_k} \left(1 + \left\lfloor \frac{\mathcal{W}_k^{(p)FP/FIFO}(\alpha) + C_k - C_i + J_j}{T_j} \right\rfloor \right) \cdot C_j \\
 &\quad + \sum_{j \in sp_k \cup \{k\}} \left(1 + \left\lfloor \frac{\alpha + J_j}{T_j} \right\rfloor \right) \cdot C_j + \max(0; \max_{j \in lp_k} \{C_j\} - 1) \\
 &= \mathcal{W}_k^{(p+1)FP/FIFO}(\alpha) + C_k.
 \end{aligned}$$

Nous avons donc : $\mathcal{W}_i^{FP/FIFO}(\alpha) + C_i = \mathcal{W}_k^{FP/FIFO}(\alpha) + C_k$.

D'après les étapes précédentes, nous obtenons :

$$\begin{aligned}
 \mathcal{W}_k^{FP/FIFO}(\alpha) + C_k &= \mathcal{W}_i^{FP/FIFO}(\alpha) + C_i && \text{(étape 3)} \\
 &\geq \mathcal{W}_i^{FP/EDF}(t) + C_i && \text{(étape 2)} \\
 &> t + D_i = \alpha + D_k && \text{(notre hypothèse).}
 \end{aligned}$$

Par conséquent, si le paquet m du flux τ_i dépasse son échéance lorsque les flux sont ordonnancés FP/EDF, alors le paquet m' du flux τ_k aurait dépassé son échéance si les flux avaient été ordonnancés FP/FIFO. ■

Propriété 5.8.2 *En contexte monoprocesseur, un ensemble de flux tel que les flux de même priorité fixe ont le même temps de traitement maximum faisable avec l'ordonnancement FP/EDF n'est pas nécessairement faisable avec l'ordonnancement FP/FIFO.*

Démonstration L'exemple suivant prouve cette propriété. En effet, les échéances des flux considérés sont satisfaites avec FP/EDF, mais pas avec FP/FIFO. ■

RÉCAPITULATIF : Dans cette section, nous avons établi les propriétés 5.8.1 et 5.8.2 qui montrent la dominance de FP/EDF sur FP/FIFO en contexte monoprocesseur, lorsque les flux de même priorité fixe ont le même temps de traitement maximum.

5.8.2 Exemples

Nous proposons ici deux exemples montrant les améliorations importantes que nous pouvons obtenir en considérant un ordonnancement FP/EDF plutôt qu'un ordonnancement FP sur :

- les temps de réponse pire cas des flux partageant la même priorité fixe ;
- la région d'ordonnançabilité.

5.8.2.a Temps de réponse pire cas

Soient cinq flux sporadiques, avec des giggers d'activation nulles et dont les caractéristiques sont données dans le tableau 5.5. Trois flux partagent la plus basse priorité. La charge est égale à 100%.

TAB. 5.5: Comparaison entre les ordonnancements FP/EDF et FP/FIFO

Flow	P_i	C_i	T_i	D_i	FP/EDF		FP/FIFO	
					$W_i(t)$	R_i	$W_i(t)$	R_i
τ_1	1	4	20	26	20	24	24	28
τ_2	1	4	20	28	22	26	24	28
τ_3	1	4	20	30	24	28	24	28
τ_4	2	4	20	15	11	15	11	15
τ_5	3	8	40	11	3	11	3	11

Nous pouvons remarquer que l'échéance du flux τ_1 est satisfaite lorsque les flux sont ordonnancés FP/EDF, mais est dépassée lorsque les flux sont ordonnancés FP/FIFO.

5.8.2.b Région d'ordonnançabilité

Nous considérons maintenant trois flux dont les caractéristiques sont données dans le tableau 5.6.

TAB. 5.6: Caractéristiques des flux considérés

Flow	P_i	T_i	D_i
τ_1	2	50	100
τ_2	1	10	25
τ_3	1	10	35

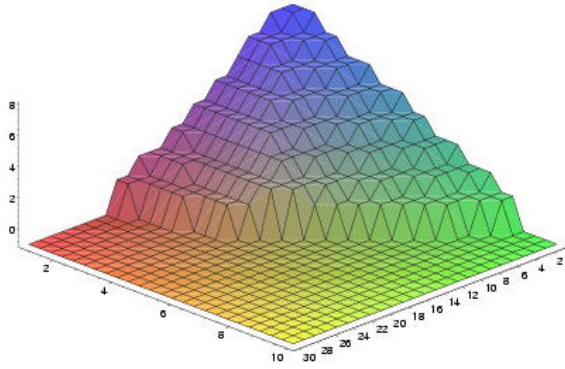
La figure 5.4.a représente la région d'ordonnançabilité obtenue en ordonnant les trois flux considérés selon FP/FIFO. Pour calculer cette région, nous testons pour chaque triplet (C_1, C_2, C_3) la condition suivante : l'échéance de chacun des flux est respectée, soit $\forall i \in [1, 3], R_i \leq D_i$. Si cette condition est satisfaite, le triplet (C_1, C_2, C_3) appartient à la région d'ordonnançabilité. Sur cette figure, les axes sont les suivants :

- axe de droite : C_1 ;
- axe de gauche : C_2 ;
- axe vertical : C_3 .

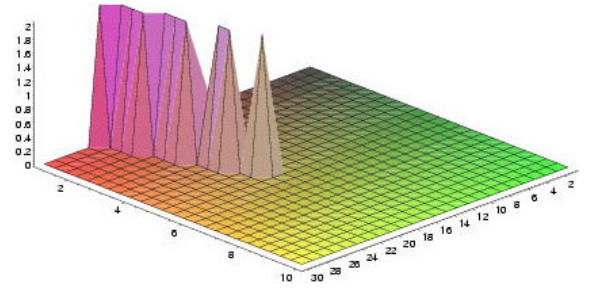
La figure 5.4.b montre l'amélioration qu'apporte l'ordonnancement FP/EDF par rapport à l'ordonnancement FP/FIFO. En effet, cette figure représente la différence entre les régions d'ordonnançabilité obtenues respectivement avec FP/EDF et FP/FIFO. Ainsi, les axes sont les suivants :

- axe de droite : C_1 ;
- axe de gauche : C_2 ;
- axe vertical : différence entre la valeur maximale de C_3 ordonnançable avec FP/EDF et la valeur maximale de C_3 ordonnançable avec FP/FIFO.

Remarque 5.8.1 Cette figure corrobore la propriété 5.8.2.



(a) Région d'ordonnabilité avec FP/FIFO



(b) Améliorations obtenues avec FP/EDF

FIG. 5.4: Comparaison des régions d'ordonnabilité obtenues avec FP/FIFO et FP/EDF

5.9 CONCLUSION

Nous avons déterminé dans ce chapitre le temps de réponse pire cas d'un flux sporadique quelconque τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/DP. Pour cela, nous avons procédé en plusieurs étapes.

- Tout d'abord, nous nous sommes intéressés à un paquet quelconque m du flux τ_i , généré à l'instant t . Nous avons alors étudié la période active dans laquelle ce paquet était traité, ce qui nous a permis de déterminer $W_i(t)$, l'instant de démarrage au plus tard de m (propriété 5.3.1, page 57).
- Comme l'expression de cet instant est une formule récursive, nous avons ensuite prouvé l'existence d'une solution lorsque $U_{j \in hp_i \cup sp_i(t)} < 1$ (condition 5.3.1, page 58), où $U_{j \in hp_i \cup sp_i(t)}$ désigne le taux d'utilisation du processeur pour les flux τ_j , $j \neq i$, (i) ayant une priorité fixe strictement supérieure à celle de τ_i ou (ii) de même priorité fixe et pouvant générer un paquet de priorité dynamique supérieure ou égale à celle de m . Cette condition n'est pas contraignante puisqu'une condition nécessaire pour la faisabilité d'un ensemble de flux est que le facteur d'utilisation du processeur, noté U , soit inférieur ou égal à 1, or $U_{j \in hp_i \cup sp_i(t)} < U$.
- Le temps de réponse pire cas du flux τ_i , noté R_i , est égal au maximum des temps de réponse pire cas de ses paquets, soit : $R_i = \max_t \{W_i(t) - t\} + C_i$, avec C_i le temps de traitement maximum d'un paquet de τ_i . Afin de limiter le nombre d'instants t à tester pour le calcul du temps de réponse pire cas de τ_i , nous avons établi trois lemmes (pages 59- 60).

Nous sommes ainsi arrivés à l'expression du temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/DP (propriété 5.3.2, page 62).

Nous avons alors souligné l'intérêt de combiner priorités fixes et priorités dynamiques en montrant que tout ensemble de flux faisable avec l'ordonnancement FP est faisable avec l'ordonnancement FP/DP (propriété 5.4.1, page 63) mais que tout ensemble de flux faisable avec l'ordonnancement FP/DP n'est pas nécessairement faisable avec l'ordonnancement FP (propriété 5.4.2, page 63).

Ensuite, nous avons appliqué ces résultats à deux ordonnancements DP particuliers, à savoir : FIFO et EDF. Cela a permis d'établir le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur lorsque les flux sont ordonnancés FP/FIFO (propriété 5.6.1, page 67) ou FP/EDF (propriété 5.7.1, page 72).

Puis, afin d'isoler l'ordonnancement FP de l'ordonnancement DP, nous avons considéré deux cas :

- 1) Les flux ont tous la même priorité fixe. L'ordonnancement FP/DP est alors équivalent à un ordonnancement DP seul. Nous avons ainsi établi des résultats pour FIFO et EDF en contexte monoprocesseur et avons montré que nous améliorons les résultats connus pour ces deux ordonnancements en testant un nombre plus limité d'instants pour obtenir le temps de réponse pire cas d'un flux.
- 2) Les flux ont tous une priorité fixe différente. L'ordonnancement FP/DP est alors équivalent à un ordonnancement FP seul. Les résultats que nous obtenons sont les mêmes, dans ce cas, que ceux existants.

Différents exemples ont été présentés pour illustrer les améliorations qu'apportent les ordonnancements FP/FIFO et FP/EDF par rapport à l'ordonnancement FP, en termes de temps de réponse pire cas et de région d'ordonnançabilité.

Enfin, la dominance de FP/EDF sur FP/FIFO en contexte monoprocesseur a été démontrée pour un ensemble de flux tels que les flux de même priorité fixe ont le même temps de traitement maximum.

CHAPITRE 6

Ligne de diffusion

6.1	Introduction	82
6.2	Démarche suivie	82
6.3	Analyse pire cas	83
6.3.1	Approche par trajectoire	83
6.3.2	Evaluation des différents délais	84
6.3.3	Instant de démarrage au plus tard sur le dernier nœud visité	88
6.3.4	Instants à tester	90
6.3.5	Temps de réponse pire cas de bout-en-bout	93
6.4	Cas particuliers	94
6.4.1	La ligne de diffusion est réduite à un seul nœud	94
6.4.2	Les flux ont tous une priorité fixe différente	94
6.4.3	Les flux ont tous la même priorité fixe	95
6.5	Ordonnancement FP/FIFO*	96
6.5.1	Temps de réponse pire cas de bout-en-bout	97
6.5.2	Ordonnancement FIFO*	98
6.5.3	Exemples	99
6.6	Ordonnancement FP/EDF*	101
6.6.1	Temps de réponse pire cas	102
6.6.2	Ordonnancement EDF*	102
6.6.3	Exemples	104
6.7	Conclusion	105

6.1 INTRODUCTION

Dans le chapitre précédent, nous avons présenté de nouveaux résultats en contexte monoprocesseur pour un ordonnancement de type FP/DP non-oisif et non-préemptif de flux sporadiques dont les instants d'arrivée ne sont pas connus à priori. Nous avons appliqué nos résultats à deux ordonnancements FP/DP particuliers, à savoir FP/FIFO et FP/EDF, puis nous avons montré que dans le cas particulier où les flux partagent tous la même priorité fixe, nous améliorons les résultats connus en monoprocesseur pour les ordonnancements FIFO et EDF. Enfin, la dominance de FP/EDF sur FP/FIFO a été démontrée sous certaines hypothèses.

Dans ce chapitre, nous étendons les résultats obtenus en contexte monoprocesseur au cas où les flux suivent tous une même ligne de diffusion, c'est-à-dire une même séquence de nœuds. De plus, nous considérons que les flux sont traités dans chacun des nœuds visités suivant un ordonnancement de type FP/DP*. En effet, nous rappelons que tout paquet entrant dans le réseau se voit assigner une priorité généralisée, c'est-à-dire une priorité fixe (celle du flux auquel il appartient) et une priorité dynamique. Ainsi, dans chacun des nœuds du réseau, les paquets sont ordonnancés selon leurs priorités généralisées. Plus précisément, les paquets sont traités dans l'ordre de leurs priorités fixes (ordonnancement FP) puis ceux de même priorité fixe dans l'ordre de leurs priorités dynamiques (ordonnancement DP*). Ainsi, dans ce chapitre :

- nous déterminons le temps de réponse pire cas de bout-en-bout d'un flux sporadique quelconque lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités ;
- nous étudions deux cas particuliers concernant les priorités fixes des flux et obtenons des résultats pour l'ordonnancement FP seul, puis pour un ordonnancement DP* seul dans le cas d'une ligne de diffusion ;
- nous appliquons les résultats obtenus à deux ordonnancements DP* particuliers, à savoir FIFO* et EDF*, puis présentons des exemples numériques ;
- nous comparons nos résultats avec ceux obtenus par l'approche holistique, puis avec ceux fournis par une approche exhaustive.

Dans le chapitre suivant, nous généraliserons nos résultats au cas général, c'est-à-dire au cas où les flux suivent des séquences de nœuds différentes.

6.2 DÉMARCHE SUIVIE

Afin de déterminer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , $i \in [1, n]$, lorsque les flux suivent tous une même ligne de diffusion et les paquets sont ordonnancés suivant un algorithme du type FP/DP* dans chacun des nœuds visités, nous adoptons l'approche par trajectoire. Cette approche consiste à examiner l'ordonnancement produit sur l'ensemble des nœuds visités par un flux [69]. Ainsi, les seuls scénarios considérés sont des scénarios pouvant se produire. Le modèle fluide, par exemple, relève de l'approche par trajectoire. Cette approche produit de meilleurs résultats que l'approche holistique puisqu'aucun scénario irréalisable n'est considéré, mais elle est plus complexe à utiliser.

Plus précisément, nous nous intéressons à un paquet quelconque m de τ_i généré à l'instant t et nous identifions sur chacun des nœuds visités (en commençant par le dernier puis en remontant de nœud en nœud) la période active et les paquets affectant son temps de réponse de bout-en-bout. Nous procédons ainsi jusqu'à la détermination, sur le nœud source, du premier paquet ayant affecté le temps de réponse de bout-en-bout du paquet m . Le temps passé par m dans le réseau est alors égal à l'instant d'activation du premier paquet gênant m dans le nœud source, plus la somme des temps de traitement des paquets identifiés dans les périodes actives considérées, moins l'instant de génération du paquet m .

6.3 ANALYSE PIRE CAS

Nous déterminons dans cette section le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux (i) suivent tous une même ligne de diffusion \mathcal{L} composée de q nœuds, numérotés de 1 à q et (ii) sont ordonnancés FP/DP* dans chacun des nœuds visités. Pour cela, nous adoptons l'approche par trajectoire décrite dans la section 6.2, qui consiste à étudier les périodes actives qui affectent le temps de réponse de bout-en-bout du paquet m . Cela nous conduira à la détermination de l'instant de démarrage au plus tard de m sur le nœud q , puis au temps de réponse pire cas du flux τ_i . Par ailleurs, nous rappelons que les notations et définitions utilisées dans ce chapitre sont définies page 50.

Remarque 6.3.1 Dans cette analyse, tout paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, généré à un instant $t' < 0$ subit une gigue d'activation et est activé à l'instant 0.

6.3.1 Approche par trajectoire

Pour calculer l'instant de démarrage au plus tard du paquet m sur le nœud q , nous identifions les périodes actives de niveau $PG_i(t)$ qui affectent son temps de réponse de bout-en-bout sur chacun des nœuds visités. Pour cela, nous considérons la période active de niveau $PG_i(t)$ dans laquelle m est traité sur le nœud q . Soient bp^q cette période active et $f(q)$ le premier paquet traité dans bp^q de priorité supérieure ou égale à $PG_i(t)$.

Le paquet $f(q)$ a été traité sur le nœud $q - 1$ dans une période active de niveau au moins égal à $PG_i(t)$. Soient bp^{q-1} cette période active et $f(q-1)$ le premier paquet traité dans bp^{q-1} avec une priorité supérieure ou égale à $PG_i(t)$. Nous procédons ainsi jusqu'à la détermination sur le nœud 1 de la période active bp^1 dans laquelle $f(2)$ est traité et où $f(1)$ est le premier paquet de priorité supérieure ou égale à $PG_i(t)$. La figure 6.1 illustre cette décomposition.

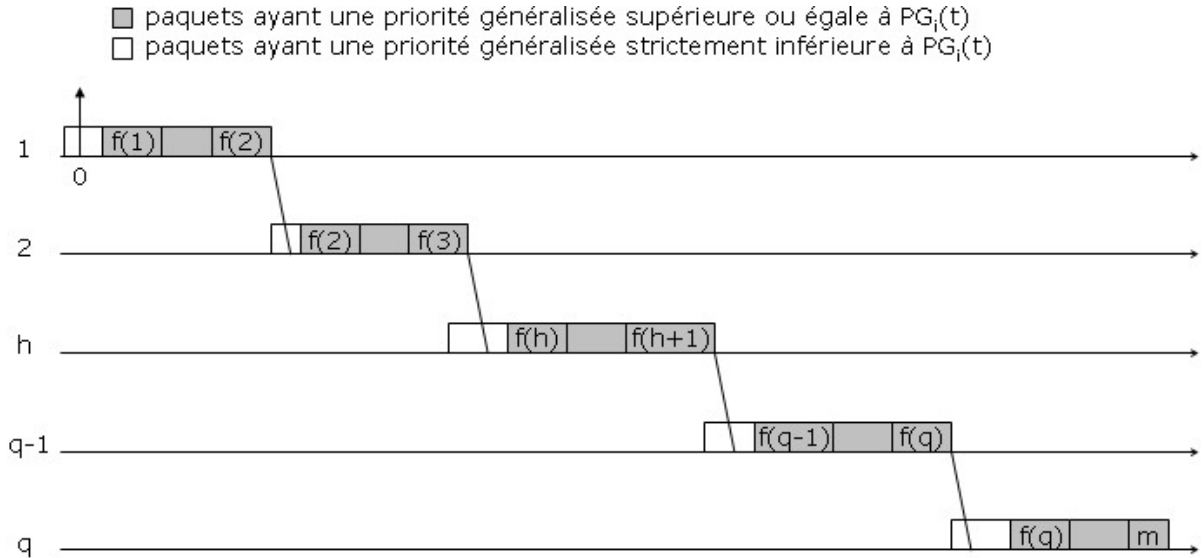


FIG. 6.1: Détermination du temps de réponse de bout-en-bout avec l'approche par trajectoire

Pour des raisons de clarté, nous numérotons successivement les paquets des périodes actives considérées. Ainsi, nous notons $m' - 1$ (respectivement $m' + 1$) le paquet traité avant (respectivement après) m' . De plus, nous considérons l'instant d'activation du paquet $f(1)$ comme origine des temps et posons $f(q + 1) = m$.

Remarque 6.3.2 *Un paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, généré avant $-J_j$ n'appartient pas aux périodes actives de niveau $PG_i(t)$ considérées.*

Il est important de souligner que notre décomposition vérifie la propriété suivante.

Propriété 6.3.1 *Seuls les paquets $f(h+1)$, $h \in [1, q]$, sont considérés deux fois dans l'ensemble $\bigcup_{h=1}^q [f(h), f(h+1)]$.*

Démonstration Aucun paquet m' traité entre $f(h)$ et $f(h+1)$ sur le nœud h , avec $m' \neq f(h)$ et $m' \neq f(h+1)$, ne peut être traité entre $f(h')$ et $f(h'+1)$ sur le nœud h' , avec $h' \in [1, q]$ et $h' \neq h$. En effet, par récurrence, si m' est traité avant $f(h+1)$ sur le nœud h , alors m' arrive avant $f(h+1)$ sur le nœud $h+1$, les liens étant supposés FIFO. Par construction, l'instant d'arrivée de $f(h+1)$ sur le nœud $h+1$ est un instant oisif de niveau $PG_i(t)$. Par conséquent, tout paquet de priorité supérieure ou égale à $PG_i(t)$ est traité avant $f(h+1)$ s'il arrive avant sur ce nœud. Le paquet m' ne peut donc pas appartenir à bp^{h+1} . ■

A partir de cette décomposition, nous pouvons déterminer $W_i^q(t)$, l'instant de démarrage au plus tard du paquet m sur le nœud q . En effet, $W_i^q(t)$ est égal :

- au temps de traitement sur le nœud 1 des paquets $f(1)$ à $f(2) + L_{max}$
- + le temps de traitement sur le nœud 2 des paquets $f(2)$ à $f(3) + L_{max}$
- + ...
- + le temps de traitement sur le nœud q des paquets $f(q)$ à $(m-1)$
- + $\delta_i^{1,q}(t)$, le délai maximum subi par m dans les nœuds 1 à q à cause de l'effet non-préemptif direct¹.

Puisque, par convention, $f(q+1) = m$, nous obtenons :

$$W_i^q(t) = \sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}. \quad (6.1)$$

Dans la sous-section suivante, nous évaluons les termes composant l'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q , à savoir : $\delta_i^{1,q}(t)$ et $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$.

6.3.2 Evaluation des différents délais

Nous nous intéressons à l'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q donnée dans l'équation 6.1. Plus précisément, nous étudions les deux termes suivants afin de les maximiser : $\delta_i^{1,q}(t)$ et $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$.

¹L'effet non-préemptif direct est défini dans la sous-section 3.2.1, page 27.

- Le premier terme représente le délai maximum que m peut subir tout au long de la ligne de diffusion à cause de l'effet non-préemptif direct et est étudié dans le paragraphe 6.3.2.a.
- Le second terme représente le délai maximum que m peut subir tout au long de la ligne de diffusion à cause de paquets plus prioritaires et est étudié dans le paragraphe 6.3.2.b.

6.3.2.a Délai dû à l'effet non-préemptif direct

Les deux lemmes suivants permettent de borner le délai maximum que le paquet m peut subir tout au long de la ligne de diffusion à cause de l'effet non-préemptif direct.

Lemme 6.3.1 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/DP* dans chacun des nœuds visités, le délai maximum subi sur les nœuds 1 à q par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est égal à : $\delta_i^{1,q}(t) \leq \sum_{h=1}^q \max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} - 1)$, avec $\max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} = 0$ si $lp_i \cup \overline{sp}_i(t) = \emptyset$.*

Démonstration Par définition, aucun des paquets appartenant à $lp_i \cup \overline{sp}_i(t)$ ne peut être traité dans une période active de niveau $PG_i(t)$, à l'exception du premier paquet de la période active (voir le lemme 5.3.1). Par conséquent, le délai maximum subi par le paquet du flux τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est au plus égal à $\max(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} - 1)$ sur chaque nœud h visité. ■

Il est possible d'affiner cette borne dans le cas particulier où les délais réseau sont constants et, sur un nœud quelconque $h \in [1, q]$, les paquets ont le même temps de traitement, à savoir C^h . Ce cas se produit, entre autre, lorsque les paquets ont la même taille tout au long d'un chemin, égale au MTU (*Maximum Transmission Unit*) minimum du chemin. Cela permet d'éviter la segmentation et le réassemblage des paquets dans les nœuds intermédiaires, allégeant ainsi leur traitement. Dans IPv6, par exemple, le Path MTU Discovery détermine ce MTU minimum. Dans le cas d'une simple ligne de diffusion, cela conduit à considérer un temps de traitement identique pour tous les paquets dans chacun des nœuds visités. De plus, les liens point-à-point sont un exemple de liens fournissant des délais réseau constants.

Lemme 6.3.2 *Lorsque (i) les flux suivent tous la même ligne de diffusion composée de q nœuds, (ii) les paquets sont ordonnancés FP/DP* dans chacun des nœuds visités, (iii) sur chaque nœud $h \in [1, q]$, le temps de traitement de tout flux est égal à C^h et (iv) les délais réseau sont constants, alors le délai maximum subi par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct en visitant les nœuds 1 à h , $h \in [1, q]$, satisfait :*

$$\begin{cases} \delta_i^{1,h}(t) = \delta_i^{1,h-1}(t) & \text{si } C^h \leq \max_{k=1..h-1} \{C^k\} \\ \delta_i^{1,h}(t) \leq \delta_i^{1,h-1}(t) + C^h - 1 & \text{sinon,} \end{cases} \quad \text{avec } \delta_i^{1,1}(t) = C^1 - 1.$$

Démonstration Puisque les délais réseau sont constants, le délai d'interarrivée de deux paquets successifs sur un nœud quelconque $h \in [1, q]$ est égal au délai d'interdépart de ces deux paquets sur le nœud $h - 1$. De plus, soit s le nœud le plus lent parmi les nœuds 1 à $h - 1$. Puisque, sur chaque nœud, le temps de traitement est constant, le délai d'interdépart sur le nœud s est au moins égal à C^s . Le délai d'interarrivée sur le nœud h est donc au moins égal à C^s . En effet, $\forall h' \in [s + 1, h - 1]$, $C^{h'} \leq C^s$. Par conséquent, pour tout flux τ_i :

- si $C^h \leq C^s$, aucun délai supplémentaire dû à l'effet non-préemptif direct n'est subi par τ_i ;
- si $C^h > C^s$, τ_i subit un délai supplémentaire sur le nœud h dû à l'effet non-préemptif direct. ■

6.3.2.b Délai dû aux paquets plus prioritaires

Nous procédons en trois étapes pour évaluer le délai maximum que le paquet m peut subir tout au long de la ligne de diffusion à cause de paquets ayant une priorité généralisée supérieure ou égale à $PG_i(t)$.

- Le lemme 6.3.3 permet d'évaluer ce délai.
- Le lemme 6.3.4 détermine les intervalles de temps dans lesquels les paquets plus prioritaires doivent être générés pour gêner m .
- Le lemme 6.3.5 maximise le délai obtenu dans le lemme 6.3.3 en appliquant le lemme 6.3.4.

Lemme 6.3.3 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnés FP/DP* dans chacun des nœuds visités, le délai maximum subi sur les nœuds 1 à q par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à des paquets plus prioritaires est égal à :*

$$\sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q \leq \sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}.$$

Démonstration Le délai maximum que m peut subir tout au long de la ligne de diffusion à cause de paquets plus prioritaires est égal à : $\sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q$. En distinguant les nœuds visités par les flux avant le nœud $slow$ - le nœud le plus lent de la ligne de diffusion \mathcal{L}_i - et ceux visités après, nous obtenons :

$$\sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) = \underbrace{\sum_{h=1}^{slow-1} \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right)}_{\text{nœuds visités avant slow}} + \underbrace{\sum_{g=f(slow)}^{f(slow+1)} C_{\tau(g)}^{slow}}_{\text{nœud slow}} + \underbrace{\sum_{h=slow+1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right)}_{\text{nœuds visités après slow}}.$$

D'après la propriété 6.3.1, seuls les paquets $f(h+1)$, $h \in [1, q]$, sont considérés deux fois parmi les paquets $\bigcup_{h=1}^q [f(h), f(h+1)]$. Par conséquent, en isolant les paquets $f(h+1)$ sur les nœuds $h < slow$ et $f(h)$ sur les nœuds $h > slow$, nous obtenons :

$$\begin{aligned} \sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) &= \sum_{h=1}^{slow-1} \left(\sum_{g=f(h)}^{f(h+1)-1} C_{\tau(g)}^h + C_{\tau(f(h+1))}^h \right) + \sum_{g=f(slow)}^{f(slow+1)} C_{\tau(g)}^{slow} + \sum_{h=slow+1}^q \left(\sum_{g=f(h)+1}^{f(h+1)} C_{\tau(g)}^h + C_{\tau(f(h))}^h \right) \\ &= \sum_{h=1}^{slow-1} \left(\sum_{g=f(h)}^{f(h+1)-1} C_{\tau(g)}^h \right) + \sum_{h=slow+1}^q \left(\sum_{g=f(h)+1}^{f(h+1)} C_{\tau(g)}^h \right) + \sum_{h=1}^{slow-1} C_{\tau(f(h+1))}^h + \sum_{h=slow+1}^q C_{\tau(f(h))}^h. \end{aligned}$$

Puisque $slow$ est le nœud le plus lent, nous avons :

$$\sum_{h=1}^{slow-1} \left(\sum_{g=f(h)}^{f(h+1)-1} C_{\tau(g)}^h \right) + \sum_{h=slow+1}^q \left(\sum_{g=f(h)+1}^{f(h+1)} C_{\tau(g)}^h \right) \leq \sum_{g=f(1)}^m C_{\tau(g)}^{slow}.$$

De plus, pour tout nœud $h \in [1, q]$ et pour tout paquet m' de priorité généralisée supérieure ou égale à $PG_i(t)$ visitant h , le temps de traitement de m' dans le nœud h est inférieur ou égal à $\max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}$, soit :

$$\sum_{h=1}^{slow-1} C_{\tau(f(h+1))}^h + \sum_{h=slow+1}^q C_{\tau(f(h))}^h \leq \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}.$$

Par conséquent, $\sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) \leq \sum_{g=f(1)}^m C_{\tau(g)}^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}$. ■

Lemme 6.3.4 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnés FP/DP* dans chacun des nœuds visités, alors hormis l'effet non-préemptif direct, le traitement du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est retardé par :*

- les paquets des flux τ_j , $j \in hp_i$, générés dans l'intervalle $[-J_j, W_i^q(t) - Smin_j^q]$;
- les paquets des flux τ_j , $j \in sp_i(t)$, générés dans l'intervalle $[-J_j, \min(G_{j,i}(t), W_i^q(t) - Smin_j^q)]$;
- les paquets du flux τ_i générés dans l'intervalle $[-J_i, t]$.

Démonstration Un paquet m' d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t)$, peut affecter le temps de réponse de bout-en-bout du paquet m s'il existe un nœud $h \in [1, q]$ tel que $m' \in [f(h), f(h+1)]$. D'après la propriété 6.3.1, si m' arrive avant $f(1)$ sur le nœud 1 ou après $W_i^q(t)$ sur le nœud q , alors m' n'appartient pas aux paquets considérés. Par conséquent, aucun paquet de τ_j ne peut gêner m s'il est généré avant $-J_j$, le paquet $f(1)$ étant activé à l'instant 0, ou après $W_i^q(t) - Smin_j^q$, avec $Smin_j^q$ le temps minimum mis par un paquet de τ_j pour arriver dans le nœud q , soit : $Smin_j^q = \sum_{h=1}^{q-1} C_j^h + (q-1) \cdot Lmin$.

De plus, si un paquet d'un flux quelconque τ_j , $j \in sp_i(t)$, est généré après $G_{j,i}(t)$, alors sa priorité généralisée est strictement plus petite que $PG_i(t)$. Si tel est le cas, il ne peut pas être traité dans les périodes actives de niveau $PG_i(t)$ considérées. Par conséquent, le temps de réponse de bout-en-bout de m n'est pas affecté par les paquets de τ_j générés après $G_{j,i}(t)$ ou $W_i^q(t) - Smin_j^q$.

Enfin, aucun paquet du flux τ_i ne peut gêner m s'il est généré après t . ■

Grâce au lemme précédent, nous pouvons maximiser le terme $\sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow}$ du lemme 6.3.3 et ainsi évaluer plus précisément le délai maximum que peut subir m à cause de paquets plus prioritaires.

Lemme 6.3.5 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnés FP/DP* dans chacun des nœuds visités, le retard maximum subi par le paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ dû à des paquets de priorités généralisées supérieures ou égales est égal à :

$$\begin{aligned} & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^q(t) - Smin_j^q + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^q(t) - Smin_j^q) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \\ & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}. \end{aligned}$$

Démonstration Le terme $\sum_{g=f(1)}^m C_{\tau(g)}^{slow}$ est borné par la quantité maximale de travail² générée par les paquets définis dans le lemme 6.3.4 sur le nœud $slow$, soit :

- pour les flux τ_j , $j \in hp_i$: $\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^q(t) - Smin_j^q + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow}$;
- pour les flux τ_j , $j \in sp_i(t)$: $\sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^q(t) - Smin_j^q) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow}$;
- pour le flux τ_i : $\left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow}$. ■

²Définie dans [66], la quantité maximale de travail générée par un flux quelconque τ_j sur un nœud h dans l'intervalle $[t_1, t_2]$ est le temps nécessaire au processeur du nœud h pour traiter les paquets de τ_j générés dans cet intervalle, soit : $(1 + \lfloor (t_2 - t_1)/T_j \rfloor) \cdot C_j^h$.

RÉCAPITULATIF : A partir de la décomposition présentée au début de cette section (voir figure 6.1), nous sommes arrivés à l'équation 6.1 donnant l'instant de démarrage au plus tard du paquet m sur le nœud q , soit : $W_i^q(t) = \sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$.

- Les lemmes 6.3.1 et 6.3.2 établis précédemment permettent de borner le terme $\delta_i^{1,q}(t)$, c'est-à-dire le délai maximum que le paquet m peut subir tout au long de la ligne de diffusion à cause de l'effet non-préemptif direct.
- Nous avons alors étudié le terme $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$, représentant le retard maximum introduit par des paquets de priorités généralisées supérieures ou égales à celle de m . Après avoir déterminé les paquets à considérer dans l'ensemble $\bigcup_{h=1}^q [f(h), f(h+1)]$, nous avons établi le lemme 6.3.5, maximisant ce terme.

6.3.3 Instant de démarrage au plus tard sur le dernier nœud visité

A partir de l'analyse réalisée dans les deux paragraphes précédents, nous pouvons établir l'instant de démarrage au plus tard du paquet m sur le nœud q .

Propriété 6.3.2 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/DP* dans chacun des nœuds visités, l'instant de démarrage au plus tard sur le nœud q du paquet d'un flux quelconque τ_i généré à l'instant $t \geq -J_i$ est égal à :

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^q(t) - \text{Min}_j^q + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} + \sum_{j \in sp_i} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^q(t) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}.$$

Démonstration D'après l'équation 6.1, $W_i^q(t) = \sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$. Une borne sur $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h)$ est donnée dans le lemme 6.3.3. En appliquant cette borne à l'expression de $W_i^q(t)$, il vient :

$$W_i^q(t) = \sum_{g=f(1)}^m C_{\tau(g)}^{slow} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}.$$

Le lemme 6.3.5 maximise le terme $\sum_{g=f(1)}^m C_{\tau(g)}^{slow} - C_i^q$. En remplaçant ce terme par sa borne dans l'expression de $W_i^q(t)$, nous obtenons le lemme. ■

L'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q est donc une formule récursive. Nous montrons la convergence de la suite $W_i^q(t)$ définie ci-dessous afin de prouver l'existence de sa limite, à savoir $W_i^q(t)$.

$$\left\{ \begin{array}{l} \mathcal{W}_i^{q(0)}(t) = \sum_{j \in hp_i \cup sp_i(t)} C_j^{slow} + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) \\ \mathcal{W}_i^{q(k+1)}(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - S_{min_j^q} + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} \\ \quad + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - S_{min_j^q}) + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} \\ \quad + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t), \end{array} \right.$$

avec $\mathcal{A}_i(t) = \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \left\{ C_j^h \right\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$.

Une condition suffisante pour que l'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q ait une solution est que le facteur d'utilisation du processeur sur le nœud $slow$ pour les flux τ_j , $j \in hp_i \cup sp_i(t)$, soit inférieur à 1. Cette condition n'est pas contraignante puisqu'une condition nécessaire évidente pour la faisabilité des flux est que le facteur d'utilisation du processeur du nœud $slow$ soit inférieur ou égal à 1, or $U_{j \in hp_i \cup sp_i(t)}^{slow} < U^{slow}$ (voir remarque 4.3.1, page 51).

Condition 6.3.1 Pour tout $t \geq -J_i$, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, c'est-à-dire si le facteur d'utilisation sur le nœud $slow$ des flux τ_j tels que $j \in hp_i \cup sp_i(t)$ est strictement inférieur à 1, alors la suite $\mathcal{W}_i^q(t)$ est convergente.

Démonstration La suite $\mathcal{W}_i^q(t)$ est une suite non-décroissante car la fonction *partie entière* ($\lfloor \cdot \rfloor$) est non-décroissante. De plus, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, $\mathcal{W}_i^q(t)$ est bornée supérieurement par $X/(1 - U_{j \in hp_i \cup sp_i(t)}^{slow})$, où :

$$X = \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t).$$

En effet, par récurrence, nous avons : $\mathcal{W}_i^{q(0)}(t) \leq X \leq X/(1 - U_{j \in hp_i \cup sp_i(t)}^{slow})$ si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$. Nous montrons maintenant que si la récurrence est vraie au rang k , alors elle est vraie au rang $k+1$.

$$\begin{aligned} \mathcal{W}_i^{q(k+1)}(t) &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - S_{min_j^q} + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - S_{min_j^q}) + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} \\ &\quad + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t). \end{aligned}$$

Puisque $\mathcal{W}_i^{q(k)}(t) - S_{min_j^q} \leq \mathcal{W}_i^{q(k)}(t)$ et pour tout $x \in \mathbb{R}^+$, $\lfloor x \rfloor \leq x$, il vient :

$$\begin{aligned}
\mathcal{W}_i^{q(k+1)}(t) &\leq \sum_{j \in hp_i} \left(1 + \frac{\mathcal{W}_i^{q(k)}(t) + J_j}{T_j}\right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \frac{\mathcal{W}_i^{q(k)}(t) + J_j}{T_j}\right) \cdot C_j^{slow} \\
&\quad + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) \\
&\leq \mathcal{W}_i^{q(k)}(t) \cdot \sum_{j \in hp_i \cup sp_i(t)} \frac{C_j^{slow}}{T_j} + \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{J_j}{T_j}\right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) \\
&\leq \mathcal{W}_i^{q(k)}(t) \cdot U_{j \in hp_i \cup sp_i(t)}^{slow} + X \\
&\leq X \cdot U_{j \in hp_i \cup sp_i(t)}^{slow} / (1 - U_{j \in hp_i \cup sp_i(t)}^{slow}) + X = X / (1 - U_{j \in hp_i \cup sp_i(t)}^{slow}).
\end{aligned}$$

La suite $\mathcal{W}_i^q(t)$ est non-décroissante et bornée supérieurement, elle est donc convergente et la limite vérifie :

$$\begin{aligned}
\mathcal{W}_i^q(t) &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - \text{Sin}_j^q + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - \text{Sin}_j^q) + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} \\
&\quad + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t). \quad \blacksquare
\end{aligned}$$

D'après la condition précédente, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, alors $\mathcal{W}_i^q(t)$ existe. Nous pouvons alors calculer le temps de réponse pire cas de bout-en-bout du paquet m , égal à son instant de démarrage au plus tard sur le nœud q plus son temps de traitement sur le nœud q moins son instant de génération, soit : $W_i^q(t) + C_i^q - t$. Nous nous intéressons maintenant au temps de réponse pire cas de bout-en-bout du flux τ_i , obtenu en calculant le maximum des temps de réponse pire cas de ses paquets, soit : $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. Dans la sous-section suivante, nous déterminons les instants à considérer parmi les instants $t \geq -J_i$. Nous verrons alors dans la sous-section 6.3.5 l'expression du temps de réponse pire cas de bout-en-bout du flux τ_i .

6.3.4 Instants à tester

Le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à : $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. Nous déterminons les instants à considérer parmi les instants $t \geq -J_i$. Les deux lemmes suivants permettent ainsi de réduire l'ensemble des instants à tester pour le calcul du temps de réponse pire cas de bout-en-bout du flux τ_i .

Lemme 6.3.6 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont traités FP/DP* dans chacun des nœuds visités, alors pour tout flux $\tau_i \in \tau$ et pour tout instant t tel que $\overline{sp}_i(t) = \emptyset$, nous avons : $W_i^q(t + \mathcal{B}_i^{slow}) \leq W_i^q(t) + \mathcal{B}_i^{slow}$, avec $\mathcal{B}_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil \mathcal{B}_i^{slow} / T_j \rceil \cdot C_j^{slow}$.*

Démonstration Nous considérons la suite $\mathcal{W}_i^q(t)$ définie page 89 et prouvons ce lemme par récurrence. Pour tout instant t tel que $\overline{sp}_i(t) = \emptyset$, c'est-à-dire tel que $\forall j \in sp_i, P_j(-J_j) \geq P_i(t)$, nous avons pour tout $\alpha \in \mathbb{N} : sp_i(t + \alpha) = sp_i(t)$ et $\overline{sp}_i(t + \alpha) = \emptyset$. Ainsi, pour tout t tel que $\overline{sp}_i(t) = \emptyset$, pour tout $\alpha \in \mathbb{N}$, $\mathcal{A}_i(t + \alpha) = \sum_{h=1}^q \max_{h \neq slow} \max_{j \in hp_i \cup sp_i(t+\alpha) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t + \alpha) + (q - 1) \cdot Lmax = \mathcal{A}_i(t)$.

Nous obtenons alors au rang 0 :

$$\begin{aligned}
\mathcal{W}_i^{q(0)}(t + \mathcal{B}_i^{slow}) &= \sum_{j \in hp_i \cup sp_i(t + \mathcal{B}_i^{slow})} C_j^{slow} + \left(1 + \left\lfloor \frac{t + \mathcal{B}_i^{slow} + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t + \mathcal{B}_i^{slow}) \\
&= \sum_{j \in hp_i \cup sp_i(t)} C_j^{slow} + \left(1 + \left\lfloor \frac{t + \mathcal{B}_i^{slow} + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t).
\end{aligned}$$

Pour tout $(a, b) \in \mathbb{R}^{+2}$, $\lfloor a + b \rfloor \leq \lfloor a \rfloor + \lfloor b \rfloor$, soit :

$$\begin{aligned}
\mathcal{W}_i^{q(0)}(t + \mathcal{B}_i^{slow}) &\leq \sum_{j \in hp_i \cup sp_i(t)} C_j^{slow} + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) + \left\lceil \frac{\mathcal{B}_i^{slow}}{T_i} \right\rceil \cdot C_i^{slow} \\
&\leq \mathcal{W}_i^{q(0)}(t) + \left\lceil \frac{\mathcal{B}_i^{slow}}{T_i} \right\rceil \cdot C_i^{slow}.
\end{aligned}$$

Puisque $B_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil B_i^{slow}/T_j \rceil \cdot C_j^{slow}$, nous obtenons $\mathcal{W}_i^{q(0)}(t + \mathcal{B}_i^{slow}) \leq \mathcal{W}_i^{q(0)}(t) + \mathcal{B}_i^{slow}$.

En supposant la récurrence vraie au rang k , nous montrons qu'elle est vraie au rang $k + 1$. En effet :

$$\begin{aligned}
\mathcal{W}_i^{q(k+1)}(t + \mathcal{B}_i^{slow}) &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t + \mathcal{B}_i^{slow}) - \text{Min}_j^q + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\
&\quad + \sum_{j \in sp_i(t + \mathcal{B}_i^{slow})} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t + \mathcal{B}_i^{slow}); \mathcal{W}_i^{q(k)}(t + \mathcal{B}_i^{slow}) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\
&\quad + \left(1 + \left\lfloor \frac{t + \mathcal{B}_i^{slow} + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t + \mathcal{B}_i^{slow}).
\end{aligned}$$

Par conséquent, pour tout instant t tel que $\overline{sp}_i(t) = \emptyset$, nous obtenons :

$$\begin{aligned}
\mathcal{W}_i^{q(k+1)}(t + \mathcal{B}_i^{slow}) &\leq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - \text{Min}_j^q + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} + \sum_{j \in hp_i} \left\lceil \frac{\mathcal{B}_i^{slow}}{T_j} \right\rceil \cdot C_j^{slow} \\
&\quad + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left\lceil \frac{\mathcal{B}_i^{slow}}{T_j} \right\rceil \cdot C_j^{slow} \\
&\quad + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \left\lceil \frac{\mathcal{B}_i^{slow}}{T_i} \right\rceil \cdot C_i^{slow} + \mathcal{A}_i(t) \\
&\leq \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - \text{Min}_j^q + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\
&\quad + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\
&\quad + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) + \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \left\lceil \frac{\mathcal{B}_i^{slow}}{T_j} \right\rceil \cdot C_j^{slow} \\
&\leq \mathcal{W}_i^{q(k+1)}(t) + \mathcal{B}_i^{slow}.
\end{aligned}$$

■

Lemme 6.3.7 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont traités FP/DP* dans chacun des nœuds visités, le temps de réponse pire cas d'un flux quelconque τ_i est atteint par un paquet généré à un instant t tel que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.*

Démonstration Nous considérons deux instants, t_1 et t_2 , tels que $G_{j,i}(t_1)$ et $G_{j',i}(t_2)$ soient deux instants consécutifs parmi ceux définis dans l'énoncé du lemme. Nous prouvons que si un paquet de τ_i est généré à un instant $t' \in]t_1, t_2[$, alors son instant de démarrage au plus tard sur le nœud q est le même que s'il avait été généré en t_1 . Nous commençons par montrer que $\overline{sp}_i(t') = \overline{sp}_i(t_1)$ en procédant en deux étapes :

Etape 1 : Nous prouvons que si $j \in \overline{sp}_i(t')$, alors $j \in \overline{sp}_i(t_1)$. Pour tout instant $t' > t_1$, nous avons $P_i(t') < P_i(t_1)$. Par conséquent, si $j \in \overline{sp}_i(t')$, c'est-à-dire si $P_j(-J_j) < P_i(t')$, alors $j \in \overline{sp}_i(t_1)$, puisque $P_j(-J_j) < P_i(t') < P_i(t_1)$.

Etape 2 : Nous prouvons que si $j \in \overline{sp}_i(t_1)$, alors $j \in \overline{sp}_i(t')$. Pour cela, nous raisonnons par l'absurde. Supposons qu'il existe un flux τ_j tel que $j \in \overline{sp}_i(t_1)$ et $j \notin \overline{sp}_i(t')$. Dans ce cas, nous avons : $P_i(t') \leq P_j(-J_j) < P_i(t_1)$. Par conséquent, il existe un instant $t \in]t_1, t']$ tel que $P_i(t) = P_j(-J_j)$, c'est-à-dire tel que $G_{j,i}(t) = -J_j + k_j \cdot T_j$, avec $j \in sp_i(t)$ et $k_j = 0$. Or d'après l'hypothèse de départ, il n'existe aucun instant $t \in]t_1, t_2[$ tel que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$. D'où la contradiction.

D'après les deux points précédents, $\overline{sp}_i(t') = \overline{sp}_i(t_1)$. Puisque pour tout $t \geq -J_i$, $sp_i = sp_i(t) \cup \overline{sp}_i(t)$, nous obtenons également $sp_i(t') = sp_i(t_1)$.

Par ailleurs, nous avons par définition : $\forall j \in sp_i(t') \cup \{i\}$, $\lfloor (G_{j,i}(t') + J_j)/T_j \rfloor = \lfloor (G_{j,i}(t_1) + J_j)/T_j \rfloor$. De plus, puisque $G_{i,i}(t') = t'$, il vient $\lfloor (t' + J_i)/T_i \rfloor = \lfloor (t_1 + J_i)/T_i \rfloor$. Nous procédons alors par récurrence sur la suite $\mathcal{W}_i^q(t)$ pour montrer que pour tout instant $t' \in]t_1, t_2[$, $W_i^q(t') = W_i^q(t_1)$. Au rang 0, nous avons :

$$\begin{aligned} \mathcal{W}_i^{q(0)}(t') &= \sum_{j \in hp_i \cup sp_i(t')} C_j^{slow} + \left(1 + \left\lfloor \frac{t' + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t') \\ &= \sum_{j \in hp_i \cup sp_i(t_1)} C_j^{slow} + \left(1 + \left\lfloor \frac{t_1 + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t_1) = \mathcal{W}_i^{q(0)}(t_1). \end{aligned}$$

En supposant la récurrence vraie au rang k , nous avons $\mathcal{W}_i^{q(k+1)}(t')$ égal à :

$$\begin{aligned} &\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{q(k)}(t') - \text{Min}_j^q + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} + \sum_{j \in sp_i(t')} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t'); W_i^{q(k)}(t') - \text{Min}_j^q) + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\ &+ \left(1 + \left\lfloor \frac{t' + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t'), \\ &= \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{q(k)}(t_1) - \text{Min}_j^q + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} + \sum_{j \in sp_i(t_1)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t_1); W_i^{q(k)}(t_1) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor\right)^+ \cdot C_j^{slow} \\ &+ \left(1 + \left\lfloor \frac{t_1 + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t_1) \\ &= \mathcal{W}_i^{q(k+1)}(t_1). \end{aligned}$$

Les deux séries étant convergentes, leurs limites satisfont $W_i^q(t') = W_i^q(t_1)$. ■

RÉCAPITULATIF : Les deux lemmes précédents ont permis de limiter, puis de réduire le nombre d'instants à tester pour le calcul du temps de réponse pire cas de bout-en-bout du flux τ_i . En effet, au début de cette section, la définition du temps de réponse pire cas de bout-en-bout du flux τ_i était la suivante : $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. Les lemmes 6.3.6 et 6.3.7 ont montré qu'il n'était pas nécessaire de tester tous les instants $t \geq -J_i$. Plus précisément, si nous notons \bar{t}_i^\emptyset le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$, seuls les instants vérifiant les deux conditions suivantes doivent être considérés :

- $-J_i \leq t < \bar{t}_i^\emptyset + B_i^{slow}$, avec $B_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil B_i^{slow} / T_j \rceil \cdot C_j^{slow}$;
- $G_{j,i}(t) = -J_l + k_l \cdot T_l$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.

6.3.5 Temps de réponse pire cas de bout-en-bout

La propriété suivante précise le temps de réponse pire cas de bout-en-bout du flux τ_i .

Propriété 6.3.3 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/DP* dans chacun des nœuds visités, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lceil \frac{W_i^q(t) - S_{min_j^q} + J_j}{T_j} \right\rceil^+ \right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lceil \frac{\min(G_{j,i}(t); W_i^q(t) - S_{min_j^q}) + J_j}{T_j} \right\rceil^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$$

et \mathcal{S}_i l'ensemble des instants t tels que :

- $-J_i \leq t < \bar{t}_i^\emptyset + B_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$.

Démonstration L'expression de $W_i^q(t)$ est donnée par la propriété 6.3.2, page 88. Son expression est une formule récursive pour laquelle il existe une solution si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$ (voir la condition 6.3.1).

De plus, le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. D'après les lemmes 6.3.6 et 6.3.7, seuls les instants t appartenant à l'ensemble \mathcal{S}_i sont à considérer. ■

Remarque 6.3.3 *Nous pouvons remarquer qu’avec un ordonnancement de type FP/DP*, les résultats obtenus pour une ligne de diffusion dépendent principalement d’un calcul sur le nœud le plus lent, similaire à celui effectué en contexte monoprocesseur. Nous n’avons pas cette similitude avec un ordonnancement de type FP/DP, c’est-à-dire lorsque la priorité dynamique d’un paquet dépend du nœud visité³.*

Après avoir déterminé le temps de réponse pire cas d’un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités, nous étudions dans la section suivante trois hypothèses concernant le nombre de nœuds visités et les priorités fixes des flux. Nous comparons ainsi nos résultats avec ceux établis dans le chapitre précédent et obtenons des résultats pour l’ordonnancement FP seul, puis pour un ordonnancement DP seul.

6.4 CAS PARTICULIERS

Nous nous intéressons dans cette section à trois cas particuliers, à savoir :

- 1) la ligne de diffusion est réduite à un seul nœud ;
- 2) les flux ont tous une priorité fixe différente ;
- 3) les flux ont tous la même priorité fixe.

Le premier cas permet de comparer les résultats obtenus pour une ligne de diffusion avec ceux obtenus en contexte monoprocesseur. Les deux autres cas permettent d’obtenir des résultats en contexte monoprocesseur lorsque les flux sont ordonnancés FP (cas 2) ou DP (cas 3).

6.4.1 La ligne de diffusion est réduite à un seul nœud

Il est intéressant de souligner que dans le cas où la ligne de diffusion est réduite à un seul nœud, nous retrouvons les résultats établis en contexte monoprocesseur (chapitre 5). En effet, dans ce cas, nous avons pour tout flux $\tau_j : S_{min_j}^1 = 0$, soit $W_i^1(t) - S_{min_j}^1 = W_i^1(t)$. Le nombre d’instantanés à tester est toutefois plus important dans la propriété 6.3.3 (ligne de diffusion) que dans la propriété 5.3.2 (contexte monoprocesseur). Cela est dû au fait qu’en contexte monoprocesseur, le temps de réponse pire cas d’un flux est obtenu dans un scénario particulier (ou plus exactement dans la première période active du scénario). Il est alors possible de limiter davantage les instantanés à tester, en ne considérant pas ceux de la forme $t + \alpha \cdot T_i$, $\alpha \in \mathbb{N}^*$, si $W_i(t) + C_i \leq t + T_i$.

6.4.2 Les flux ont tous une priorité fixe différente

Lorsque tous les flux ont une priorité fixe différente, le traitement DP* n’intervient pas dans l’ordonnancement FP/DP*, puisque DP* permet de départager des paquets ayant la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l’algorithme FP. Si tous les flux ont une priorité fixe différente, alors pour tout flux $\tau_i : sp_i = \emptyset$. En appliquant ce résultat à la propriété 6.3.3, nous obtenons la propriété suivante.

³Les ordonnancements de type FP/DP et FP/DP* sont présentés dans la section 2.2.3, page 17.

Propriété 6.4.1 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP dans chacun des nœuds visités, si $U_{j \in hp_i}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^q(t) - \text{min}_j^q + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow}$$

$$+ \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot Lmax$$

et $\mathcal{S}_i = \{t, t = -J_i + k \cdot T_i < -J_i + \mathcal{B}_i^{slow}, k \in \mathbb{N}\}$.

Démonstration Nous obtenons l'expression de $W_i^q(t)$ ci-dessus en appliquant $sp_i = \emptyset$ à la propriété 6.3.3. Par ailleurs, les instants t à tester doivent satisfaire la condition suivante : $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$. Puisque $sp_i = \emptyset$ et $G_{i,i}(t) = t$, seuls les instants de la forme $-J_i + k \cdot T_i$ doivent être considérés. De plus, le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$ est l'instant $-J_i$. ■

6.4.3 Les flux ont tous la même priorité fixe

Lorsque tous les flux partagent la même priorité fixe, le traitement FP n'intervient pas dans l'ordonnancement FP/DP*. En effet, seul l'ordonnancement permettant de départager les paquets ayant la même priorité fixe est sollicité. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant un ordonnancement DP*.

Si tous les flux ont la même priorité fixe, alors pour tout flux $\tau_i : hp_i = lp_i = \emptyset$. Par ailleurs, nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m .

De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i ne pouvant pas générer de paquets avec une priorité dynamique supérieure ou égale à celle de m , mais susceptibles de générer un paquet pouvant gêner m à cause de la non-préemption.

En appliquant ces résultats à la propriété 6.3.3, nous obtenons la propriété suivante, donnant le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont traités dans chacun des nœuds de la ligne de diffusion suivant un ordonnancement de type DP*.

Propriété 6.4.2 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés DP^* dans chacun des nœuds visités, si $U_{j \in sp_i(t)}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :

$$R_i^{1,q} = \max_{t \in S_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^q(t) - S_{min_i^q}) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot Lmax$$

et S_i l'ensemble des instants t tels que :

- $-J_i \leq t < \bar{t}_i^\emptyset + \mathcal{B}_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l$, $k_l \in \mathbb{N}$.

Démonstration D'après la propriété 6.3.3, avec $hp_i = lp_i = \emptyset$. ■

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP^* dans chacun des nœuds visités (section 6.3), nous avons considéré trois cas, à savoir (i) la ligne de diffusion est réduite à un seul nœud, (ii) les flux ont tous une priorité fixe différente et (iii) les flux ont tous la même priorité fixe. Ces trois cas ont permis de comparer les résultats obtenus avec ceux établis dans le chapitre précédent en contexte monoprocesseur, puis de déterminer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux sont ordonnancés respectivement FP et DP^* dans les nœuds de la ligne de diffusion.

Nous nous intéressons maintenant à deux ordonnancements DP^* particuliers, à savoir $FIFO^*$ (section 6.5) et EDF^* (section 6.6).

6.5 ORDONNANCEMENT $FP/FIFO^*$

Dans cette section, nous montrons comment calculer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous une même ligne de diffusion composée de q nœuds et sont ordonnancés $FP/FIFO^*$ dans chacun des nœuds visités. Nous utilisons pour cela les résultats établis dans la section 6.3 et les appliquons à l'ordonnancement $FP/FIFO^*$. Nous retrouvons les résultats établis par ailleurs [70].

Remarque 6.5.1 Avec l'ordonnancement FP/FIFO*, l'ordre de traitement des paquets est le même dans chacun des nœuds visités. En effet, les paquets sont traités dans le premier nœud suivant leurs instants de génération. Les liens étant supposés FIFO, l'ordre des instants d'arrivée des paquets dans les nœuds suivants est le même que celui des instants de génération. Dans le cas d'une simple ligne de diffusion, l'ordonnancement FP/FIFO* est donc équivalent à l'ordonnancement FP/FIFO⁴.

Nous considérons un flux quelconque τ_i . Soit m le paquet de τ_i généré à l'instant t . Il a été montré dans la section 5.6, page 66, les trois points suivants :

- $\forall t \geq -J_i, sp_i(t) = \{j \in sp_i, -J_j \leq t\}$;
- $\forall t \geq -J_i, \overline{sp}_i(t) = \emptyset$;
- $\forall j \in sp_i(t), G_{j,i}(t) = t$.

Nous avons établi page 93 la propriété 6.3.3 donnant le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités. En appliquant les résultats ci-dessus à cette propriété, nous définissons le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont ordonnancés FP/FIFO*.

6.5.1 Temps de réponse pire cas de bout-en-bout

La propriété suivante établit le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/FIFO* dans chacun des nœuds visités, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs instants de génération.

Propriété 6.5.1 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/FIFO* dans chacun des nœuds visités, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lceil \frac{W_i^q(t) - S_{min_j^q} + J_j}{T_j} \right\rceil^+ \right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lceil \frac{\min(t, W_i^q(t) - S_{min_j^q}) + J_j}{T_j} \right\rceil^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$$

$$\text{et } \mathcal{S}_i = \{t, -J_i \leq t = -J_j + k_j \cdot T_j < -J_i + B_i^{slow}, j \in sp_i(t) \cup \{i\}, k_j \in \mathbb{N}\}.$$

Démonstration D'après la propriété 6.3.3, avec $\bar{t}_i^\emptyset = -J_i$ (puisque $\overline{sp}_i(t) = \emptyset$ pour tout $t \geq -J_i$), $sp_i(t) = \{j \in sp_i, -J_j \leq t\}$ et pour tout flux τ_j tel que $j \in sp_i(t)$, $G_{j,i}(t) = t$. ■

⁴Les ordonnancements de type FP/DP et FP/DP* sont présentés dans la section 2.2.3, page 17.

6.5.2 Ordonnancement FIFO*

Nous avons vu dans la section 6.4 que le traitement FP n'intervient pas dans l'ordonnancement FP/FIFO* lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme FIFO*. Si tous les flux ont la même priorité fixe, alors pour tout flux τ_i : $hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 6.5.1, nous obtenons la propriété suivante.

Propriété 6.5.2 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FIFO* dans chacun des nœuds visités, si $U_{j \neq i, -J_j \leq t}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :*

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \neq i, -J_j \leq t} \left(1 + \left\lfloor \frac{\min(t; W_i^q(t) - S \min_j^q) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j \leq t} \{C_j^h\} - C_i^q + (q-1) \cdot L_{max}$$

$$\text{et } \mathcal{S}_i = \{t, -J_i \leq t = -J_j + k_j \cdot T_j < -J_i + B_i^{slow}, -J_j \leq t, k_j \in \mathbb{N}\}.$$

Démonstration D'après la propriété 6.5.1, avec $hp_i = lp_i = \emptyset$. ■

Remarque 6.5.2 *Nous venons d'établir de nouveaux résultats pour l'ordonnancement FIFO* (équivalent à l'ordonnancement FIFO) lorsque les flux suivent tous la même ligne de diffusion. Afin de réduire le temps de calcul du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , il est possible de considérer la relation suivante dans la propriété 6.5.2 : $\min(t; W_i^q(t) - S \min_j^q) \leq t$. Ainsi, nous obtenons simplement :*

$$R_i^{1,q} \leq \max_{t \in \mathcal{S}_i} \left\{ \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t + J_j}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j \leq t} \{C_j^h\} + (q-1) \cdot L_{max} - t \right\}.$$

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* (section 6.3), nous avons appliqué nos résultats à un ordonnancement DP* particulier : FIFO*. Le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/FIFO* a ainsi été établi. Nous avons ensuite considéré le cas où les flux ont tous la même priorité fixe. Cela nous a permis d'établir de nouveaux résultats pour une ligne de diffusion lorsque les flux sont ordonnancés FIFO* (équivalent à FIFO).

Nous présentons maintenant des exemples comparant les résultats obtenus avec l'ordonnancement FP/FIFO* aux résultats obtenus avec un outil que nous avons développé, donnant la solution exhaustive à un problème d'ordonnancement temps-réel dans un réseau. Afin de montrer l'intérêt de notre approche, nous comparons également nos résultats à ceux obtenus avec l'approche holistique.

6.5.3 Exemples

Dans cette section, nous présentons des exemples de bornes sur les temps de réponse de bout-en-bout de flux dans un réseau, lorsque les flux suivent tous une même ligne de diffusion composée de cinq nœuds et sont ordonnancés FP/FIFO* dans chacun des nœuds visités. Nous supposons que $\tau = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, chaque flux τ_i ayant une période $T_i = 36$ et une gigue d'activation nulle. De plus, nous posons $L_{max} = L_{min} = 1$. Les priorités et les échéances de bout-en-bout de ces flux sont données dans le tableau 6.1.

TAB. 6.1: Caractéristiques des flux considérés

	τ_1	τ_2	τ_3	τ_4	τ_5
P_i	1	1	2	2	3
D_i	47	50	44	45	39

Dans un premier temps, nous considérons que les flux ont tous le même temps de traitement maximum égal à 4 dans chacun des nœuds visités, soit : $\forall i \in [1, 5], \forall h \in [1, 5], C_i^h = 4$. La figure 6.2 présente pour chaque flux τ_i la valeur exacte de son temps de réponse pire cas de bout-en-bout et la valeur calculée avec nos résultats. Les résultats obtenus avec l'approche holistique sont également représentés. Nous montrons ainsi l'intérêt d'appliquer l'approche par trajectoire pour la détermination du temps de réponse pire cas d'un flux.

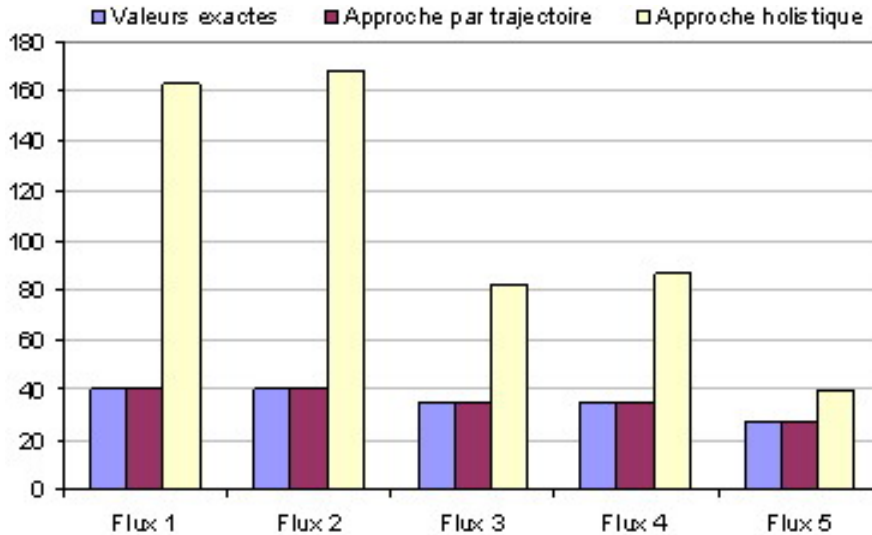


FIG. 6.2: Temps de réponse pire cas de bout-en-bout

Les valeurs obtenues avec l'approche par trajectoire sont exactes pour tous les flux considérés, alors que celles obtenues avec l'approche holistique surestiment jusqu'à 450% les valeurs exactes.

⁴Cette valeur est fournie par un outil informatique que nous avons développé (voir section 2.4, page 22).

Nous considérons maintenant un cas plus général en supposant que le temps de traitement maximum d'un flux n'est pas le même dans chacun des nœuds visités. Plus précisément, nous étudions dix configurations, présentées dans le tableau 6.2. La charge est égale à 83,3% pour chaque configuration.

TAB. 6.2: Temps de traitement maximum de tout flux τ_i sur chaque nœud

	1	2	3	4	5	6	7	8	9	10
C_i^1	6	2	5	4	3	4	5	5	2	6
C_i^2	2	6	3	2	5	6	4	3	6	2
C_i^3	5	4	6	6	2	2	3	4	3	3
C_i^4	3	3	2	3	6	5	2	6	5	4
C_i^5	4	5	4	5	4	3	6	2	4	5

La figure 6.3 présente pour le flux τ_4 son temps de réponse pire cas de bout-en-bout exact et celui calculé avec l'approche par trajectoire (propriété 6.5.1). Les résultats obtenus avec l'approche holistique sont également représentés.

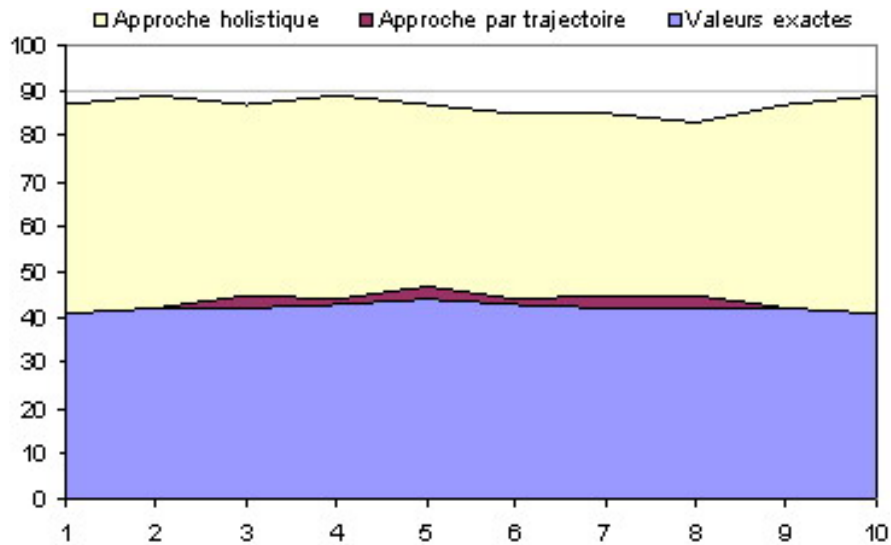


FIG. 6.3: Temps de réponse pire cas de bout-en-bout du flux τ_4

Nous pouvons remarquer que nos résultats sont exacts ou très proches des valeurs exactes (surestimation maximale de l'ordre de 7%). Par contre, les bornes obtenues avec l'approche holistique sont très pessimistes. En effet, ces valeurs sont environ deux fois plus élevées que les valeurs exactes, quelque soit la configuration étudiée.

6.6 ORDONNANCEMENT FP/EDF*

Dans cette section, nous montrons comment calculer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/EDF* dans chacun des nœuds visités. Nous utilisons pour cela les résultats établis dans la section 6.3 et les appliquons à l'ordonnancement FP/EDF*. Nous retrouvons alors les résultats établis dans [71].

Nous considérons un flux quelconque τ_i . Soit m le paquet de τ_i généré à l'instant t . Nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m . De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i susceptibles de générer un paquet de priorité dynamique strictement inférieure à celle de m mais pouvant gêner m dû à la non-préemption.

Avec l'ordonnancement EDF*, nous considérons que la priorité dynamique du paquet de τ_i généré à l'instant t est égale à : $t + D'_i$, avec D'_i l'échéance relative du flux τ_i sur son nœud source. L'avantage de cette solution est qu'elle ne nécessite pas la synchronisation des horloges des nœuds au cœur du réseau. Seuls les nœuds en bordure (et plus précisément les nœuds d'entrée) doivent avoir des horloges synchronisées afin d'assigner à chaque paquet entrant sa priorité dynamique. Le calcul de D'_i est réalisé à partir de l'échéance de bout-en-bout. Si deux flux ont la même échéance de bout-en-bout, celui parcourant le plus grand nombre de nœuds doit avoir la plus petite échéance intermédiaire. Une manière simple d'obtenir ce résultat est de poser : $D'_i = \lfloor D_i/q \rfloor$. D'autres solutions existent [72, 16], basées par exemple sur la prise en compte de la charge dans les nœuds visités. La sélection de la meilleure valeur n'est pas dans le cadre de cette thèse.

Ainsi, dans chaque nœud de la ligne de diffusion, le paquet m est ordonnancé en fonction de sa priorité fixe P_i puis de son échéance absolue $t + D'_i$, calculée sur le nœud 1. Il est important de noter que la priorité dynamique de m n'est utilisée que pour ordonnancer ce paquet avec ceux ayant la même priorité fixe. En d'autres termes, les nœuds appliquent l'ordonnancement EDF sur $t + D'_i$ mais l'échéance à respecter est $t + D_i$. Par conséquent, un paquet peut dépasser son échéance intermédiaire dans un nœud et respecter son échéance de bout-en-bout.

Un paquet d'un flux quelconque τ_j , $j \in sp_i$, a une priorité dynamique supérieure ou égale à m si son échéance absolue est plus petite que celle de m . Puisque la priorité dynamique de m est $t + D'_i$, nous avons pour tout instant $t \geq -J_i$:

- $sp_i(t) = \{j \in sp_i, P_j(-J_j) \geq P_i(t)\} = \{j \in sp_i, -J_j + D'_j \leq t + D'_i\}$;
- $\overline{sp}_i(t) = \{j \in sp_i, P_j(-J_j) < P_i(t)\} = \{j \in sp_i, -J_j + D'_j > t + D'_i\}$.

Par ailleurs, puisqu'un paquet d'un flux quelconque τ_j , $j \in sp_i(t)$, est plus prioritaire que m s'il est généré au plus tard en $t + D'_i - D'_j$, nous obtenons :

- $\forall j \in sp_i(t), G_{j,i}(t) = t + D'_i - D'_j$.

Nous avons établi page 93 la propriété 6.3.3 donnant le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités. En appliquant les résultats ci-dessus à cette propriété, nous définissons le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont ordonnancés FP/EDF*.

6.6.1 Temps de réponse pire cas

La propriété suivante établit le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/EDF* dans chacun des nœuds visités, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs échéances absolues.

Propriété 6.6.1 *Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés FP/EDF* dans chacun des nœuds visités, si $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :*

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^q(t) - \text{Min}_j^q + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t + D'_i - D'_j; W_i^q(t) - \text{Min}_j^q) + J_j}{T_j} \right\rfloor^+\right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$$

$$\text{et } \mathcal{S}_i = \left\{ t, -J_i \leq t = -J_l + k_l \cdot T_l - D'_i + D'_j < \max_{k \in sp_i} \{D'_k - J_k\} - D'_i + \mathcal{B}_i^{slow}, \right. \\ \left. \text{avec } j \text{ et } l \text{ appartenant à } sp_i(t) \cup \{i\} \text{ et } k_l \in \mathbb{N} \right\}.$$

Démonstration D'après la propriété 6.3.3, en appliquant $sp_i(t) = \{j \in sp_i, -J_j + D'_j \leq t + D'_i\}$, $\overline{sp}_i(t) = \{j \in sp_i, -J_j + D'_j > t + D'_i\}$ et $\forall j \in sp_i(t), G_{j,i}(t) = t + D'_i - D'_j$, nous trouvons l'expression de $W_i^q(t)$ lorsque les flux sont ordonnancés FP/EDF*. Concernant les instants à tester, seuls ceux respectant les deux conditions suivantes doivent être considérés :

- $-J_i \leq t < \bar{t}_i^\emptyset + \mathcal{B}_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists (j, l) \in sp_i(t) \cup \{i\}, t + D'_i - D'_j = -J_l + k_l \cdot T_l, k_l \in \mathbb{N}$.

Or pour tout flux τ_j tel que $j \in sp_i$, si $t \geq \max_{k \in sp_i} \{-J_k + D'_k\} - D'_i$, alors nous obtenons : $t + D'_i \geq \max_{k \in sp_i} \{-J_k + D'_k\} \geq -J_j + D'_j$. Par conséquent, $P_j(-J_j) \geq P_i(t)$, soit $j \in sp_i(t)$, d'où $\overline{sp}_i(t) = \emptyset$. L'instant \bar{t}_i^\emptyset est donc égal à : $\max_{k \in sp_i(t)} \{D'_k - J_k\} - D'_i$. Les instants à tester sont alors ceux de l'ensemble \mathcal{S}_i défini dans la propriété 6.6.1. ■

6.6.2 Ordonnancement EDF*

Nous avons vu dans la section 6.4 que le traitement FP n'intervient pas dans l'ordonnancement FP/EDF* lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme EDF*. Si tous les flux ont la même priorité fixe, alors pour tout flux $\tau_i : hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 6.6.1, nous obtenons la propriété suivante.

Propriété 6.6.2 Lorsque les flux suivent tous la même ligne de diffusion composée de q nœuds et sont ordonnancés EDF* dans chacun des nœuds visités, si $U_{j \neq i, -J_j + D'_j - D'_i \leq t}^{slow} < 1$, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \neq i, -J_j + D'_j - D'_i \leq t} \left(1 + \left\lfloor \frac{\min(t + D'_i - D'_j; W_i^q(t) - S_{min_j^q}) + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \\ + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j + D'_j - D'_i \leq t} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot Lmax$$

$$\text{et } \mathcal{S}_i = \left\{ t, -J_i \leq t = -J_l + k_l \cdot T_l - D'_i + D'_j < \max_{k \neq i} \{D'_k - J_k\} - D'_i + \mathcal{B}_i^{slow}, \right. \\ \left. -J_j + D'_j - D'_i \leq t, -J_l + D'_l - D'_i \leq t, k_l \in \mathbb{N} \right\}.$$

Démonstration D'après la propriété 6.5.1, avec $hp_i = lp_i = \emptyset$. ■

Remarque 6.6.1 Nous venons d'établir de nouveaux résultats pour l'ordonnancement EDF* lorsque les flux suivent tous la même ligne de diffusion. Afin de réduire le temps de calcul du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , il est possible de considérer la relation suivante dans la propriété 6.6.2 : $\min(t + D'_i - D'_j; W_i^q(t) - S_{min_j^q}) \leq t + D'_i - D'_j$. Ainsi, nous obtenons simplement :

$$R_i^{1,q} \leq \max_{t \in \mathcal{S}_i} \left\{ \sum_{j, -J_j + D'_j - D'_i \leq t} \left(1 + \left\lfloor \frac{t + D'_i - D'_j + J_j}{T_j} \right\rfloor \right) \cdot C_j^{slow} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j + D'_j - D'_i \leq t} \{C_j^h\} + (q-1) \cdot Lmax - t \right\}.$$

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux dans le cas d'une ligne de diffusion avec un ordonnancement de type FP/DP* (section 6.3), nous avons appliqué nos résultats à un ordonnancement DP* particulier : EDF*. Le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/EDF* a ainsi été établi. Nous avons ensuite considéré le cas où les flux ont tous la même priorité fixe. Cela nous a permis d'établir de nouveaux résultats pour une ligne de diffusion lorsque les flux sont ordonnancés EDF*.

Nous présentons maintenant des exemples comparant les résultats obtenus avec l'ordonnancement FP/EDF* aux résultats exacts fournis par un outil informatique que nous avons développé. Afin de montrer l'intérêt de notre approche, nous comparons également nos résultats à ceux obtenus avec l'approche holistique.

6.6.3 Exemples

Nous présentons dans cette section des exemples de bornes sur les temps de réponse de bout-en-bout de flux dans un réseau, lorsque les flux suivent tous une même ligne de diffusion composée de cinq nœuds et sont ordonnancés FP/EDF* dans chacun des nœuds visités. Nous supposons que $\tau = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, chaque flux τ_i ayant une période $T_i = 36$ et une gigue d'activation nulle. De plus, nous posons $L_{max} = L_{min} = 1$. Les priorités et les échéances de bout-en-bout de ces flux sont données dans le tableau 6.3.

TAB. 6.3: Caractéristiques des flux considérés

	τ_1	τ_2	τ_3	τ_4	τ_5
P_i	1	1	2	2	3
D_i	47	50	44	45	39

Dans un premier temps, nous considérons que les flux ont tous le même temps de traitement maximum égal à 4 dans chacun des nœuds visités, soit : $\forall i \in [1, 5], \forall h \in [1, 5], C_i^h = 4$. La figure 6.4 présente pour chaque flux τ_i la valeur exacte de son temps de réponse pire cas de bout-en-bout fournie par un outil informatique que nous avons développé (voir section 2.4, page 2.4) et la valeur calculée avec nos résultats. Les résultats obtenus avec l'approche holistique sont également représentés. Nous montrons ainsi l'intérêt d'appliquer l'approche par trajectoire pour la détermination du temps de réponse pire cas d'un flux. Les valeurs obtenues avec l'approche par trajectoire sont exactes pour tous les flux considérés, alors que celles obtenues avec l'approche holistique surestiment jusqu'à 450% les valeurs exactes.

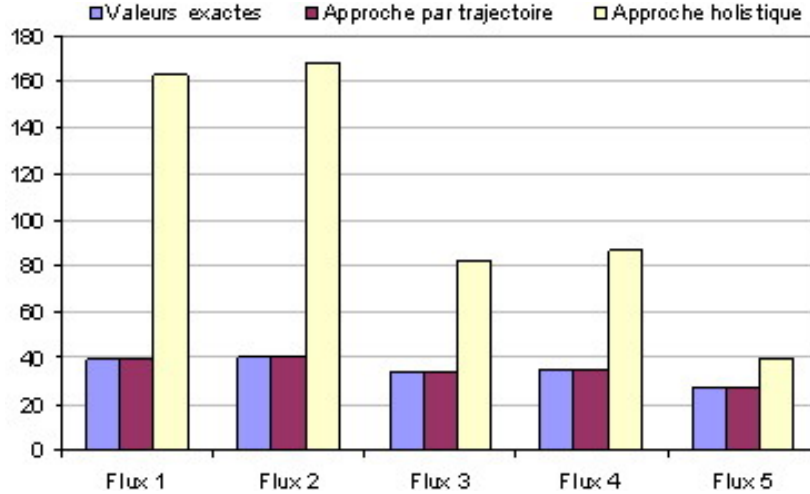
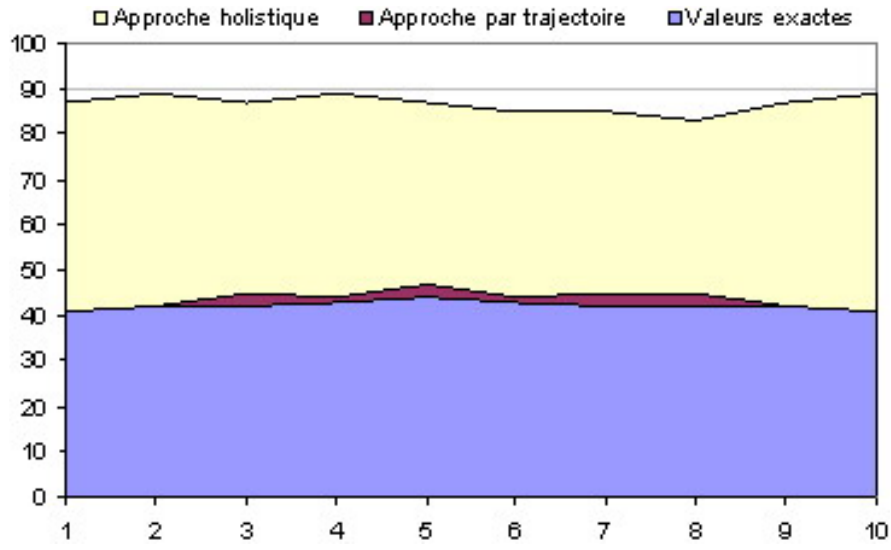


FIG. 6.4: Temps de réponse pire cas de bout-en-bout

Nous considérons maintenant un cas plus général en supposant que le temps de traitement maximum d'un flux n'est pas le même dans chacun des nœuds visités. Plus précisément, nous étudions dix configurations, présentées dans le tableau 6.4. La charge est égale à 83,3% pour chaque configuration. La figure 6.5 présente pour le flux τ_4 son temps de réponse pire cas de bout-en-bout exact et celui calculé avec l'approche par trajectoire (propriété 6.6.1). Les résultats obtenus avec l'approche holistique sont également représentés. Nous pouvons remarquer que nos résultats sont exacts ou très proches des valeurs exactes (surestimation maximale de l'ordre de 7%). Par contre, les bornes obtenues avec l'approche holistique sont très pessimistes. En effet, ces valeurs sont environ deux fois plus élevées que les valeurs exactes, quelque soit la configuration étudiée.

TAB. 6.4: Temps de traitement maximum de tout flux τ_i sur chaque nœud

	1	2	3	4	5	6	7	8	9	10
C_i^1	6	2	5	4	3	4	5	5	2	6
C_i^2	2	6	3	2	5	6	4	3	6	2
C_i^3	5	4	6	6	2	2	3	4	3	3
C_i^4	3	3	2	3	6	5	2	6	5	4
C_i^5	4	5	4	5	4	3	6	2	4	5

FIG. 6.5: Temps de réponse pire cas de bout-en-bout du flux τ_4

6.7 CONCLUSION

Nous avons déterminé dans ce chapitre le temps de réponse pire cas d'un flux sporadique quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités. Pour cela, nous avons procédé en plusieurs étapes.

- Tout d'abord, nous nous sommes intéressés à un paquet quelconque m du flux τ_i , généré à l'instant t et avons déterminé son instant de démarrage au plus tard sur le dernier nœud (propriété 6.3.2, page 88). Pour ce faire, nous avons utilisé l'approche par trajectoire, consistant à identifier sur chacun des nœuds visités la période active et les paquets affectant le temps de réponse de bout-en-bout de m .
- Comme l'expression de cet instant $W_i^q(t)$ est une formule récursive, nous avons ensuite prouvé l'existence d'une solution lorsque $U_{j \in hp_i \cup sp_i(t)}^{slow} < 1$ (condition 6.3.1, page 89), où $U_{j \in hp_i \cup sp_i(t)}^{slow}$ désigne le taux d'utilisation du processeur du nœud le plus lent de la ligne de diffusion pour les flux τ_j , $j \neq i$, (i) ayant une priorité fixe strictement supérieure à celle de τ_i et (ii) de même priorité fixe et pouvant générer un paquet de priorité dynamique supérieure ou égale à celle de m . Cette condition n'est pas contraignante puisque une condition nécessaire pour la faisabilité d'un ensemble de flux est que le facteur d'utilisation du processeur de chaque nœud h , noté U^h , soit inférieur ou égal à 1, or $U_{j \in hp_i \cup sp_i(t)}^{slow} < U^{slow}$.

- Nous avons alors pu déterminer le temps de réponse pire cas de bout-en-bout de τ_i , égal au maximum des temps de réponse pire cas de bout-en-bout de ses paquets, soit : $R_i = \max_t \{W_i^q(t) - t\} + C_i^q$, avec C_i^q le temps de traitement maximum d'un paquet de τ_i sur le nœud q . Afin de limiter le nombre d'instants t à tester pour le calcul du temps de réponse pire cas de τ_i , nous avons établi deux lemmes (pages 90 à 91).

Nous sommes ainsi arrivés à l'expression du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux suivent tous la même ligne de diffusion et sont ordonnancés FP/DP* dans chacun des nœuds visités (propriété 6.3.3, page 93).

Nous avons ensuite considérés trois cas particuliers afin de (i) comparer nos résultats avec ceux établis en monoprocesseur et (ii) déterminer le temps de réponse pire cas d'un flux lorsque les paquets sont ordonnancés respectivement FP et DP* sur chacun des nœuds visités. Puis nous avons appliqué ces résultats à deux ordonnancements DP* particuliers : FIFO* (équivalent à FIFO dans le cas d'une ligne de diffusion) et EDF*.

Cela a permis d'établir le temps de réponse pire cas de bout-en-bout d'un flux dans le cas d'une ligne de diffusion pour les ordonnancements FP/FIFO* (propriété 6.5.1, page 97) et FP/EDF* (propriété 6.6.1, page 102), mais aussi pour les ordonnancements FIFO* (propriété 6.5.2, page 98) et EDF* (propriété 6.6.2, page 103).

Différents exemples ont été présentés pour comparer nos résultats (i) aux valeurs exactes et (ii) aux résultats obtenus avec l'approche holistique. Nous avons ainsi montré que nos résultats étaient soit très proches soit les mêmes que ceux obtenus avec un outil informatique donnant les valeurs exactes. Il a été également montré que l'approche holistique fournissait dans certains cas des bornes très pessimistes, contrairement à l'approche par trajectoire.

CHAPITRE 7

Cas distribué

7.1	Introduction	108
7.2	Demarche suivie	108
7.3	Analyse pire cas	109
7.3.1	Approche par trajectoire	109
7.3.2	Evaluation des différents délais	113
7.3.3	Instant de démarrage au plus tard sur le dernier nœud visité	116
7.3.4	Instants à tester	118
7.3.5	Temps de réponse pire cas de bout-en-bout	119
7.4	Généralisation	120
7.5	Cas particuliers	120
7.5.1	Les flux suivent tous une même ligne de diffusion	120
7.5.2	Les flux ont tous une priorité fixe différente	120
7.5.3	Les flux partagent tous la même priorité fixe	121
7.6	Ordonnancement FP/FIFO*	122
7.6.1	Temps de réponse pire cas de bout-en-bout	123
7.6.2	Ordonnancement FIFO*	124
7.7	Ordonnancement FP/EDF*	125
7.7.1	Temps de réponse pire cas	125
7.7.2	Ordonnancement EDF*	126
7.8	Conclusion	128

7.1 INTRODUCTION

Dans le chapitre précédent, nous avons présenté de nouveaux résultats dans le cas d'une ligne de diffusion (tous les flux suivent la même séquence de nœuds) pour un ordonnancement de type FP/DP* non-oisif et non-préemptif de flux sporadiques dont les instants d'arrivée ne sont pas connus à priori. En effet, nous rappelons que tout paquet entrant dans le réseau se voit assigner une priorité généralisée, c'est-à-dire une priorité fixe (celle du flux auquel il appartient) et une priorité dynamique. Ainsi, dans chacun des nœuds du réseau, les paquets sont ordonnancés selon leurs priorités généralisées. Plus précisément, les paquets sont traités dans l'ordre de leurs priorités fixes (ordonnancement FP) puis ceux de même priorité fixe dans l'ordre de leurs priorités dynamiques (ordonnancement DP*). Nous avons appliqué nos résultats à deux ordonnancements DP* particuliers, à savoir FIFO* et EDF*. De nouveaux résultats ont ainsi été établis dans le cas d'une ligne de diffusion pour les ordonnancements FP/FIFO* et FP/EDF*, puis pour les ordonnancements FP, FIFO* et EDF* en considérant deux cas particuliers sur les priorités fixes des flux. Enfin, nous avons présenté plusieurs exemples comparant nos résultats avec (i) ceux obtenus par l'approche holistique et (ii) ceux fournis par une approche exhaustive, obtenus par un outil que nous avons développé.

Dans ce chapitre, nous étendons les résultats obtenus au cas général, c'est-à-dire lorsque les flux suivent des lignes de diffusion différentes. Ainsi, dans ce chapitre :

- nous déterminons le temps de réponse pire cas de bout-en-bout d'un flux lorsque les flux suivent des lignes de diffusion quelconques et sont ordonnancés FP/DP* dans chacun des nœuds visités. Pour cela, nous adoptons l'approche par trajectoire¹ ;
- nous étudions deux cas particuliers concernant les priorités fixes des flux et obtenons des résultats pour l'ordonnancement FP seul, puis pour un ordonnancement DP* seul en environnement distribué ;
- nous appliquons les résultats obtenus à deux ordonnancements DP* particuliers, à savoir FIFO* et EDF* ;
- nous comparons nos résultats à ceux obtenus avec l'approche holistique et aux valeurs exactes, obtenues avec un outil de simulation.

7.2 DEMARCHE SUIVIE

Afin de déterminer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , $i \in [1, n]$, lorsque les flux sont ordonnancés suivant un algorithme du type FP/DP* dans chacun des nœuds visités, nous étendons l'approche par trajectoire présentée dans le chapitre précédent. Plus précisément, nous nous intéressons à un paquet quelconque m du flux τ_i généré à l'instant t et nous identifions sur chacun des nœuds visités par ce paquet (en commençant par le dernier puis en remontant de nœud en nœud) la période active et les paquets affectant son temps de réponse de bout-en-bout. Cette décomposition permet d'établir l'instant de démarrage au plus tard du paquet m sur son nœud de sortie. Après une analyse de cet instant dont l'expression est une formule récursive, nous obtenons le temps de réponse pire cas de bout-en-bout du flux τ_i .

¹L'approche par trajectoire a été présentée dans le chapitre 6. Nous avons appliqué cette méthode pour la détermination du temps de réponse pire cas de bout-en-bout d'un flux dans le cas d'une ligne de diffusion.

7.3 ANALYSE PIRE CAS

Pour des raisons de clarté, nous supposons dans cette section l'hypothèse suivante (nous verrons dans la section 7.4 comment lever cette hypothèse).

Hypothèse 7.3.1 Soient deux flux τ_i et τ_j suivant respectivement les lignes \mathcal{L}_i et \mathcal{L}_j , avec $\mathcal{L}_i \neq \mathcal{L}_j$ et $\mathcal{L}_i \cap \mathcal{L}_j \neq \emptyset$. S'il existe un nœud $h \in \mathcal{L}_i \cap \mathcal{L}_j$ tel que le nœud suivant visité par τ_j n'est pas le nœud suivant visité par τ_i , alors τ_j ne visite plus aucun nœud de \mathcal{L}_i après h .

En d'autres termes, nous considérons que les flux peuvent se croiser au plus une fois. Cela est illustré par la figure 7.1 : l'hypothèse est respectée dans le cas (a), mais pas dans le cas (b) où les flux τ_1 et τ_2 se croisent deux fois.

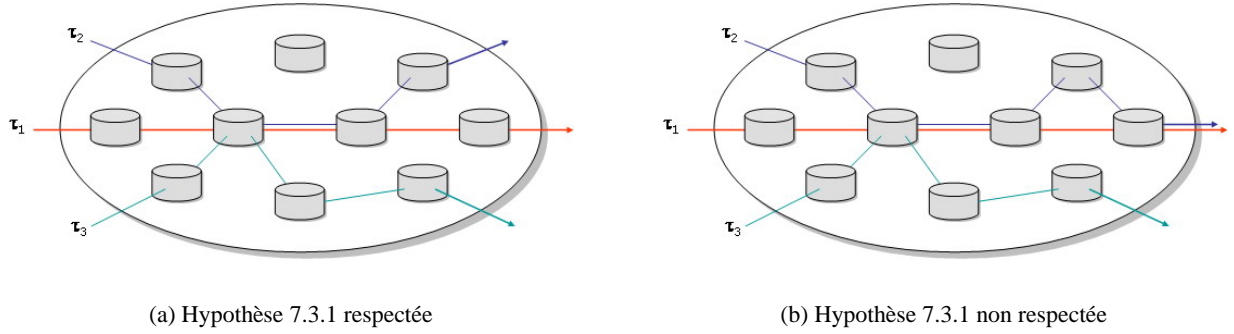


FIG. 7.1: Les flux peuvent se croiser au plus une fois

Nous nous intéressons donc à un flux quelconque τ_i , $i \in [1, n]$, suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds numérotés de 1 à q . Avant de déterminer le temps de réponse pire cas de bout-en-bout du flux τ_i , nous nous intéressons à l'un de ses paquets, m , généré à un instant t et nous calculons son instant de démarrage au plus tard sur le dernier nœud visité, le nœud q . Pour ce faire, nous identifions les périodes actives qui retardent l'instant de démarrage au plus tard de m sur le nœud q .

7.3.1 Approche par trajectoire

Nous considérons la période active de niveau $PG_i(t)$ dans laquelle m est traité sur le nœud q . Soient bp^q cette période active et $f(q)$ le premier paquet traité dans bp^q avec une priorité supérieure ou égale à $PG_i(t)$. Puisque les flux ne suivent pas tous la même ligne de diffusion, le paquet $f(q)$ ne vient pas nécessairement du nœud $q - 1$. Par conséquent, afin de remonter sur les nœuds visités par m , nous considérons un paquet supplémentaire, à savoir $p(q - 1)$: le premier paquet de priorité supérieure ou égale à $PG_i(t)$ traité entre $f(q)$ et m sur le nœud q et venant du nœud $q - 1$.

Le paquet $p(q - 1)$ a été traité sur le nœud $q - 1$ dans une période active de niveau au moins égal à $PG_i(t)$. Soient bp^{q-1} cette période active et $f(q - 1)$ le premier paquet traité dans bp^{q-1} avec une priorité supérieure ou égale à $PG_i(t)$. Nous définissons alors $p(q - 2)$: le premier paquet de priorité supérieure ou égale à $PG_i(t)$ traité entre $f(q - 1)$ et m sur le nœud $q - 1$ et venant du nœud $q - 2$.

Nous procédons ainsi jusqu'à la détermination sur le nœud 1 de la période active bp^1 dans laquelle $p(1)$ est traité et où $f(1)$ est le premier paquet de priorité supérieure ou égale à $PG_i(t)$. Par conséquent, nous avons :

- $\forall h \in [1, q], f(h)$ est le premier paquet traité dans bp^h avec une priorité supérieure ou égale à $PG_i(t)$;
- $\forall h \in [1, q], p(h-1)$ est le premier paquet de priorité supérieure ou égale à $PG_i(t)$ traité entre $f(h)$ et m sur le nœud h et venant du nœud $h-1$.

La figure 7.2 illustre cette décomposition.

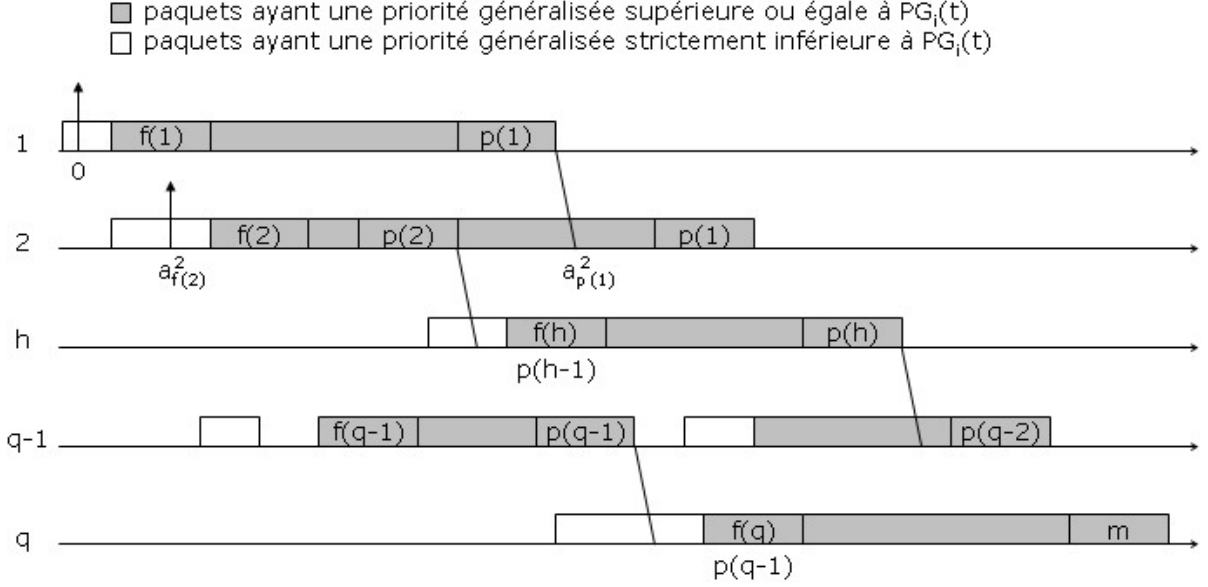


FIG. 7.2: Périodes actives de niveau $PG_i(t)$ considérées

Sur chaque nœud $h \in [1, q]$, nous numérotons successivement les paquets de priorités généralisées supérieures ou égales à $PG_i(t)$ et traités dans bp^h . De plus, nous considérons l'instant d'activation du paquet $f(1)$ comme origine des temps et posons $p(q) = m$.

A partir de cette décomposition, nous déterminons l'instant de démarrage au plus tard du paquet m sur le nœud q . En effet, $W_i^q(t)$ est égal :

$$\begin{aligned}
 & \text{au temps de traitement sur le nœud 1 des paquets } f(1) \text{ à } p(1) + L_{max} \\
 & + \text{ le temps de traitement sur le nœud 2 des paquets } f(2) \text{ à } p(2) - \left(a_{p(1)}^2 - a_{f(2)}^2 \right) + L_{max} \\
 & + \dots \\
 & + \text{ le temps de traitement sur le nœud } q \text{ des paquets } f(q) \text{ à } (m-1) - \left(a_{p(q-1)}^q - a_{f(q)}^q \right) \\
 & + \delta_i^{1,q}(t), \text{ le délai maximum subi par } m \text{ dans les nœuds 1 à } q \text{ à cause de l'effet non-préemptif direct.}
 \end{aligned}$$

Puisque, par convention, $p(q) = m$, nous obtenons :

$$W_i^q(t) = \sum_{h=1}^q \left(\sum_{g=f(h)}^{p(h)} C_{\tau(g)}^h \right) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max} - \sum_{h=2}^q \left(a_{p(h-1)}^h - a_{f(h)}^h \right).$$

D'après l'expression de $W_i^q(t)$, l'instant de démarrage au plus tard du paquet m sur le nœud q est maximum lorsque pour tout nœud $h \in]1, q]$, nous avons $a_{p(h-1)}^h - a_{f(h)}^h = 0$, soit $f(h) = p(h-1)$. Nous obtenons alors l'équation suivante.

$$W_i^q(t) = \sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}. \quad (7.1)$$

Dans la suite, nous considérerons la décomposition pire cas illustrée par la figure 7.3, c'est-à-dire lorsque pour tout nœud $h \in]1, q]$, $f(h) = p(h-1)$.

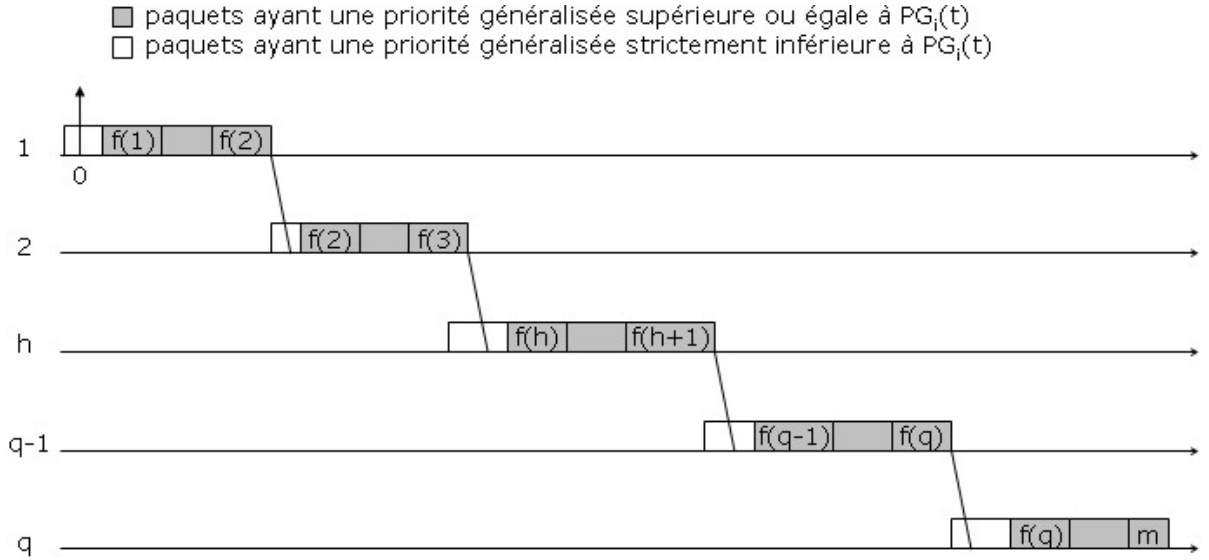


FIG. 7.3: Périodes actives de niveau $PG_i(t)$ considérées dans le pire cas

Nous établissons alors trois propriétés concernant les paquets $\bigcup_{h=1}^q [f(h), f(h+1)]$.

Propriété 7.3.1 *Les paquets appartenant à l'ensemble $\bigcup_{h=1}^q [f(h), f(h+1)]$ ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$.*

Démonstration Par construction, sur tout nœud $h \in [1, q]$, $f(h+1)$ appartient à une période active de niveau $PG_i(t)$ dans laquelle le premier paquet traité de priorité généralisée supérieure ou égale à $PG_i(t)$ est $f(h)$. Par conséquent, sur tout nœud $h \in [1, q]$, les paquets $f(h)$ à $f(h+1)$ ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$. ■

Propriété 7.3.2 *Seuls les paquets $f(h+1)$, $h \in [1, q]$, sont considérés deux fois dans l'ensemble $\bigcup_{h=1}^q [f(h), f(h+1)]$.*

Démonstration Aucun paquet m' traité entre $f(h)$ et $f(h+1)$ sur le nœud h , avec $m' \neq f(h)$ et $m' \neq f(h+1)$, ne peut être traité entre $f(h')$ et $f(h'+1)$ sur le nœud h' , avec $h' \in [1, q]$ et $h' \neq h$. En effet :

- si m' ne visite pas le nœud $h+1$ après le nœud h , alors d'après la propriété 7.3.1, m' ne visitera aucun des nœuds $[h+1, q]$. Par conséquent, nous avons : $\forall h' \in [h+1, q], m' \notin [f(h'), f(h'+1)]$;
- si m' visite le nœud $h+1$ après le nœud h , alors m' arrive avant $f(h+1)$ sur le nœud $h+1$ puisque m' est traité avant $f(h+1)$ sur le nœud h et les liens sont supposés FIFO. L'instant d'arrivée de $f(h+1)$ sur le nœud $h+1$ étant, par construction, un instant oisif de niveau $PG_i(t)$, tout paquet de priorité supérieure ou égale à $PG_i(t)$ est donc traité avant $f(h+1)$ s'il arrive plus tôt sur ce nœud. Le paquet m' ne peut donc pas appartenir à bp^{h+1} . Par récurrence, il vient : $\forall h' \in [h+1, q], m' \notin [f(h'), f(h'+1)]$.

Par conséquent, seuls les paquets $f(h+1)$, $h \in [1, q]$, sont considérés deux fois dans $\bigcup_{h=1}^q [f(h), f(h+1)]$. ■

Propriété 7.3.3 *Un paquet m' d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, n'appartient pas à l'ensemble $\bigcup_{h=1}^q [f(h), f(h+1)]$ si (i) m' arrive sur le nœud $first_j$ avant le paquet $f(first_j)$ ou (ii) m' arrive sur le nœud $last_j$ après l'instant de début d'exécution du paquet m .*

Démonstration D'après l'hypothèse 7.3.1, un paquet m' d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, ne peut gêner le paquet m que sur les nœuds $first_j$ à $last_j$, c'est-à-dire les nœuds communs à \mathcal{L}_j et \mathcal{L}_i . Nous avons alors : $m' \notin \bigcup_{h=1}^{first_j-1} [f(h), f(h+1)]$ et $m' \notin \bigcup_{h=last_j+1}^q [f(h), f(h+1)]$. Nous prouvons donc que si (i) m' arrive sur le nœud $first_j$ avant le paquet $f(first_j)$ ou (ii) m' arrive sur le nœud $last_j$ après que le paquet m ait commencé son exécution sur ce nœud, alors $m' \notin \bigcup_{h=first_j}^{last_j} [f(h), f(h+1)]$.

- 1) Si m' arrive sur le nœud $first_j$ avant le paquet $f(first_j)$, alors il est traité avant puisque l'instant d'arrivée de $f(first_j)$ sur ce nœud est un instant oisif de niveau $PG_i(t)$. De plus, si m' est traité avant $f(first_j)$, alors il est traité avant $f(first_j+1)$. Les liens étant supposés FIFO, m' arrive avant $f(first_j+1)$ sur le nœud $first_j+1$. Il est donc traité avant puisque l'instant d'arrivée de $f(first_j+1)$ sur ce nœud est un instant oisif de niveau $PG_i(t)$. Par récurrence, le paquet m' est toujours traité avant $f(h)$ sur le nœud $h \in [first_j, last_j]$, soit : $m' \notin \bigcup_{h=first_j}^{last_j} [f(h), f(h+1)]$.
- 2) Si m' arrive sur le nœud $last_j$ après que le paquet $f(last_j+1)$ ait commencé son exécution, alors $m' \notin [f(last_j), f(last_j+1)]$. Dans le pire cas, nous avons : $f(last_j+1) = m$. De plus, si m' arrive après $W_i^{last_j}(t)$ sur le nœud $last_j$, alors m' a été traité après m sur le nœud $last_j-1$, les liens étant supposés FIFO. Par récurrence, le paquet m' est toujours traité après m sur le nœud $h \in [first_j, last_j]$. Puisque $[f(h), f(h+1)] \subseteq [f(h), m]$, nous obtenons : $m' \notin \bigcup_{h=first_j}^{last_j} [f(h), f(h+1)]$. ■

RÉCAPITULATIF : Nous avons adopté l'approche par trajectoire pour déterminer $W_i^q(t)$, l'instant de démarrage au plus tard de m sur le nœud q . Ainsi, en étudiant la décomposition illustrée par la figure 7.3, nous sommes arrivés à l'équation 7.1 : $W_i^q(t) = \sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$. Puis, nous avons établi trois propriétés concernant l'ensemble des paquets $[f(1), m] = \bigcup_{h=1}^q [f(h), f(h+1)]$. En effet, il a été montré que :

- les paquets appartenant à $[f(1), m]$ ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$;
- seuls les paquets $f(h+1)$, $h \in [1, q]$, sont considérés deux fois dans l'ensemble $[f(1), m]$;
- un paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, ne peut appartenir à l'ensemble $[f(1), m]$ s'il arrive avant le paquet $f(first_j)$ sur le nœud $first_j$ ou après l'instant de début d'exécution du paquet m sur le nœud $last_j$.

En nous appuyant sur les trois propriétés ci-dessus, nous évaluons maintenant les termes composant l'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q donnée dans l'équation 7.1, à savoir : $\delta_i^{1,q}(t)$ et $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$.

7.3.2 Evaluation des différents délais

Dans cette sous-section, nous nous intéressons à l'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q donnée dans l'équation 7.1, à savoir :

$$W_i^q(t) = \sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}.$$

Plus précisément, nous étudions les deux termes suivants : $\delta_i^{1,q}(t)$ et $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$.

- Le premier terme représente le délai maximum que m peut subir tout au long de sa ligne de diffusion à cause de l'effet non-préemptif direct et est étudié dans le paragraphe 7.3.2.a.
- Le second terme représente le délai maximum que m peut subir tout au long de sa ligne de diffusion à cause de paquets plus prioritaires. En effet, le terme $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$ est égal à $\sum_{h=1}^{q-1} (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) + \sum_{g=f(q)}^{m-1} C_{\tau(g)}^q$. Or d'après la propriété 7.3.1, les paquets appartenant à $[f(1), m-1] = \bigcup_{h=1}^{q-1} [f(h), f(h+1)] \cup [f(q), m-1]$ ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$. Ce terme est étudié dans le paragraphe 7.3.2.b.

7.3.2.a Délai dû à l'effet non-préemptif direct

Le lemme suivant permet de borner le délai maximum que le paquet m peut subir tout au long de sa ligne de diffusion à cause de l'effet non-préemptif direct.

Lemme 7.3.1 Soit FP/DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le délai maximum subi par le paquet de τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est égal à : $\delta_i^{1,q}(t) \leq \sum_{h=1}^q \max \left(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} - 1 \right)$, avec $\max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} = 0$ si $lp_i \cup \overline{sp}_i(t) = \emptyset$.

Démonstration Par définition, aucun des paquets appartenant à $lp_i \cup \overline{sp}_i(t)$ ne peut être traité dans une période active de niveau $PG_i(t)$, à l'exception du premier paquet de la période active (voir le lemme 5.3.1).

Par conséquent, le délai maximum subi par le paquet du flux τ_i généré à l'instant $t \geq -J_i$ dû à l'effet non-préemptif direct est au plus égal à $\max(0; \max_{j \in lp_i \cup sp_i(t)} \{C_j^h\} - 1)$ sur chaque nœud h visité. ■

7.3.2.b Délai dû aux paquets plus prioritaires

Nous procédons en trois étapes pour évaluer le délai maximum que le paquet m peut subir tout au long de sa ligne de diffusion à cause de paquets ayant une priorité généralisée supérieure ou égale à $PG_i(t)$.

- Le lemme 7.3.2 permet d'évaluer ce délai.
- Le lemme 7.3.3 détermine les intervalles de temps dans lesquels les paquets plus prioritaires doivent être générés pour gêner m .
- Le lemme 7.3.4 applique les résultats obtenus dans le lemme 7.3.3 au lemme 7.3.2.

Lemme 7.3.2 *Soit FP/DP* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le délai maximum subi par le paquet de τ_i généré à l'instant $t \geq -J_i$ dû à des paquets plus prioritaires est égal à :*

$$\sum_{h=1}^q \left(\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h \right) - C_i^q \leq \sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow_{\tau(g)}} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}.$$

Démonstration La démonstration faite dans le cas d'une simple ligne de diffusion peut s'appliquer ici (voir le lemme 6.3.3, page 86), en remplaçant pour tout flux τ_j , $j \in hp_i \cup sp_i(t)$, C_j^{slow} par $C_j^{slow_j}$. ■

Nous déterminons maintenant les intervalles de temps dans lesquels les paquets plus prioritaires doivent être générés pour appartenir à l'ensemble $[f(1), m]$. Cela nous permettra d'évaluer le terme $\sum_{g=f(1)}^m C_{\tau(g)}^{slow_{\tau(g)}}$ du lemme précédent.

Lemme 7.3.3 *Soit FP/DP* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Hormis l'effet non-préemptif direct, le paquet de τ_i généré à l'instant $t \geq -J_i$ peut est retardé par :*

- les paquets des flux τ_j , $j \in hp_i$, générés dans l'intervalle $[M_i^{first_j}(t) - S_{max_j}^{first_j} - J_j, W_i^{last_j}(t) - S_{min_j}^{last_j}]$
- les paquets des flux τ_j , $j \in sp_i(t)$, générés dans l'intervalle $[M_i^{first_j}(t) - S_{max_j}^{first_j} - J_j, \min(G_{j,i}(t); W_i^{last_j}(t) - S_{min_j}^{last_j})]$
- les paquets du flux τ_i générés dans l'intervalle $[-J_i, t]$.

Démonstration Un paquet d'un flux quelconque τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$, suivant une ligne de diffusion \mathcal{L}_j telle que $\mathcal{L}_j \cap \mathcal{L}_i \neq \emptyset$ ne peut gêner m que sur les nœuds communs à \mathcal{L}_j et \mathcal{L}_i , soit $[first_j, last_j]$. De plus, d'après la propriété 7.3.3 :

- un paquet de τ_j ne peut gêner m s'il arrive avant le paquet $f(first_j)$ sur le nœud $first_j$, donc s'il est généré avant l'instant d'arrivée du paquet $f(first_j)$ sur le nœud $first_j$ moins le temps maximum nécessaire pour arriver sur le nœud $first_j$, soit : $a_{f(first_j)}^{first_j} - S_{max_j}^{first_j} - J_j$.

Puisque $a_{f(first_j)}^{first_j} \geq a_{f(1)}^1 + \sum_{h=1}^{first_j-1} \min_{k \in hp_i \cup sp_i(t) \cup \{i\}} \{C_k^h\} - (first_j - 1)L_{min}$ et $a_{f(1)}^1 = 0$, un paquet de τ_j ne peut gêner m s'il est généré avant : $M_i^{first_j}(t) - S_{max_j}^{first_j} - J_j$.

- un paquet de τ_j ne peut gêner m s'il arrive après l'instant de début d'exécution du paquet m sur le nœud $last_j$, donc s'il est généré après $W_i^{last_j}(t)$ moins le temps minimum nécessaire pour arriver sur le nœud $last_j$, soit : $W_i^{last_j}(t) - S_{min_j}^{last_j}$.

Par ailleurs, un paquet m' d'un flux quelconque τ_j , $j \in sp_i(t)$, ne peut gêner m que s'il appartient à l'ensemble $[f(1), m]$. Puisque les paquets de cet ensemble ont tous une priorité généralisée supérieure ou égale à $PG_i(t)$, m' doit être généré dans l'intervalle $[M_i^{first_j}(t) - S_{max_j}^{first_j} - J_j, W_i^{last_j}(t) - S_{min_j}^{last_j}]$, mais également avant $G_{j,i}(t)$ pour être plus prioritaire que m . Par conséquent, seuls les paquets de τ_j générés dans l'intervalle $[M_i^{first_j}(t) - S_{max_j}^{first_j} - J_j, \min(G_{j,i}(t); W_i^{last_j}(t) - S_{min_j}^{last_j})]$ peuvent gêner m .

Enfin, seuls les paquets de τ_i générés dans l'intervalle $[-J_i, t]$ peuvent retarder le traitement de m . ■

Grâce au lemme précédent, nous pouvons maximiser le terme $\sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow_{\tau(g)}}$ du lemme 7.3.2 et ainsi évaluer plus précisément le délai maximum que peut subir m tout au long de sa ligne de diffusion à cause de paquets plus prioritaires.

Lemme 7.3.4 *Soit FP/DP* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le délai maximum subi par le paquet de τ_i généré à l'instant $t \geq -J_i$ dû à des paquets plus prioritaires est inférieur ou égal à :*

$$\begin{aligned} & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{last_j}(t) - S_{min_j}^{last_j} - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^{last_j}(t) - S_{min_j}^{last_j}) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}. \end{aligned}$$

Démonstration D'après le lemme 7.3.2, le délai maximum subi par m dû à des paquets plus prioritaires est inférieur ou égal à : $\sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow_{\tau(g)}} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\}$.

Par ailleurs, le terme $\sum_{g=f(1)}^{f(m)} C_{\tau(g)}^{slow_{\tau(g)}}$ est borné par la quantité maximale de travail générée par les paquets définis dans le lemme 7.3.3, soit :

- pour les flux τ_j , $j \in hp_i$: $\sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{last_j}(t) - S_{min_j}^{last_j} - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j + J_i^1}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$;
- pour les flux τ_j , $j \in sp_i(t)$: $\sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^{last_j}(t) - S_{min_j}^{last_j}) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$;
- pour le flux τ_i : $\left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow}$. ■

RÉCAPITULATIF : Après avoir identifié les périodes actives qui affectaient le temps de réponse pire cas de bout-en-bout du paquet m , nous sommes arrivés à l'équation 7.1 donnant l'instant de démarrage au plus tard de m sur le nœud q , à savoir : $W_i^q(t) = \sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$.

Nous avons alors étudié dans cette sous-section les termes $\delta_i^{1,q}(t)$ et $\sum_{h=1}^q (\sum_{g=f(h)}^{f(h+1)} C_{\tau(g)}^h) - C_i^q$. Le premier représente le délai maximum que peut subir le paquet m à cause de l'effet non-préemptif direct. Le second représente le délai maximum que peut subir le paquet m à cause des paquets plus prioritaires. Nous avons ainsi établi les lemmes 7.3.1 et 7.3.4 qui maximisent ces deux termes.

7.3.3 Instant de démarrage au plus tard sur le dernier nœud visité

A partir de l'analyse réalisée dans les deux paragraphes précédents, nous pouvons établir l'instant de démarrage au plus tard du paquet m sur le nœud q .

Propriété 7.3.4 Soit FP/DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . L'instant de démarrage au plus tard sur le nœud q du paquet de τ_i généré à l'instant t est égal à :

$$\begin{aligned}
 W_i^q(t) = & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{last_j}(t) - S_{min_j}^{last_j} - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\
 & + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^{last_j}(t) - S_{min_j}^{last_j}) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\
 & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}, \\
 \text{avec } \delta_i^{1,q}(t) \leq & \sum_{h=1}^q \max \left(0; \max_{j \in lp_i \cup sp_i(t)} \{C_j^h\} - 1 \right).
 \end{aligned}$$

Démonstration En appliquant les lemmes 7.3.1 et 7.3.4 à l'équation 7.1. ■

Remarque 7.3.1 L'expression de l'instant de démarrage au plus tard du paquet m sur le nœud q que nous obtenons est une formule récursive. Pour écrire la suite $\mathcal{W}_i^q(t)$ correspondante, nous distinguons les flux τ_j , $j \in hp_i \cup sp_i(t)$, tels que $last_j = q$. Ainsi, l'expression de $\mathcal{W}_i^q(t)$ est la suivante :

$$\left\{ \begin{array}{l} \mathcal{W}_i^{q(0)}(t) = \sum_{j \in hp_i \cup sp_i(t)} C_j^{slow_j} + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t) \\ \\ \mathcal{W}_i^{q(k+1)}(t) = \sum_{\substack{j \in hp_i \\ last_j = q}} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - S_{min_j^q} - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ \\ + \sum_{\substack{j \in sp_i(t) \\ last_j = q}} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - S_{min_j^q}) - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ \\ + \sum_{\substack{j \in hp_i \\ last_j \neq q}} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{last_j}(t) - S_{min_j^{last_j}} - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ \\ + \sum_{\substack{j \in sp_i(t) \\ last_j \neq q}} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{last_j}(t) - S_{min_j^{last_j}}) - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ \\ + \left(1 + \left\lfloor \frac{t+J_i}{T_i} \right\rfloor\right) \cdot C_i^{slow} + \mathcal{A}_i(t), \end{array} \right.$$

avec $\mathcal{A}_i(t) = \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} - C_i^q + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}$.

Nous montrons alors la convergence de la suite $\mathcal{W}_i^q(t)$ définie ci-dessus afin de prouver l'existence de sa limite : $W_i^q(t)$.

Condition 7.3.1 Pour tout $t \geq -J_i$, si $\sum_{j \in hp_i \cup sp_i(t)} U_j^{slow_j} < 1$, alors la suite $\mathcal{W}_i^q(t)$ est convergente.

Démonstration La suite $\mathcal{W}_i^q(t)$ est une suite non-décroissante car la fonction *partie entière* ($\lfloor \cdot \rfloor$) est non-décroissante. Par ailleurs, pour un t donné, seuls les termes concernant les flux τ_j , $j \in hp_i \cup sp_i(t)$, tels que $last_j = q$ évoluent, les autres termes étant constants. Nous étudions alors la convergence de la suite $\mathcal{W}_i^{q(k)}(t)$ définie ci-dessous, avec $\mathcal{W}_i^{q(0)}(t) = \sum_{j \in hp_i \cup sp_i(t)} C_j^{slow_j}$:

$$\begin{aligned} \mathcal{W}_i^{q(k+1)}(t) &= \sum_{\substack{j \in hp_i \\ last_j = q}} \left(1 + \left\lfloor \frac{\mathcal{W}_i^{q(k)}(t) - S_{min_j^q} - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ &= + \sum_{\substack{j \in sp_i(t) \\ last_j = q}} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); \mathcal{W}_i^{q(k)}(t) - S_{min_j^q}) - M_i^{first_j}(t) + S_{max_j^{first_j}} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}. \end{aligned}$$

La démonstration faite dans le cas d'une simple ligne de diffusion peut s'appliquer ici (voir le lemme 7.3.1, page 117) pour montrer que si $\sum_{j \in hp_i \cup sp_i(t)} U_j^{slow_j} < 1$, alors la suite $\mathcal{W}_i^q(t)$ est bornée supérieurement par $X/(1 - \sum_{j \in hp_i \cup sp_i(t)} U_j^{slow_j})$, avec :

$$X = \sum_{j \in hp_i \cup sp_i(t)} \left(1 + \frac{S_{max_j}^{first_j} + J_j}{T_j} \right) \cdot C_j^{slow_j}.$$

La suite $\mathcal{W}_i^q(t)$ est alors non-décroissante et bornée supérieurement, elle est donc convergente et sa limite vérifie $W_i^q(t)$. ■

Nous pouvons alors calculer le temps de réponse pire cas de bout-en-bout du paquet m , égal à son instant de démarrage au plus tard sur le nœud q plus son temps de traitement sur le nœud q moins son instant de génération, soit : $W_i^q(t) + C_i^q - t$. Nous nous intéressons maintenant au temps de réponse pire cas de bout-en-bout du flux τ_i , obtenu en calculant le maximum des temps de réponse pire cas de ses paquets.

7.3.4 Instants à tester

Le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à : $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. Afin de ne pas tester tous les instants $t \geq -J_i$, nous établissons les deux lemmes suivants.

Lemme 7.3.5 Soit FP/DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Pour tout instant t tel que $\overline{sp}_i(t) = \emptyset$, nous avons $W_i^q(t + \mathcal{B}_i^{slow}) \leq W_i^q(t) + \mathcal{B}_i^{slow}$, avec $\mathcal{B}_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil \mathcal{B}_i^{slow} / T_j \rceil \cdot C_j^{slow_j}$.

Démonstration La démonstration faite dans le cas d'une simple ligne de diffusion peut s'appliquer ici (voir le lemme 6.3.6, page 90), en remplaçant pour tout flux τ_j , $j \in hp_i \cup sp_i(t)$, C_j^{slow} par $C_j^{slow_j}$. ■

Lemme 7.3.6 Soit FP/DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de τ_i est atteint par un paquet généré à un instant t tel que $G_{j,i}(t) = -J_l + k_l \cdot T_l + M_i^{first_l}(t) - S_{max_l}^{first_l}$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.

Démonstration La démonstration faite dans le cas d'une simple ligne de diffusion peut s'appliquer ici (voir le lemme 6.3.7, page 91), en tenant compte des termes supplémentaires suivants pour les flux τ_j , $j \in hp_i \cup sp_i(t)$: $M_i^{first_j}(t) - S_{max_j}^{first_j}$. ■

RÉCAPITULATIF : Les deux lemmes précédents ont permis de limiter, puis de réduire le nombre d'instants à tester pour le calcul du temps de réponse pire cas de bout-en-bout du flux τ_i . En effet, la définition du temps de réponse pire cas de bout-en-bout du flux τ_i est la suivante : $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. Les lemmes 7.3.5 et 7.3.6 limitent les instants t à tester à ceux vérifiant les deux conditions suivantes :

- $-J_i \leq t < \bar{t}_i^\emptyset + B_i^{slow}$, avec \bar{t}_i^\emptyset le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$
 et $B_i^{slow} = \sum_{j \in hp_i \cup sp_i(t) \cup \{i\}} \lceil B_i^{slow} / T_j \rceil \cdot C_j^{slow_j}$;
- $G_{j,i}(t) = -J_l + k_l \cdot T_l + M_i^{first_l}(t) - S_{max_l}^{first_l}$, où j et l appartiennent à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.

7.3.5 Temps de réponse pire cas de bout-en-bout

La propriété suivante précise le temps de réponse pire cas de bout-en-bout du flux τ_i .

Propriété 7.3.5 Soit FP/DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$\begin{aligned} W_i^q(t) = & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{lastj}(t) - S_{min_j}^{lastj} - M_i^{firstj}(t) + S_{max_j}^{firstj} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^{lastj}(t) - S_{min_j}^{lastj}) - M_i^{firstj}(t) + S_{max_j}^{firstj} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}, \end{aligned}$$

$$\delta_i^{1,q}(t) \leq \sum_{h=1}^q \max \left(0; \max_{j \in lp_i \cup \overline{sp}_i(t)} \{C_j^h\} - 1 \right)$$

et \mathcal{S}_i l'ensemble des instants t tels que :

- $-J_i \leq t < \bar{t}_i^\emptyset + B_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l + M_i^{firstl}(t) - S_{max_l}^{firstl}$, $k_l \in \mathbb{N}$.

Démonstration L'expression de $W_i^q(t)$ est donnée par la propriété 7.3.4, page 116. Son expression est une formule récursive pour laquelle il existe une solution si $\sum_{j \in hp_i \cup sp_i(t)} U_j^{slow_j} < 1$ (voir la condition 7.3.1).

De plus, le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à $R_i^{1,q} = \max_{t \geq -J_i} \{W_i^q(t) - t\} + C_i^q$. D'après les lemmes 7.3.5 et 7.3.6, seuls les instants t appartenant à l'ensemble \mathcal{S}_i sont à considérer. ■

Remarque 7.3.2 Dans la propriété du temps de réponse pire cas de bout-en-bout d'un flux donnée ci-dessus, les termes $M_i^{firstj}(t)$ et $S_{min_j}^{lastj}$ peuvent être calculés par une simple somme. En revanche, il convient d'apporter quelques précisions sur la détermination du terme $S_{max_j}^{firstj}$, $j \in hp_i \cup sp_i(t)$. Ce terme représente le temps maximum mis par un paquet du flux τ_j depuis son instant de génération pour arriver sur le nœud $firstj$, premier nœud de la ligne de diffusion \mathcal{L}_i visité par τ_j .

Pour déterminer ce terme, nous verrons dans le chapitre 9 (section 9.2.4, page 151) une solution basée sur une décomposition de l'échéance de bout-en-bout du flux τ_j en échéances intermédiaires. Pour être accepté par le contrôle d'admission, le flux τ_j devra satisfaire son échéance de bout-en-bout mais également ses échéances intermédiaires dans les différents nœuds visités, ainsi que les flux croisés. Le terme $S_{max_j}^{first_j}$ sera alors borné par la somme des échéances intermédiaires du flux τ_j depuis son nœud source jusqu'au nœud $first_j$, plus les délais réseau.

Remarque 7.3.3 Nous pouvons également remarquer qu'avec un ordonnancement de type FP/DP*, les résultats obtenus pour un flux quelconque τ_i en environnement distribué dépendent principalement d'un calcul sur les nœuds les plus lents de la ligne \mathcal{L}_i visités par les flux τ_j , $j \in hp_i \cup sp_i(t) \cup \{i\}$. Ce calcul est similaire à celui effectué en contexte monoprocesseur. Nous n'avons plus cette similitude avec un ordonnancement de type FP/DP, c'est-à-dire lorsque la priorité dynamique d'un paquet dépend du nœud visité².

7.4 GÉNÉRALISATION

Il est possible de lever l'hypothèse 7.3.1 afin d'étendre la propriété 7.3.5 au cas général. Pour cela, il suffit de considérer un flux croisant la ligne de diffusion \mathcal{L}_i après l'avoir quittée comme un nouveau flux et de procéder par itération jusqu'à satisfaire l'hypothèse 7.3.1. Nous pouvons appliquer la propriété 7.3.5 en considérant tous les flux.

7.5 CAS PARTICULIERS

7.5.1 Les flux suivent tous une même ligne de diffusion

Il est intéressant de souligner que dans le cas où les flux suivent tous la même ligne de diffusion, nous retrouvons les résultats établis dans le chapitre 5. En effet, dans ce cas, nous avons pour tout flux τ_j : $first_j = 1$ et $last_j = q$, ce qui conduit à :

- $M_i^{first_j}(t) = M_i^1(t) = 0$;
- $S_{max_j}^{first_j} = S_{max_j}^1 = 0$;
- $S_{min_j}^{last_j} = S_{min_j}^q$.

7.5.2 Les flux ont tous une priorité fixe différente

Lorsque tous les flux ont une priorité fixe différente, le traitement DP* n'intervient pas dans l'ordonnancement FP/DP*, puisque DP* permet de départager des paquets ayant la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme FP.

Si tous les flux ont une priorité fixe différente, alors pour tout flux τ_i : $sp_i = \emptyset$. En appliquant ce résultat à la propriété 6.3.3, nous obtenons la propriété suivante.

²Les ordonnancements de type FP/DP et FP/DP* sont présentés dans la section 2.2.3, page 17.

Propriété 7.5.1 Soit FP l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{last_j}(t) - S_{min_j}^{last_j} - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$$

$$+ \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max},$$

$$\delta_i^{1,q}(t) \leq \sum_{h=1}^q \max \left(0; \max_{j \in lp_i} \{C_j^h\} - 1 \right)$$

et $\mathcal{S}_i = \{t, t = -J_i + k \cdot T_i < -J_i + B_i^{slow}, k \in \mathbb{N}\}$.

Démonstration Nous obtenons l'expression de $W_i^q(t)$ ci-dessus en appliquant $sp_i = \emptyset$ à la propriété 7.3.5. Par ailleurs, les instants t à tester doivent satisfaire la condition suivante : $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l + M_i^{first_l}(t) - S_{max_l}^{first_l}$, $k_l \in \mathbb{N}$. Puisque $sp_i = \emptyset$ et $G_{i,i}(t) = t$, seuls les instants de la forme $-J_i + k \cdot T_i$ doivent être considérés. De plus, le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$ est l'instant $-J_i$. ■

7.5.3 Les flux partagent tous la même priorité fixe

Lorsque tous les flux partagent la même priorité fixe, le traitement FP n'intervient pas dans l'ordonnancement FP/DP^* . En effet, seul l'ordonnancement permettant de départager les paquets ayant la même priorité fixe est sollicité. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant un ordonnancement DP^* .

Si tous les flux ont la même priorité fixe, alors pour tout flux τ_i : $hp_i = lp_i = \emptyset$. Par ailleurs, nous rappelons que $sp_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i susceptibles de générer des paquets de priorités dynamiques supérieures ou égales à celle de m .

De même, $\overline{sp}_i(t)$ est l'ensemble des flux de même priorité fixe que τ_i ne pouvant pas générer de paquets avec une priorité dynamique supérieure ou égale à celle de m , mais susceptibles de générer un paquet pouvant gêner m à cause de la non-préemption.

En appliquant ces résultats à la propriété 6.3.3, nous obtenons la propriété suivante, donnant le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont traités dans chacun des nœuds de la ligne de diffusion suivant un ordonnancement de type DP^* .

Propriété 7.5.2 Soit DP^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(G_{j,i}(t); W_i^{lastj}(t) - S_{min_j}^{lastj}) - M_i^{firstj}(t) + S_{max_j}^{firstj} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$$

$$+ \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max},$$

$$\delta_i^{1,q}(t) \leq \sum_{h=1}^q \max \left(0; \max_{j \in \overline{sp}_i(t)} \{C_j^h\} - 1 \right)$$

et \mathcal{S}_i l'ensemble des instants t tels que :

- $-J_i \leq t < \bar{t}_i^\emptyset + \mathcal{B}_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists j$ et l appartenant à $sp_i(t) \cup \{i\}$ tels que $G_{j,i}(t) = -J_l + k_l \cdot T_l + M_i^{firstl}(t) - S_{max_l}^{firstl}$, $k_l \in \mathbb{N}$.

Démonstration D'après la propriété 7.3.5, avec $hp_i = lp_i = \emptyset$. ■

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux sont ordonnancés FP/DP* dans chacun des nœuds du réseau (section 7.3), nous avons considéré trois cas, à savoir (i) les flux suivent tous la même ligne de diffusion, (ii) les flux ont tous une priorité fixe différente et (iii) les flux ont tous la même priorité fixe. Ces trois cas ont permis de comparer les résultats obtenus avec ceux établis dans le chapitre précédent, puis de déterminer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux sont ordonnancés respectivement FP et DP* dans les nœuds du réseau.

Nous nous intéressons maintenant à deux ordonnancements DP* particuliers, à savoir FIFO* (section 7.6) et EDF* (section 7.7).

7.6 ORDONNANCEMENT FP/FIFO*

Dans cette section, nous montrons comment calculer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i dans un réseau lorsque les flux sont ordonnancés FP/FIFO*. Nous utilisons pour cela les résultats établis dans la section 7.3 et les appliquons à l'ordonnancement FP/FIFO*. Nous considérons un flux quelconque τ_i . Soit m le paquet de τ_i généré à l'instant t . Nous avons montré dans la section 5.6, page 66, les deux points suivants :

- $\forall t \geq -J_i, sp_i(t) = \{j \in sp_i, -J_j \leq t\}$;
- $\forall j \in sp_i(t), G_{j,i}(t) = t$.

Concernant $\overline{sp}_i(t)$, c'est-à-dire les flux de même priorité fixe que τ_i mais n'appartenant pas à $sp_i(t) \cup \{i\}$ et pouvant gêner le paquet m de τ_i généré à l'instant t à cause de l'effet non-préemptif, nous avons :

- $\forall t \geq -J_i, \overline{sp}_i(t) = \{j \in sp_i, -J_j > t, \text{ avec } \tau_j \text{ généré sur } h \neq 1\}$;

En effet, cet ensemble était nul lorsque les flux suivaient tous la même ligne de diffusion puisque dans ce cas, l'ordonnancement FIFO* est équivalent à l'ordonnancement FIFO. Lorsque les flux suivent des lignes de diffusion différentes, il est possible que le paquet d'un flux $\tau_j, j \in sp_i$, généré après m sur un nœud $h \neq 1$ arrive avant m dans un nœud de la ligne \mathcal{L}_i .

Nous avons établi page 119 la propriété 7.3.5 donnant le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i pour un ordonnancement de type FP/DP*. En appliquant les résultats ci-dessus à cette propriété, nous définissons le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont ordonnancés FP/FIFO*.

7.6.1 Temps de réponse pire cas de bout-en-bout

La propriété suivante établit le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les nœuds du réseau ordonnancent les flux selon FP/FIFO*, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs instants de génération.

Propriété 7.6.1 Soit FP/FIFO* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$\begin{aligned} W_i^q(t) = & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{lastj}(t) - S_{min_j}^{lastj} - M_i^{firstj}(t) + S_{max_j}^{firstj} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t; W_i^{lastj}(t) - S_{min_j}^{lastj}) - M_i^{firstj}(t) + S_{max_j}^{firstj} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\ & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max} \end{aligned}$$

et \mathcal{S}_i l'ensemble des instants t tels que :

$$-J_i \leq t = -J_j + k_j \cdot T_j + M_i^{firstj}(t) - S_{max_j}^{firstj} < B_i^{slow}, j \in sp_i(t) \cup \{i\}, k_j \in \mathbb{N}.$$

Démonstration D'après la propriété 7.3.5, avec $\bar{t}_i^\emptyset = 0, sp_i(t) = \{j \in sp_i, -J_j \leq t\}$ et pour tout flux τ_j tel que $j \in sp_i(t), G_{j,i}(t) = t$. ■

7.6.2 Ordonnancement FIFO*

Nous avons vu dans la section 7.5 que le traitement FP n'intervient pas dans l'ordonnancement FP/FIFO* lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme FIFO* en distribué. Si tous les flux ont la même priorité fixe, alors pour tout flux τ_i : $hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 7.6.1, nous obtenons la propriété suivante.

Propriété 7.6.2 *Soit FIFO* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :*

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = + \sum_{j \neq i, -J_j \leq t} \left(1 + \left\lfloor \frac{\min(t; W_i^{last_j}(t) - S_{min_j}^{last_j}) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$$

$$+ \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j \leq t} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max},$$

$$\text{et } \mathcal{S}_i = \left\{ t, -J_i \leq t = -J_j + k_j \cdot T_j + M_i^{first_j}(t) - S_{max_j}^{first_j} < B_i^{slow}, -J_j \leq t, k_j \in \mathbb{N} \right\}.$$

Démonstration D'après la propriété 6.5.1, avec $hp_i = lp_i = \emptyset$. ■

Remarque 7.6.1 Nous venons d'établir de nouveaux résultats pour l'ordonnancement FIFO* en environnement distribué. Afin de réduire le temps de calcul du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , il est possible de considérer la relation suivante dans la propriété 7.6.2 : pour tout $j \in sp_i(t)$, $\min(t; W_i^{last_j}(t) - S_{min_j}^{last_j}) \leq t$. Ainsi, nous obtenons $R_i^{1,q}$ inférieur ou égal à :

$$\max_{t \in \mathcal{S}_i} \left\{ \sum_{j, -J_j \leq t} \left(1 + \left\lfloor \frac{t - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j \leq t} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max} \right\}.$$

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les nœuds ordonnancement les flux selon FP/DP* (section 7.3), nous avons appliqué nos résultats à un ordonnancement DP* particulier : FIFO*. Le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i en environnement distribué pour l'ordonnancement FP/FIFO* a ainsi été établi.

Nous avons ensuite considéré le cas où les flux ont tous la même priorité fixe. Cela nous a permis d'établir de nouveaux résultats pour l'ordonnancement FIFO*.

Nous verrons un exemple de calcul du temps de réponse pire cas avec FP/FIFO* dans le chapitre 9.

7.7 ORDONNANCEMENT FP/EDF*

Dans cette section, nous montrons comment calculer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i en environnement distribué pour l'ordonnancement FP/EDF*. Nous utilisons pour cela les résultats établis dans la section 7.3 et les appliquons à l'ordonnancement FP/EDF*.

Nous considérons un flux τ_i . Soit m le paquet de τ_i généré à l'instant t . Comme cela a été présenté dans la section 7.7 (page 125), la priorité dynamique du paquet de τ_i généré à l'instant t est égale à : $t + D'_i$, avec D'_i l'échéance relative du flux τ_i sur son nœud source. Il a également été montré les points suivants :

- $\forall t \geq -J_i, sp_i(t) = \{j \in sp_i, P_j(-J_j) \geq P_i(t)\} = \{j \in sp_i, -J_j + D'_j \leq t + D'_i\}$;
- $\forall t \geq -J_i, \overline{sp}_i(t) = \{j \in sp_i, P_j(-J_j) < P_i(t)\} = \{j \in sp_i, -J_j + D'_j > t + D'_i\}$;
- $\forall j \in sp_i(t), G_{j,i}(t) = t + D'_i - D'_j$.

Nous avons établi page 119 la propriété 7.3.5 donnant le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i en environnement distribué pour l'ordonnancement FP/DP*. En appliquant les résultats ci-dessus à cette propriété, nous définissons le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les flux sont ordonnancés FP/EDF*.

7.7.1 Temps de réponse pire cas

La propriété suivante établit le temps de réponse pire cas de bout-en-bout du flux τ_i lorsque les nœuds du réseau ordonnancement les flux selon FP/EDF*, c'est-à-dire lorsque les paquets de même priorité fixe sont traités dans l'ordre de leurs échéances absolues.

Propriété 7.7.1 Soit FP/EDF^* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q$, avec :

$$\begin{aligned}
 W_i^q(t) = & \sum_{j \in hp_i} \left(1 + \left\lfloor \frac{W_i^{last_j}(t) - S_{min_j}^{last_j} - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\
 & + \sum_{j \in sp_i(t)} \left(1 + \left\lfloor \frac{\min(t + D'_i - D'_j; W_i^{last_j}(t) - S_{min_j}^{last_j}) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j} \\
 & + \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{h=1}^q \max_{j \in hp_i \cup sp_i(t) \cup \{i\}} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max}, \\
 \mathcal{S}_i = & \{t, -J_i \leq t = -J_l + k_l \cdot T_l - D'_i + D'_j + M_i^{first_l}(t) - S_{max_l}^{first_l} < \max_{k \in sp_i} \{D'_k - J_k\} - D'_i + \mathcal{B}_i^{slow}, \\
 & \text{avec } j \text{ et } l \text{ appartenant à } sp_i(t) \cup \{i\} \text{ et } k_l \in \mathbb{N}\}.
 \end{aligned}$$

Démonstration D'après la propriété 7.3.5, en appliquant $sp_i(t) = \{j \in sp_i, -J_j + D'_j \leq t + D'_i\}$, $\overline{sp}_i(t) = \{j \in sp_i, -J_j + D'_j > t + D'_i\}$ et $\forall j \in sp_i(t), G_{j,i}(t) = t + D'_i - D'_j$, nous trouvons l'expression de $W_i^q(t)$ lorsque les flux sont ordonnancés FP/EDF^* . Concernant les instants à tester, seuls ceux respectant les deux conditions suivantes doivent être considérés :

- $-J_i \leq t < \bar{t}_i^\emptyset + \mathcal{B}_i^{slow}$, où \bar{t}_i^\emptyset représente le plus petit instant t tel que $\overline{sp}_i(t) = \emptyset$;
- $\exists (j, l) \in sp_i(t) \cup \{i\}, t + D'_i - D'_j = -J_l + k_l \cdot T_l + M_i^{first_l}(t) - S_{max_l}^{first_l}, k_l \in \mathbb{N}$.

Or pour tout flux τ_j tel que $j \in sp_i$, si $t \geq \max_{k \in sp_i} \{-J_k + D'_k\} - D'_i$, alors nous obtenons : $t + D'_i \geq \max_{k \in sp_i} \{-J_k + D'_k\} \geq -J_j + D'_j$. Par conséquent, $P_j(-J_j) \geq P_i(t)$, soit $j \in sp_i(t)$, d'où $\overline{sp}_i(t) = \emptyset$. L'instant \bar{t}_i^\emptyset est donc égal à : $\max_{k \in sp_i(t)} \{D'_k - J_k\} - D'_i$. Les instants à tester sont alors ceux de l'ensemble \mathcal{S}_i défini dans la propriété 7.7.1. ■

7.7.2 Ordonnancement EDF^*

Nous avons vu dans la section 7.5 que le traitement FP n'intervient pas dans l'ordonnancement FP/EDF^* lorsque tous les flux partagent la même priorité fixe. Nous établissons donc, dans ce cas, des résultats pour un ensemble de flux sporadiques ordonnancés suivant l'algorithme EDF^* . Si tous les flux ont la même priorité fixe, alors pour tout flux $\tau_i : hp_i = lp_i = \emptyset$. En appliquant ce résultat à la propriété 6.6.1, nous obtenons la propriété suivante.

Propriété 7.7.2 Soit EDF* l'ordonnancement appliqué dans chacun des nœuds du réseau. Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Le temps de réponse pire cas de bout-en-bout de τ_i est égal à :

$$R_i^{1,q} = \max_{t \in \mathcal{S}_i} \{W_i^q(t) - t\} + C_i^q, \text{ avec :}$$

$$W_i^q(t) = \sum_{j \neq i, -J_j + D'_j - D'_i \leq t} \left(1 + \left\lfloor \frac{\min(t + D'_i - D'_j; W_i^{last j}(t) - S_{min_j}^{last j}) - M_i^{first j}(t) + S_{max_j}^{first j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow_j}$$

$$+ \left(1 + \left\lfloor \frac{t + J_i}{T_i} \right\rfloor \right) \cdot C_i^{slow} - C_i^q + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j + D'_j - D'_i \leq t} \{C_j^h\} + \delta_i^{1,q}(t) + (q-1) \cdot L_{max},$$

$$\mathcal{S}_i = \{t, -J_i \leq t = -J_l + k_l \cdot T_l - D'_i + D'_j + M_i^{first l}(t) - S_{max_l}^{first l} < \max_{k \in sp_i} \{D'_k - J_k\} - D'_i + \mathcal{B}_i^{slow},$$

avec j et l appartenant à $sp_i(t) \cup \{i\}$ et $k_l \in \mathbb{N}$.

Démonstration D'après la propriété 7.6.1, avec $hp_i = lp_i = \emptyset$. ■

Remarque 7.7.1 Nous venons d'établir de nouveaux résultats pour l'ordonnancement EDF* en environnement distribué. Afin de réduire le temps de calcul du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i , il est possible de considérer la relation suivante dans la propriété 7.7.2 : pour tout $j \in sp_i(t)$, $\min(t + D'_i - D'_j; W_i^q(t) - S_{min_j}^q) \leq t + D'_i - D'_j$. Ainsi, nous obtenons $R_i^{1,q}$ inférieur ou égal à :

$$\max_{t \in \mathcal{S}_i} \left\{ \sum_{j, -J_j + D'_j - D'_i \leq t} \left(1 + \left\lfloor \frac{t + D'_i - D'_j - M_i^{first j}(t) + S_{max_j}^{first j} + J_j}{T_j} \right\rfloor^+ \right) \cdot C_j^{slow} \right.$$

$$\left. + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{j, -J_j + D'_j - D'_i \leq t} \{C_j^h\} + (q-1) \cdot L_{max} - t \right\}.$$

RÉCAPITULATIF : Après avoir déterminé le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les nœuds ordonnancement les flux selon FP/DP* (section 7.3), nous avons appliqué nos résultats à un ordonnancement DP* particulier : EDF*. Le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i en environnement distribué pour l'ordonnancement FP/EDF* a ainsi été établi. Nous avons ensuite considéré le cas où les flux ont tous la même priorité fixe. Cela nous a permis d'établir de nouveaux résultats pour l'ordonnancement EDF*.

7.8 CONCLUSION

Nous avons déterminé dans ce chapitre le temps de réponse pire cas d'un flux sporadique quelconque τ_i en environnement distribué lorsque les nœuds du réseau ordonnancent les flux selon FP/DP*. Pour cela, nous avons procédé en plusieurs étapes.

- Tout d'abord, nous nous sommes intéressés à un paquet quelconque m du flux τ_i , généré à l'instant t et avons déterminé son instant de démarrage au plus tard sur le dernier nœud (propriété 7.3.4, page 116). Pour ce faire, nous avons utilisé l'approche par trajectoire, consistant à identifier sur chacun des nœuds visités la période active et les paquets affectant le temps de réponse de bout-en-bout de m .
- Nous sommes parvenu à une expression récursive de l'instant $W_i^q(t)$, pour laquelle nous avons prouvé l'existence d'une solution lorsque $\sum_{j \in hp_i \cup sp_i(t)} U_j^{slow_j} < 1$ (condition 7.3.1, page 117).
- Nous avons alors pu déterminer le temps de réponse pire cas de bout-en-bout de τ_i , égal au maximum des temps de réponse pire cas de bout-en-bout de ses paquets, soit : $R_i = \max_t \{W_i^q(t) - t\} + C_i^q$, avec C_i^q le temps de traitement maximum d'un paquet de τ_i sur le nœud q . Afin de limiter le nombre d'instant t à tester pour le calcul du temps de réponse pire cas de τ_i , nous avons établi deux lemmes (page 118).

Nous sommes ainsi arrivés à l'expression du temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i lorsque l'ordonnancement FP/DP* est utilisé dans chacun des nœuds (propriété 7.3.5, page 119).

Nous avons ensuite considérés trois cas particuliers afin de (i) comparer nos résultats avec ceux établis dans le cas d'une ligne de diffusion unique et (ii) déterminer le temps de réponse pire cas d'un flux lorsque les paquets sont ordonnancés respectivement FP et DP* sur chacun des nœuds visités. Puis nous avons appliqué ces résultats à deux ordonnancements DP* particuliers : FIFO* et EDF*.

Cela a permis d'établir, dans le cas distribué, le temps de réponse pire cas de bout-en-bout d'un flux pour les ordonnancements FP/FIFO* (propriété 7.6.1, page 123) et FP/EDF* (propriété 7.7.1, page 126), mais aussi pour les ordonnancements FIFO* (propriété 7.6.2, page 124) et EDF* (propriété 7.7.2, page 127).

Troisième partie

Extensions et applications

CHAPITRE 8

Impact de la remise en forme du trafic

8.1	Introduction	132
8.2	Notations	133
8.3	Temps de réponse et gigue de bout-en-bout	133
8.4	Sans remise en forme	134
8.5	Avec remise en forme par flux	134
8.5.1	Remise en forme par annulation de gigue	135
8.5.2	Remise en forme par seau à jetons	136
8.6	Exemple	141
8.7	Etude comparative avec l'ordonnancement FIFO	142
8.7.1	Sans remise en forme	142
8.7.2	Avec remise en forme par annulation de gigue	143
8.7.3	Avec remise en forme par seau à jetons	144
8.8	Discussion	145
8.8.1	Annulation de gigue ou seau à jetons ?	145
8.8.2	Avec ou sans remise en forme ?	145
8.9	Conclusion	146

8.1 INTRODUCTION

Dans ce chapitre, nous nous intéressons à la distorsion d'un flux quelconque due à la variation des délais réseau et des durées de séjour dans chacun des nœuds visités. Nous supposons que chacun des nœuds du réseau a la structure présentée à la figure 8.1, composée de :

- un *classificateur*, qui insère les paquets dans les files d'attente de l'ordonnanceur en fonction de leurs priorités ;
- un *composant de remise en forme (shaper)*, permettant d'éviter ou de limiter la distorsion des flux ;
- un *ordonnanceur*, qui gère un ensemble ¹ de files d'attente et sélectionne parmi les paquets en attente, selon un algorithme prédéfini, le paquet qui sera traité par le processeur ;
- un *processeur*, qui transmet les paquets sur le lien de sortie du nœud.

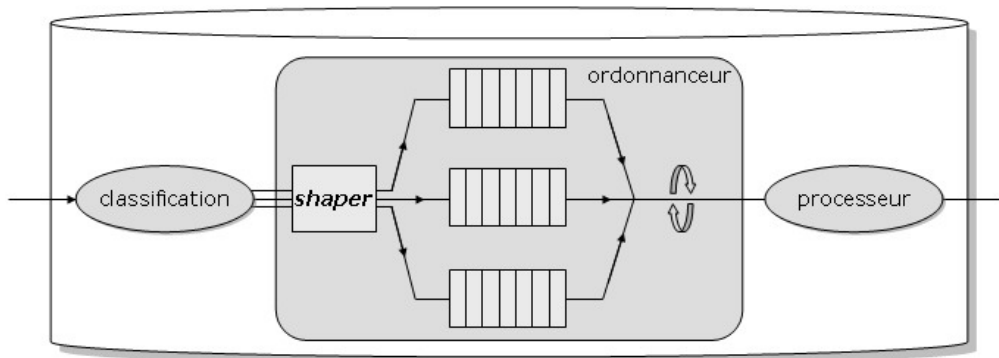


FIG. 8.1: Structure d'un nœud du réseau considéré

Ainsi, lorsqu'un nouveau paquet arrive dans un nœud, il est traité dans le composant de remise en forme (*shaper*) avant d'entrer dans l'ordonnanceur du nœud. Nous considérons trois cas : (i) les *shapers* n'appliquent aucune remise en forme, (ii) les *shapers* appliquent la technique de l'annulation de gigue et (iii) les *shapers* appliquent la technique du seau à jetons.

L'objectif est de comparer le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque avec et sans remise en forme du trafic. Nous rappelons dans la section 8.4 qu'en l'absence de remise en forme du trafic, celui-ci est continuellement déformé et nous montrons les conséquences d'une telle distorsion sur un flux, en termes de gigue et de rafales de paquets. Dans la section 8.5, nous décrivons les techniques de l'annulation de gigue et du seau à jetons et présentons des résultats sur le temps de réponse et la gigue de bout-en-bout d'un flux lorsque l'une de ces deux techniques est utilisée. Ces résultats sont établis indépendamment de l'ordonnancement appliqué par les nœuds.

Nous montrons ensuite dans la section 8.7 comment calculer des bornes sur le temps de réponse et la gigue de bout-en-bout d'un flux en se basant sur une analyse pire cas de l'ordonnancement FIFO, avec et sans remise en forme. Enfin, nous déterminons dans la section 8.8 les cas où l'une des trois techniques étudiées est préférable.

Nous retrouvons dans ce chapitre les résultats établis dans [73, 74].

¹Il peut n'y avoir qu'une seule file d'attente.

8.2 NOTATIONS

Nous considérons l'ensemble $\tau = \{\tau_1, \dots, \tau_n\}$ des n flux parcourant le réseau. Chaque flux τ_i suit une ligne de diffusion \mathcal{L}_i , c'est-à-dire une séquence de nœuds dans le réseau. Dans cette section, nous caractérisons un flux sporadique τ_i par :

- T_i , sa période, ou plus exactement le délai minimum d'interarrivée entre deux paquets successifs de τ_i ;
- L_i , la longueur d'un paquet quelconque de τ_i ;
- C_i^h , le temps de traitement sur le nœud h de tout paquet du flux τ_i . Cette quantité peut être déduite de la longueur du paquet, du type du nœud h (nœud en bordure ou au cœur du réseau) et de la capacité de traitement de ce nœud.

Nous utiliserons également les notations suivantes (voir figure 8.2, page 134).

$shap(h)$	le <i>shaper</i> du nœud h ;
$ordo(h)$	l'ordonnanceur du nœud h ;
e	un élément tel qu'un nœud, un <i>shaper</i> , un ordonnanceur ou une ligne de diffusion ;
a_m^e	l'instant d'arrivée du paquet m dans l'élément e ;
d_m^e	l'instant de départ du paquet m de l'élément e ;
R_m^e	le temps de réponse du paquet m dans l'élément e ;
$J_{in_i}^e$	la gigue pire cas du flux τ_i en entrée de l'élément e ;
$J_{out_i}^e$	la gigue pire cas du flux τ_i en sortie de l'élément e ;
S_m^e	le temps mis par le paquet m pour atteindre l'élément e depuis son instant de génération ;
$L_m^{h,h+1}$	le délai réseau du paquet m entre les nœuds h et $h + 1$;

Nous noterons également $R_{min_i}^e$ (resp. $R_{max_i}^e$) le temps de réponse minimum (resp. maximum) subi par les paquets du flux τ_i dans l'élément e : $\forall m$ de τ_i , $R_{min_i}^e \leq R_m^e \leq R_{max_i}^e$. De la même manière, nous noterons $S_{min_i}^e$ (resp. $S_{max_i}^e$) le temps minimum (resp. maximum) mis par les paquets de τ_i pour atteindre l'élément e depuis leurs instants de génération : $\forall m$ de τ_i , $S_{min_i}^e \leq S_m^e \leq S_{max_i}^e$. Enfin, d'après le modèle de réseau adopté (voir section 2.2.1, page 15), il vient : $\forall m$ de τ_i , $L_{min} \leq L_m^{h,h+1} \leq L_{max}$.

8.3 TEMPS DE RÉPONSE ET GIGUE DE BOUT-EN-BOUT

Considérons un flux quelconque τ_i , $i \in [1, n]$, suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds. Le temps de réponse de bout-en-bout d'un paquet quelconque m de τ_i dépend de ses durées de séjour dans chacun des nœuds visités et des délais réseau. La durée de séjour du paquet m dans un nœud se décompose en deux parties : (i) le temps d'attente dans le *shaper* et (ii) le temps d'attente dans l'ordonnanceur. Par conséquent, le temps de réponse de bout-en-bout du paquet m peut être décomposé de la manière suivante : (i) le délai subi dans les *shapers* visités, (ii) le délai subi dans les ordonnanceurs visités et (iii) les délais réseau. Nous obtenons ainsi :

$$R_m^{\mathcal{L}_i} = \sum_{h=1}^q R_m^{shap(h)} + \sum_{h=1}^q R_m^{ordo(h)} + \sum_{h=1}^{q-1} L_m^{h,h+1}.$$

Maximiser indépendamment chacun de ces trois termes conduit généralement à une borne pessimiste. Nous pouvons obtenir de meilleurs résultats en adoptant l'approche par trajectoire.

Par ailleurs, les paquets du flux τ_i connaissent des temps de réponse de bout-en-bout différents. La gigue de bout-en-bout de τ_i est la différence entre les temps de réponse maximum et minimum connus par ses paquets. Par conséquent, sur la ligne de diffusion \mathcal{L}_i suivie par τ_i , nous avons : $J_{out_i}^{\mathcal{L}_i} = R_{max_i}^{\mathcal{L}_i} - R_{min_i}^{\mathcal{L}_i}$.

8.4 SANS REMISE EN FORME

Les paquets d'un flux quelconque τ_i connaissent des délais réseau et des durées de séjour variables. De plus, la différence entre les temps de réponse maximum et minimum connus par les paquets de τ_i augmente avec le nombre de nœuds visités. La gigue du flux croît donc de nœud en nœud. Une des conséquences de cette accumulation de gigue est l'apparition de rafales de paquets (voir figure 8.2), de plus en plus importantes le long de la ligne de diffusion. En effet, puisque les flux arrivent sans gigue sur le nœud source, le délai minimum d'inter-arrivée entre deux paquets successifs de τ_i est égal à T_i et la gigue de τ_i en sortie du nœud source est égale à $J_{out_i}^1 = R_{max_i}^1 - R_{min_i}^1$. La gigue de τ_i en entrée du second nœud visité est égale à la gigue de τ_i en sortie du nœud source, plus la variation maximum du délai réseau entre ces deux nœuds, à savoir : $J_{in_i}^2 = R_{max_i}^1 - R_{min_i}^1 + L_{max} - L_{min}$, et ainsi de suite.

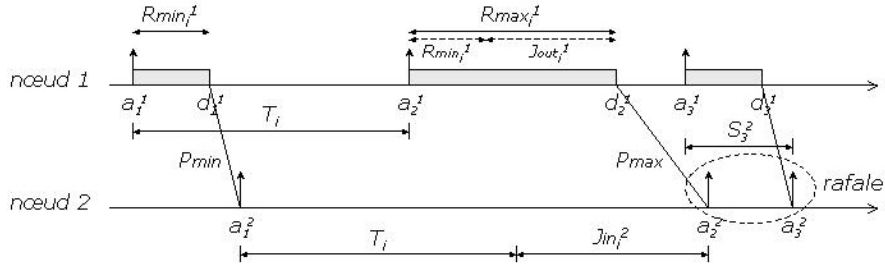


FIG. 8.2: Distorsion d'un flux τ_i sur les deux premiers nœuds visités

Par conséquent, dans tout nœud h excepté le nœud source, les paquets de τ_i arrivent sans nécessairement respecter le délai minimum d'inter-arrivée T_i puisque le délai d'inter-arrivée entre deux paquets successifs de τ_i sur le nœud h peut être égal à $\max(0; T_i - J_{in_i}^h) \leq T_i$. Des rafales de paquets peuvent alors survenir.

De plus, si nous nous intéressons, par exemple, à un système multimédia de vidéo à la demande, cette distorsion pose le problème de famine de données. En effet, le délai d'inter-arrivée entre deux paquets successifs, périodique sur le nœud source, peut être égal à $T_i + J_{in_i}^q$ sur le nœud destinataire q (voir figure 8.2, avec $q = 2$) et conduire à un manque de données chez le client ayant demandé à visionner un contenu vidéo. Il est alors nécessaire d'augmenter la taille de la mémoire tampon chez le client.

8.5 AVEC REMISE EN FORME PAR FLUX

Pour limiter, voire supprimer les comportements décrits dans la section précédente, les flux doivent être remis en forme. Nous nous intéressons à deux techniques de remise en forme, à savoir :

- la technique de l'*annulation de gigue*, qui consiste à annuler sur chaque nœud la gigue d'un flux avant que celui-ci soit considéré par l'ordonnanceur du nœud [29] : les paquets du flux sont retenus jusqu'à leurs instants d'arrivée au plus tard. Ainsi, le flux arrive sur le nœud $h + 1$ avec une gigue ne dépendant que de celle introduite par le nœud précédent h et du lien entre les nœuds h et $h + 1$. Comme nous le verrons dans la section 8.5.1, le temps de réponse pire cas de bout-en-bout est obtenu en additionnant les temps de réponse pire cas, sans gigue (puisque annulée), obtenus sur chacun des nœuds visités. Une technique dérivée de l'annulation de gigue est la technique de limitation de gigue, consistant à vérifier que la gigue d'un flux reste inférieure à une certaine valeur avant qu'il soit considéré par l'ordonnanceur du nœud. Si ce n'est pas le cas, la gigue est réduite à la valeur maximale acceptable J en retenant les paquets jusqu'à leurs instants d'arrivée au plus tard moins J ;

- la technique du *seau à jetons*, qui consiste à limiter, voire annuler sur chaque nœud les rafales de paquets (*bursts*) d'un flux avant que celui-ci soit considéré par l'ordonnanceur du nœud [75, 49, 41]. Le seau à jetons peut modéliser un flux ou un ensemble de flux. Dans le premier cas, cela nécessite de maintenir des informations par flux dans chacun des nœuds visités. Dans le deuxième cas, la détermination de valeurs pertinentes pour les paramètres du seau à jetons est complexe lorsque les flux ont des caractéristiques différentes. Avec ce modèle apparaît donc le problème de fixer de bonnes valeurs pour une application donnée. Comme cela est montré dans [49] et [50], les temps de réponse de bout-en-bout dépendent fortement des valeurs attribuées aux paramètres du seau à jetons. Un mauvais choix peut conduire à de mauvais temps de réponse. De plus, les paramètres du seau à jetons, optimisés pour une configuration valide à un instant donné, doivent être recalculés sur chaque nœud en cas d'évolution de la configuration, afin de rester optimum. Cela n'est généralement pas réalisé.

Par ailleurs, il est montré dans [49] qu'il n'est pas plus avantageux d'utiliser, pour un flux donné, des seaux à jetons n'ayant pas les mêmes paramètres sur chacun des nœuds visités. C'est la raison pour laquelle, dans cette thèse, les seaux à jetons relatifs à un même flux ont mêmes paramètres. La borne sur le délai de bout-en-bout d'un flux τ_i est donnée par la somme des délais maximum de remise en forme de τ_i , plus la somme des délais maximum subis dans les ordonnanceurs de chacun des nœuds visités. Le délai maximum du *shaper* n'est subi qu'une fois et ne dépend pas du nombre de nœuds visités.

8.5.1 Remise en forme par annulation de gigue

L'annulation de gigue consiste à annuler sur chaque nœud la gigue introduite sur les paquets d'un flux avant qu'ils soient considérés par l'ordonnanceur du nœud. Plus précisément, un paquet quelconque traité dans un nœud appliquant la technique d'annulation de gigue ne sera considéré par l'ordonnanceur du nœud qu'à son instant d'arrivée au plus tard. Chaque nœud ayant sa propre horloge, cette technique nécessite que les horloges aient une dérive bornée et soient ε -synchronisées. Pour des raisons de clarté, nous ne considérons pas ε dans cette thèse².

La propriété suivante donne le temps de réponse et la gigue pire cas de bout-en-bout d'un flux lorsque la technique d'annulation de gigue est utilisée pour remettre en forme les flux.

Propriété 8.5.1 *Soit un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds. Si la technique d'annulation de gigue est utilisée pour remettre en forme le flux τ_i , alors :*

$$\begin{cases} R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max} \\ J_{out_i}^{\mathcal{L}_i} = R_{max_i}^{ordo(q)} - C_i^q. \end{cases}$$

Démonstration Soit τ_i un flux suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds numérotés de 1 à q . Sur chacun des nœuds visités, le flux τ_i est remis en forme par la technique d'annulation de gigue. Puisque sur le nœud source τ_i n'a pas de gigue d'activation, les paquets de τ_i sont ordonnancés dès leurs générations. Sur tout autre nœud $h \in \mathcal{L}_i$, les paquets de τ_i sont retenus dans le *shaper* jusqu'à leurs instants d'arrivée au plus tard. Les délais d'inter-arrivée des paquets de τ_i sont donc les mêmes dans l'ordonnanceur du nœud h que dans le nœud source. Ainsi, τ_i arrive dans le nœud $h+1$ avec une gigue dépendant uniquement de la gigue introduite par l'ordonnanceur du nœud précédent h et du lien entre ces deux nœuds (voir figure 8.3). Le retard subi par un paquet quelconque m de τ_i dans le *shaper* du nœud h , $h \in [2, q]$, est égal à :

²Il est montré dans [76] comment prendre en compte la précision de l'horloge.

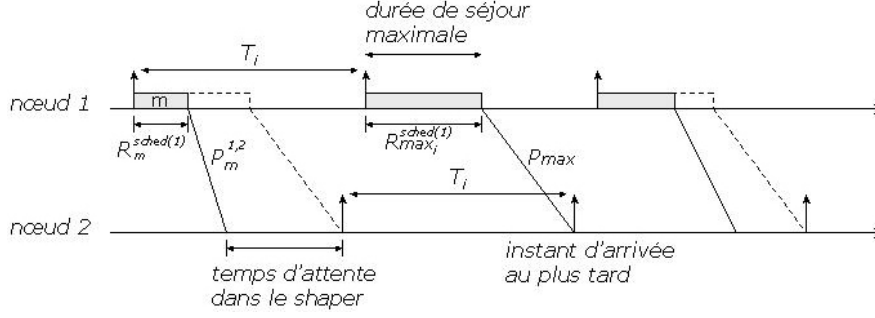


FIG. 8.3: Technique de l'annulation de gigue

$$R_m^{shap(h)} = \left(R_{max_i}^{ordo(h-1)} + L_{max} \right) - \left(R_m^{ordo(h-1)} + L_m^{h-1,h} \right).$$

Par conséquent, le temps de réponse de bout-en-bout du paquet m est égal à :

$$\begin{aligned} R_m^{\mathcal{L}_i} &= \sum_{h=1}^q R_m^{shap(h)} + \sum_{h=1}^q R_m^{ordo(h)} + \sum_{h=1}^{q-1} L_m^{h,h+1} \\ &= \sum_{h=2}^q \left(R_{max_i}^{ordo(h-1)} + L_{max} - R_m^{ordo(h-1)} - L_m^{h-1,h} \right) + \sum_{h=1}^q R_m^{ordo(h)} + \sum_{h=1}^{q-1} L_m^{h,h+1} \\ &= \sum_{h=1}^{q-1} R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max} + R_m^{ordo(q)}. \end{aligned}$$

Nous obtenons donc : $R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max}$.

La gigue de bout-en-bout du flux τ_i est égale à la gigue introduite par l'ordonnanceur du dernier nœud visité, puisque la gigue du flux est nulle en entrée de l'ordonnanceur grâce à la technique de l'annulation de gigue. Le temps de réponse d'un paquet quelconque de τ_i dans l'ordonnanceur du nœud q étant au plus égal à $R_{max_i}^{ordo(q)}$ et au moins égal à C_i^q , il vient : $J_{out_i}^{\mathcal{L}_i} = R_{max_i}^{ordo(q)} - C_i^q$. ■

8.5.2 Remise en forme par seau à jetons

La figure 8.4 représente un *shaper* par flux composé d'un seau à jetons de paramètres (σ, ρ) , où σ est la taille du seau et ρ le taux de remplissage du seau. Un seau à jetons (σ, ρ) fonctionne de la manière suivante. Le seau se remplit de ρ jetons toutes les secondes. Si σ jetons sont déjà présents dans le seau, les jetons supplémentaires sont jetés. Chaque jeton permet de transmettre une certaine quantité d'information (par exemple, un bit). Les paquets sont retenus dans la file d'attente du *shaper* jusqu'à ce que le nombre nécessaire de jetons soit atteint.

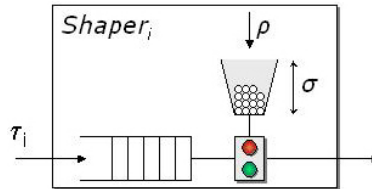


FIG. 8.4: Shaper composé d'un seau à jetons

Soit τ_i un flux sporadique remis en forme par un seau à jetons (σ, ρ) . Les propriétés suivantes fournissent :

- la configuration nécessaire des paramètres du seau pour annuler les rafales de paquets de τ_i ;
- le retard maximum subi par un paquet de τ_i lorsqu'il est remis en forme par un *shaper* bien configuré ;
- la gigue introduite par le *shaper* sur τ_i ;
- le délai d'inter-départ du *shaper* des paquets de τ_i lorsqu'ils sont générés à densité maximale.

Lemme 8.5.1 Soit τ_i un flux de délai minimum d'inter-arrivée T_i et générant des paquets de taille L_i . Soit h un nœud visité par τ_i dans lequel le flux est remis en forme par *shaper* composé d'un seau à jetons $(L_i, L_i/T_i)$. Soit x_m^h le nombre de jetons à générer à l'instant a_m^h afin de laisser passer le $m^{i\text{ème}}$ paquet de τ_i à travers le *shaper* du nœud h . Nous avons :

$$\begin{aligned} x_m^h &= \max(L_i - \sigma ; x_{m-1}^h + L_i - (a_m^h - a_{m-1}^h) \cdot \rho) \\ &= \max_{k=0..m-1} (L_i - \sigma + k \cdot L_i - (a_m^h - a_{m-k}^h) \cdot \rho). \end{aligned}$$

Démonstration Le nombre de jetons à générer à l'instant a_m^h afin de laisser passer le $m^{i\text{ème}}$ paquet de τ_i à travers le *shaper* du nœud h est égal au nombre de jetons qu'il y avait à générer à l'instant a_{m-1}^h afin de laisser passer le $(m-1)^{i\text{ème}}$ paquet de τ_i à travers le *shaper*, plus L_i jetons consommés par le paquet $(m-1)$, moins le nombre de jetons générés par le seau dans l'intervalle $[a_{m-1}^h, a_m^h]$. De plus, un paquet de τ_i nécessite au moins L_i jetons pour passer à travers le *shaper* et le nombre maximum de jetons est égal à σ . Il y a donc dans le seau, lors de l'arrivée du $m^{i\text{ème}}$ paquet :

- au plus $|L_i - \sigma|$ jetons en excès si $L_i \leq \sigma$;
- au moins $L_i - \sigma$ jetons à générer si $L_i \geq \sigma$.

x_m^h est donc borné inférieurement par $L_i - \sigma$.

Par ailleurs, si nous remplaçons successivement les termes x_j^h par leurs valeurs, avec $j = (m-1)..1$, il vient : $x_m^h = \max_{k=0..m-1} (L_i - \sigma + k \cdot L_i - (a_m^h - a_{m-k}^h) \cdot \rho)$. ■

Propriété 8.5.2 Soit τ_i un flux sporadique de délai minimum d'inter-arrivée T_i et générant des paquets de taille L_i . Si τ_i est remis en forme sur un nœud quelconque h par un seau à jetons (σ, ρ) , où $\sigma = L_i$ et $\rho = L_i/T_i$, alors ses rafales de paquets sont annulées. En effet, le délai d'inter-départ du *shaper* de deux paquets successifs m et $m+1$ de τ_i est borné inférieurement par T_i .

$$\sigma = L_i \text{ et } \rho = \frac{L_i}{T_i} \Rightarrow \begin{cases} T_i \leq d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h & \text{si } a_{m+1}^h - a_m^h > T_i ; \\ d_{m+1}^{shap(h)} - d_m^{shap(h)} = T_i & \text{sinon.} \end{cases}$$

Démonstration Soit un flux τ_i remis en forme par un seau à jetons $(L_i, L_i/T_i)$ sur chacun des nœuds visités. Soit h un des nœuds visités par τ_i . Le temps de réponse d'un paquet quelconque m de τ_i dans l'ordonnanceur du nœud h est égal à : $R_m^h = \max(0 ; x_m^h / \rho)$, où x_m^h représente le nombre de jetons à générer à l'instant a_m^h afin de laisser le paquet m traverser le *shaper* du nœud h . Le délai d'inter-départ du *shaper* de deux paquets successifs de τ_i est égal à :

$$d_{m+1}^{shap(h)} - d_m^{shap(h)} = a_{m+1}^h + R_{m+1}^{shap(h)} - (a_m^h + R_m^{shap(h)}) = a_{m+1}^h - a_m^h + \max\left(0 ; \frac{x_{m+1}^h}{\rho}\right) - \max\left(0 ; \frac{x_m^h}{\rho}\right).$$

De plus, $x_{m+1}^h = \max(L_i - \sigma ; x_m^h + L_i - (a_{m+1}^h - a_m^h) \cdot \rho)$. Par conséquent, si $\sigma = L_i$ et $\rho = L_i/T_i$,

$$d_{m+1}^{shap(h)} - d_m^{shap(h)} = a_{m+1}^h - a_m^h + \max\left(0 ; \frac{x_m^h}{\rho} + T_i - (a_{m+1}^h - a_m^h)\right) - \max\left(0 ; \frac{x_{m-1}^h}{\rho} + T_i - (a_m^h - a_{m-1}^h)\right).$$

Deux cas sont à considérer, à savoir :

- si $x_m^h = 0$. Alors, $d_{m+1}^{shap(h)} - d_m^{shap(h)} = a_{m+1}^h - a_m^h + \max(0; T_i - (a_{m+1}^h - a_m^h))$

$$T_i \leq d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h + \max(0; T_i - (a_{m+1}^h - a_m^h))$$
 - $a_{m+1}^h - a_m^h \leq T_i \Rightarrow d_{m+1}^{shap(h)} - d_m^{shap(h)} = T_i$,
 - $a_{m+1}^h - a_m^h \geq T_i \Rightarrow T_i \leq d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h$.
- si $x_m^h \geq 0$. Alors, $d_{m+1}^{shap(h)} - d_m^{shap(h)} = a_{m+1}^h - a_m^h + \max\left(0; \frac{x_m^h}{\rho} + T_i - (a_{m+1}^h - a_m^h)\right) - \frac{x_m^h}{\rho}$

$$d_{m+1}^{shap(h)} - d_m^{shap(h)} \geq a_{m+1}^h - a_m^h + \frac{x_m^h}{\rho} + T_i - (a_{m+1}^h - a_m^h) - \frac{x_m^h}{\rho} \geq T_i$$
 - si $x_m^h \leq L_i + (a_{m+1}^h - a_m^h) \cdot \rho$, $d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h$
 - $a_{m+1}^h - a_m^h \leq T_i \Rightarrow d_{m+1}^{shap(h)} - d_m^{shap(h)} = T_i$,
 - $a_{m+1}^h - a_m^h \geq T_i \Rightarrow T_i \leq d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h$.
 - si $x_m^h \geq L_i + (a_{m+1}^h - a_m^h) \cdot \rho$, $d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h + T_i - (a_{m+1}^h - a_m^h)$
 - $a_{m+1}^h - a_m^h \leq T_i \Rightarrow d_{m+1}^{shap(h)} - d_m^{shap(h)} = T_i$,
 - $a_{m+1}^h - a_m^h \geq T_i \Rightarrow T_i \leq d_{m+1}^{shap(h)} - d_m^{shap(h)} \leq a_{m+1}^h - a_m^h$. ■

Il est important de noter que le délai d'inter-départ du *shaper* de deux paquets successifs de τ_i est inférieur ou égal au délai d'inter-arrivée de ces paquets s'ils respectent le délai minimum d'inter-arrivée du flux.

La propriété 8.5.3 fournit une borne sur le retard maximum subi par un paquet de τ_i lorsque le flux est remis en forme par un seau à jetons $(L_i, L_i/T_i)$.

Propriété 8.5.3 Soit τ_i un flux sporadique de délai minimum d'inter-arrivée T_i et générant des paquets de taille L_i . Si τ_i suit une ligne de diffusion \mathcal{L}_i composée de q nœuds et est remis en forme sur chacun des q nœuds par un seau à jetons $(L_i, L_i/T_i)$, alors le retard maximum subi dans le *shaper* par un paquet de τ_i est borné par la gigue de τ_i en entrée du *shaper*.

$$\forall \text{ paquet } m \text{ de } \tau_i, \sigma = L_i \text{ et } \rho = \frac{L_i}{T_i} \Rightarrow R_m^{shap(h)} \leq S_{max_i}^h - S_m^h.$$

Cette borne étant atteinte lorsque les paquets de τ_i sont générés à densité maximale.

Démonstration Soit x_m^h le nombre de jetons à générer à l'instant a_m^h pour que le $m^{ième}$ paquet de τ_i passe à travers le *shaper* du nœud h , $h \in [1, q]$. Si x_m^h est négatif, cela signifie qu'il y a dans le seau plus de jetons que nécessaire. Dans ce cas, $|x_m^h|$ représente le nombre de jetons en excès. D'après le lemme 1 (donné en annexe), $x_m^h = \max_{k=0..m-1} (L_i - \sigma + k \cdot L_i - (a_m^h - a_{m-k}^h) \cdot \rho)$. De plus :

$$\begin{aligned} a_m^h - a_{m-k}^h &= (a_m^1 + S_m^h) - (a_{m-k}^1 + S_{m-k}^h) \\ &= k \cdot T_i + S_m^h - S_{m-k}^h \quad \text{si les paquets de } \tau_i \text{ sont générés à densité maximale,} \\ &\geq k \cdot T_i + S_m^h - S_{m-k}^h \quad \text{sinon.} \end{aligned}$$

Si, sur le nœud h , les paramètres du seau à jetons sont $(L_i, L_i/T_i)$, alors le nombre de jetons à générer à l'instant a_m^h pour que le $m^{ième}$ paquet de τ_i passe à travers le *shaper* du nœud h est borné par :

$$\begin{aligned}
x_m^h &= \max_{k=0..m-1} \left(L_i - \sigma + k \cdot L_i - (a_m^h - a_{m-k}^h) \cdot \rho \right) \\
&\leq \max_{k=0..m-1} \left(k \cdot L_i - (k \cdot T_i + S_m^h - S_{m-k}^h) \cdot \rho \right) \\
&\leq \max_{k=0..m-1} \left(k \cdot L_i - k \cdot T_i \cdot \frac{L_i}{T_i} - (S_m^h - S_{m-k}^h) \cdot \rho \right) \\
&\leq \max_{k=0..m-1} \left((S_{m-k}^h - S_m^h) \cdot \rho \right) \\
&\leq (S_{max_i^h} - S_m^h) \cdot \rho
\end{aligned}$$

Si les paquets de τ_i sont générés à densité maximale, alors $x_m^h = \max_{k=0..m-1} ((S_{m-k}^h - S_m^h) \cdot \rho)$. Par ailleurs, en notant m' le premier paquet de τ_i tel que le délai entre son instant de génération dans le nœud source et son instant d'arrivée dans le nœud h est maximum, il vient :

$$\forall m \geq m', \max_{k=0..m-1} ((S_{m-k}^h - S_m^h) \cdot \rho) = (S_{max_i^h} - S_m^h) \cdot \rho.$$

Par conséquent, le retard subi par un paquet quelconque m de τ_i dans le *shaper* du nœud h est égal à : $R_m^{shap(h)} = \max(0; x_m^h / \rho) \leq S_{max_i^h} - S_m^h$ (voir figure 8.5). Cette borne est atteinte pour tout paquet arrivant dans le *shaper* après celui ayant connu le délai maximum entre son instant de génération et son instant d'arrivée dans le nœud h , lorsque les paquets sont générés à densité maximale. ■

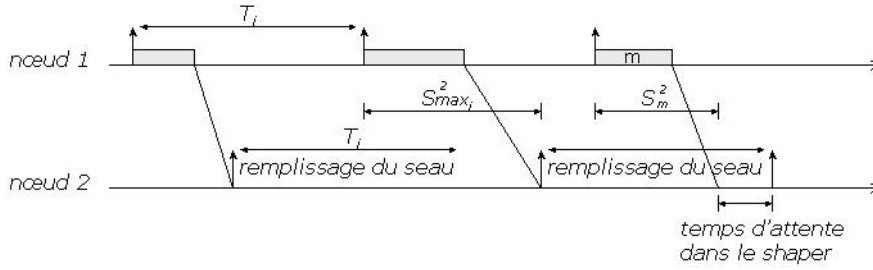


FIG. 8.5: Technique du seau à jetons

La propriété suivante montre que la remise en forme d'un flux τ_i par un *shaper* composé d'un seau à jetons $(L_i, L_i/T_i)$ n'introduit pas de gigue supplémentaire sur le flux.

Propriété 8.5.4 Soit τ_i un flux sporadique de délai minimum d'inter-arrivée T_i et générant des paquets de taille L_i . La remise en forme de τ_i dans un nœud h par un *shaper* composé d'un seau à jetons $(L_i, L_i/T_i)$ ne modifie pas la gigue de τ_i .

$$\forall \text{ paquet } m \text{ de } \tau_i, S_{min_i^h} \leq S_m^h + R_m^{shap(h)} \leq S_{max_i^h}.$$

Démonstration Pour tout paquet m de τ_i et pour tout nœud h visité par τ_i , le délai entre l'instant de génération du paquet m dans le nœud source et son instant de départ du *shaper* du nœud h est égal à : $d_m^{shap(h)} - a_m^1 = S_m^h + R_m^{shap(h)}$. Si $\sigma = L_i$ et $\rho = L_i/T_i$, alors $R_m^{shap(h)} \leq S_{max_i^h} - S_m^h$ (voir propriété 8.5.3). Par conséquent, nous obtenons :

$$\begin{aligned}
S_m^h &\leq d_m^{shap(h)} - a_m^1 \leq S_{max_i^h} - S_m^h + S_m^h \\
S_{min_i^h} &\leq S_m^h + R_m^{shap(h)} \leq S_{max_i^h}
\end{aligned}$$

■

Enfin, si les paquets de τ_i sont générés à densité maximale, alors dès qu'un paquet de τ_i connaît le délai maximum entre son instant de génération dans le nœud source et son instant d'arrivée dans le *shaper* du nœud h , le délai d'inter-départ de deux paquets successifs de τ_i est égal au délai minimum d'inter-arrivée du flux τ_i .

Propriété 8.5.5 Soit τ_i un flux sporadique remis en forme par un seau à jetons $(L_i, L_i/T_i)$ sur chacun des nœuds visités. Si les paquets de τ_i sont générés à densité maximale, alors dès qu'un paquet connaît le délai maximum entre son instant de génération dans le nœud source et son instant d'arrivée dans le shaper d'un nœud quelconque h , le $k^{ième}$ paquet de τ_i arrivé après m' dans le nœud h quitte le shaper exactement $k \cdot T_i$ unités de temps après m .

$$\forall \text{ paquet } m' \text{ de } \tau_i, S_{m'}^h = S_{max_i}^h \Rightarrow \forall k \in \mathbb{N}, d_{m'+k}^{shap(h)} - d_{m'}^{shap(h)} = k \cdot T_i.$$

Démonstration Soit τ_i un flux suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds. Pour tout paquet m et $m+1$ de τ_i , nous avons : $\forall h \in [1, q], (d_{m+1}^{shap(h)} - a_{m+1}^1) - (d_m^{shap(h)} - a_m^1) = d_{m+1}^{shap(h)} - d_m^{shap(h)} - T_i$. De plus, si $\sigma = L_i$ et $\rho = L_i/T_i$, alors $d_{m+1}^{shap(h)} - d_m^{shap(h)} \geq T_i$ (voir propriété 8.5.2). Nous avons donc : $(d_{m+1}^{shap(h)} - a_{m+1}^1) - (d_m^{shap(h)} - a_m^1) \geq 0$.

Le délai entre l'instant de génération d'un paquet de τ_i dans le nœud source et son instant de départ du shaper d'un nœud quelconque h augmente donc de paquet en paquet. D'après la propriété 8.5.6, ce délai est borné supérieurement, la borne étant atteinte lorsque les paquets sont générés à densité maximale. Soit m' un paquet de τ_i atteignant cette borne. Nous avons alors pour tout paquet $m' + k$ du flux τ_i , avec $k \in \mathbb{N}$: $d_{m'+k}^{shap(h)} - a_{m'+k}^1 = d_{m'}^{shap(h)} - a_{m'}^1$. Par conséquent, $d_{m'+k}^{shap(h)} - d_{m'}^{shap(h)} = a_{m'+k}^1 - a_{m'}^1 = k \cdot T_i$. ■

Nous évaluons maintenant le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque τ_i lorsque la technique du seau à jetons est utilisée dans chacun des nœuds visités pour remettre en forme τ_i .

Propriété 8.5.6 Soit τ_i un flux suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds. Si chacun des nœuds applique la technique du seau à jetons pour remettre en forme τ_i , avec un shaper de paramètres $(L_i, L_i/T_i)$, alors :

$$\begin{cases} R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max} \\ J_{out_i}^{\mathcal{L}_i} = \sum_{h=1}^q (R_{max_i}^{ordo(h)} - C_i^h) + (q-1) \cdot (L_{max} - L_{min}). \end{cases}$$

Démonstration Par récurrence. Soit τ_i un flux quelconque suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds. Chaque nœud utilise la technique du seau à jetons pour remettre en forme τ_i , avec un shaper de paramètres $(L_i, L_i/T_i)$. Le temps de réponse pire cas de τ_i dans le nœud source est égal au temps de séjour maximum connu par un paquet de τ_i dans l'ordonnanceur de ce nœud puisque le flux a une gigue d'activation nulle. Ainsi, nous avons : $R_{max_i}^1 = R_{max_i}^{ordo(1)}$.

Nous supposons maintenant que le temps de réponse pire cas de τ_i , après avoir visité $q-1$ nœuds, est égal à : $\sum_{h=1}^{q-1} R_{max_i}^{ordo(h)} + (q-2) \cdot L_{max}$.

Soit m' un paquet de τ_i tel que $S_{m'}^h = S_{max_i}^h$. Si les paquets de τ_i sont générés à densité maximale et si τ_i visite q nœuds, le temps de réponse d'un paquet quelconque $m \geq m'$ de τ_i est égal à :

$$\begin{aligned} R_m^{\mathcal{L}_i} &= S_m^q + R_m^{shap(q)} + R_m^{ordo(q)} \\ &= S_m^q + S_{max_i}^q - S_m^q + R_m^{ordo(q)} \quad (\text{voir propriété 8.5.3}) \\ &= \sum_{h=1}^{q-1} R_{max_i}^{ordo(h)} + (q-2) \cdot L_{max} + L_{max} + R_m^{ordo(q)} \\ &= \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max} \end{aligned}$$

En ce qui concerne la gigue du flux τ_i , si, sur un nœud $h \in [1, n]$, nous considérons seulement les paquets de τ_i arrivés après celui ayant connu le délai maximum entre son instant de génération dans le nœud source et

son instant d'arrivée dans le nœud h , alors τ_i quitte le *shaper* du nœud considéré sans gigue (voir section 8.6). Mais cela n'est plus vrai si nous considérons tous les paquets de τ_i . Par exemple, le premier paquet de τ_i visite tous les nœuds de la ligne de diffusion \mathcal{L}_i sans être retardé par les *shapers* puisque les seaux à jetons sont tous remplis. De plus, si ce paquet n'est retardé dans aucun des ordonnanceurs par des paquets d'autres flux, alors son temps de réponse minimum de bout-en-bout est égal à : $\sum_{h=1}^q C_i^h + (q-1) \cdot L_{min}$. Par conséquent, la gigue pire cas de bout-en-bout du flux τ_i est égale à : $J_{out_i}^{\mathcal{L}_i} = R_{max_i}^{\mathcal{L}_i} - R_{min_i}^{\mathcal{L}_i}$, ce qui est inférieur à $\sum_{h=1}^q (R_{max_i}^{ordo(h)} - C_i^h) + (q-1) \cdot (L_{max} - L_{min})$. Cette borne est atteinte lorsque les paquets de τ_i sont générés à densité maximale. ■

8.6 EXEMPLE

Il est important de noter que pour tout flux τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, si les paquets de τ_i sont générés à densité maximale alors le temps de réponse pire cas de bout-en-bout de τ_i est égal à : $R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max}$, quelque soit la technique de remise en forme utilisée. En effet, dès qu'un paquet de τ_i connaît le délai maximum entre son instant de génération dans le nœud source et son instant d'arrivée dans un nœud quelconque h , la technique d'annulation de gigue et celle du seau à jetons sont similaires. Cette propriété est illustrée par l'exemple suivant.

Considérons un flux quelconque τ_i avec un délai minimum d'inter-arrivée $T_i = 5$ et générant des paquets à densité maximale. Soit h un des nœuds visités par τ_i . Un paquet quelconque m de τ_i arrive dans le nœud h après S_m^h unités de temps. Supposons que $S_{min_i}^h = 3$ et $S_{max_i}^h = 10$, soit $J_{in_i}^h = 7$. Comme le montre la figure 8.6, le huitième paquet est le premier à vérifier la condition $S_m^h = S_{max_i}^h$. Tous les paquets arrivant après celui-ci connaissent le même délai entre leurs instants de génération sur le nœud source et leurs instants de départ du *shaper*, quelque soit la technique de remise en forme utilisée. En terme de temps de réponse de bout-en-bout, les deux techniques sont équivalentes.

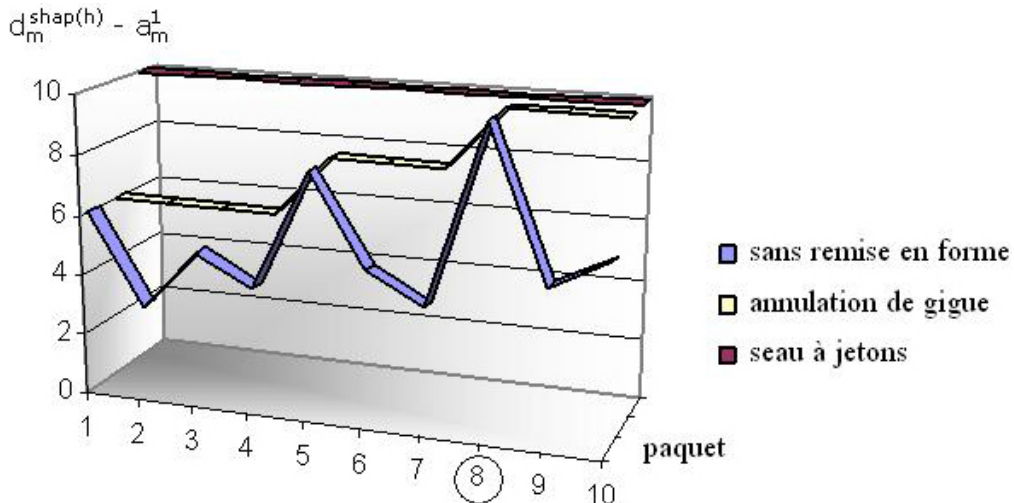


FIG. 8.6: Temps de réponse avec et sans remise en forme

Par contre, en terme de gigue de bout-en-bout, la technique de l'annulation de gigue fournit de meilleurs résultats que celle du seau à jetons. En effet, si nous considérons seulement les paquets arrivés après le

huitième, la gigue du flux τ_i est nulle en sortie du *shaper* car les paquets connaissent tous le même délai entre leurs instants de génération sur le nœud source et leurs instants de départ du *shaper* (voir section 8.5.2). Si nous considérons, en revanche, tous les paquets de τ_i (y compris ceux arrivés avant le huitième), la gigue de bout-en-bout de τ_i est pire en utilisant la technique du seau à jetons, comme nous pouvons le voir figure 8.6.

8.7 ETUDE COMPARATIVE AVEC L'ORDONNANCEMENT FIFO

Dans cette section, nous déterminons le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque τ_i , avec et sans remise en forme du trafic, lorsque les flux sont ordonnancés FIFO.

8.7.1 Sans remise en forme

Nous nous intéressons au temps de réponse et à la gigue pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux sont ordonnancés FIFO, sans remise en forme du trafic. Nous rappelons que l'ordonnancement FIFO est équivalent à l'ordonnancement FIFO* en contexte monoprocesseur ou lorsque les flux suivent tous la même ligne de diffusion, pour lequel nous avons établi des résultats dans les chapitres 5 et 6. Nous présentons ci-dessous la propriété donnant le temps de réponse pire cas d'un flux quelconque τ_i en contexte monoprocesseur lorsque les flux sont considérés sans gigue d'activation.

Propriété 8.7.1 *En contexte monoprocesseur, si les flux sont ordonnancés FIFO et sont activés sans gigue, alors le temps de réponse pire cas d'un flux quelconque τ_i est obtenu dans la première période active du scénario synchrone, c'est-à-dire lorsque les premiers paquets des flux sont activés en même temps. Nous avons alors : $R_{max_i} = \sum_{j=1}^n C_j$.*

Démonstration D'après la propriété 5.6.1 établie dans le chapitre 5, avec pour chaque flux une gigue d'activation nulle. ■

Nous adoptons maintenant l'approche holistique pour déterminer le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i dans le cas général distribué lorsque les flux sont ordonnancés FIFO. Le temps de réponse pire cas de bout-en-bout de τ_i est alors borné par la somme de ses temps de réponse sur chacun des nœuds visités, plus les délais réseau maximum, moins la gigue en entrée de chaque nœud distinct du nœud source, soit :

$$R_{max_i}^{\mathcal{L}_i} \leq \sum_{h=1}^q R_{max_i}^h + (q-1) \cdot L_{max} - \sum_{h=2}^q J_{in_i}^h, \text{ où } J_{in_i}^h = R_{max_i}^{h-1} - C_i^{h-1} + L_{max} - L_{min}.$$

Propriété 8.7.2 *En environnement distribué, lorsque les flux sont ordonnancés FIFO dans chacun des nœuds visités, alors le temps de réponse et la gigue de bout-en-bout d'un flux quelconque τ_i suivant une ligne de diffusion \mathcal{L}_i satisfont :*

$$\begin{cases} R_{max_i}^{\mathcal{L}_i} \leq \sum_{h=1}^q \left(\sum_{j=1}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor \frac{J_{in_i}^h}{T_i} \right\rfloor \cdot C_j^h ; \sum_{j=1, j \neq i}^n \frac{J_{in_i}^h}{T_i} \cdot C_j^h \right) \right) + (q-1) \cdot L_{max} \\ J_{out_i}^{\mathcal{L}_i} \leq \sum_{h=1}^q \left(\sum_{j=1, j \neq i}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor \frac{J_{in_i}^h}{T_i} \right\rfloor \cdot C_j^h ; \sum_{j=1, j \neq i}^n \frac{J_{in_i}^h}{T_i} \cdot C_j^h \right) \right) + (q-1) \cdot (L_{max} - L_{min}). \end{cases}$$

Démonstration La propriété 5.6.1, page 67 donne le temps de réponse maximum du flux τ_i en contexte monoprocesseur, lorsque les flux sont ordonnancés FIFO. Sur tout nœud h visité par τ_i , nous avons alors :

$$Rmax_i^h = \max_{-Jin_i^h \leq t} \left\{ \sum_{j=1}^n \left(1 + \left\lfloor (t + Jin_j^h)/T_j \right\rfloor \right) \cdot C_j^h - t \right\}.$$

Nous distinguons deux cas, à savoir :

- $t \leq 0$. Dans ce cas, $Rmax_i^h \leq \sum_{j=1}^n \left(1 + \left\lfloor Jin_j^h/T_j \right\rfloor \right) \cdot C_j^h + Jin_i^h$;
- $t \geq 0$. Dans ce cas, $Rmax_i^h \leq \sum_{j=1}^n \left(1 + Jin_j^h/T_j \right) \cdot C_j^h$.

Par conséquent, le temps de réponse maximum de τ_i dans le nœud h est borné par :

$$\sum_{j=1}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor Jin_j^h/T_j \right\rfloor \cdot C_j^h + Jin_i^h ; \sum_{j=1}^n \left(Jin_j^h/T_j \right) \cdot C_j^h \right).$$

En appliquant l'approche holistique, nous obtenons : $Rmax_i^{\mathcal{L}_i} \leq \sum_{h=1}^q Rmax_i^h + (q-1) \cdot Lmax - \sum_{h=2}^q Jin_i^h$. Ainsi, il vient :

$$\begin{aligned} & \sum_{h=1}^q \left(\sum_{j=1}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor \frac{Jin_j^h}{T_j} \right\rfloor \cdot C_j^h + Jin_i^h ; \sum_{j=1}^n \frac{Jin_j^h}{T_j} \cdot C_j^h \right) \right) + (q-1) \cdot Lmax - \sum_{h=2}^q Jin_i^h. \\ & \leq \sum_{h=1}^q \left(\sum_{j=1}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor \frac{Jin_j^h}{T_j} \right\rfloor \cdot C_j^h ; \sum_{j=1}^n \frac{Jin_j^h}{T_j} \cdot C_j^h \right) \right) + (q-1) \cdot Lmax. \end{aligned}$$

Puisque la gigue de bout-en-bout est égale à : $Rmax_i^{\mathcal{L}_i} - Rmin_i^{\mathcal{L}_i}$, où $Rmin_i^{\mathcal{L}_i} = \sum_{h=1}^q C_j^q + (q-1) \cdot Lmin$, $Jout_i^{\mathcal{L}_i}$ est borné par :

$$\sum_{h=1}^q \left(\sum_{j=1}^n C_j^h + \max \left(\sum_{j=1}^n \left\lfloor \frac{Jin_j^h}{T_j} \right\rfloor \cdot C_j^h ; \sum_{j=1}^n \frac{Jin_j^h}{T_j} \cdot C_j^h \right) \right) + (q-1) \cdot (Lmax - Lmin). \quad \blacksquare$$

8.7.2 Avec remise en forme par annulation de gigue

Nous commençons par nous intéresser au scénario pire cas d'un flux quelconque τ_i lorsque les paquets sont ordonnancés FIFO.

Propriété 8.7.3 *Si les nœuds du réseau ordonnancent les flux selon FIFO et utilisent la technique de l'annulation de gigue pour les remettre en forme, alors le temps de réponse pire cas de bout-en-bout d'un flux quelconque τ_i est obtenu en considérant le scénario synchrone dans chacun des nœuds visités.*

Démonstration Puisque la technique de l'annulation de gigue est utilisée dans tous les nœuds du réseau, τ_i entre dans l'ordonnancier d'un nœud sans gigue. En effet, les paquets sont retenus dans le *shaper* jusqu'à leurs instants d'arrivée au plus tard. Par conséquent, lorsque τ_i visite les nœuds h et $h+1$, les instants d'arrivée de ses paquets sur l'ordonnancier du nœud h sont translatés de la même valeur sur l'ordonnancier du nœud $h+1$.

D'après la propriété 8.7.1, tous les flux visitant le nœud h connaissent le même temps de réponse pire cas dans l'ordonnancier dû à l'ordonnancement FIFO. Ce temps de réponse pire cas est obtenu lorsque tous les flux sont synchrones sur l'ordonnancier du nœud. Par conséquent, si les flux sont synchrones sur le nœud h , ils le seront sur le nœud $h+1$. ■

La propriété suivante donne le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque τ_i lorsque la technique de l'annulation de gigue est utilisée pour remettre en forme les flux et l'ordonnancement appliqué est FIFO.

Propriété 8.7.4 *Si les nœuds du réseau ordonnancent les flux selon FIFO et utilisent la technique de l'annulation de gigue pour les remettre en forme. Le temps de réponse et la gigue de bout-en-bout d'un flux quelconque τ_i satisfont :*

$$\begin{cases} R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q \sum_{j=1}^n C_j^h + (q-1) \cdot L_{max} \\ J_{out_i}^{\mathcal{L}_i} = \sum_{\substack{j=1 \\ j \neq i}}^n C_j^q. \end{cases}$$

Démonstration Lorsque la technique de l'annulation de gigue est utilisée pour remettre en forme les flux, le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à : $\sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max}$ (voir propriété 8.5.1). De plus, si les flux sont ordonnancés FIFO, le temps de réponse pire cas de bout-en-bout est obtenu en considérant le scénario synchrone dans chacun des nœuds visités (voir propriété 8.7.3). Puisque les flux entrent dans l'ordonnanceur d'un nœud sans gigue d'activation, il vient : $R_{max_i}^{ordo(h)} = \sum_{j=1}^n C_j^h$, pour tout $h \in [1, n]$ (voir propriété 8.7.1). ■

8.7.3 Avec remise en forme par seau à jetons

Avant de calculer le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque τ_i lorsque la technique du seau à jetons est utilisée pour remettre en forme les flux, nous montrons comment obtenir le temps de réponse pire cas de bout-en-bout de τ_i .

Propriété 8.7.5 *Soit τ_i un flux remis en forme par un seau à jetons $(L_i, L_i/T_i)$ sur chacun des nœuds visités. Si les flux sont ordonnancés FIFO, alors une borne sur le temps de réponse pire cas de bout-en-bout de τ_i est obtenue en considérant le scénario synchrone dans chacun des nœuds visités.*

Démonstration Sur tout nœud h visité par τ_i , le seau à jetons $(L_i, L_i/T_i)$ garantit un délai d'inter-arrivée dans l'ordonnanceur de deux paquets successifs de τ_i au moins égal à T_i . Par conséquent, du point de vue de l'ordonnanceur, les paquets sont activés sans gigue. De plus, si ces paquets sont générés à densité maximale et sont ordonnancés FIFO dans le nœud h , le temps de réponse pire cas de τ_i dans l'ordonnanceur de ce nœud est obtenu dans le scénario synchrone (voir propriété 8.7.1). En effet, supposons que les paquets soient générés à densité maximale et que sur un nœud quelconque h de la ligne de diffusion \mathcal{L}_i suivie par τ_i , les flux connaissent à tour de rôle le temps de réponse pire cas dans l'ordonnanceur. Un flux connaît alors le temps de réponse pire cas, au plus tard après $n \cdot \text{PPCM}_{j=1..n}(T_j)$, où n est le nombre de flux dans le réseau. Par conséquent, si les flux visitant les nœuds h et $h+1$ sont synchrones sur le nœud h , alors ils seront synchrones sur le nœud $h+1$ au plus tard après $n \cdot \text{PPCM}_{j=1..n}(T_j)$. ■

La propriété 8.7.5 nous permet de déterminer une borne sur le temps de réponse et la gigue pire cas de bout-en-bout d'un flux quelconque τ_i lorsque les flux sont ordonnancés FIFO.

Propriété 8.7.6 *Soit τ_i un flux remis en forme par un seau à jetons $(L_i, L_i/T_i)$ sur chacun des nœuds visités. Soit \mathcal{L}_i la ligne de diffusion suivie par τ_i , composée de q nœuds. Si les flux sont ordonnancés FIFO, alors :*

$$\begin{cases} R_{max_i}^{\mathcal{L}_i} \leq \sum_{h=1}^q \sum_{j=1}^n C_j^h + (q-1) \cdot L_{max} \\ J_{out_i}^{\mathcal{L}_i} \leq \sum_{h=1}^q \sum_{\substack{j=1 \\ j \neq i}}^n C_j^h + (q-1) \cdot (L_{max} - L_{min}). \end{cases}$$

Démonstration D'après la propriété 8.5.6, le temps de réponse pire cas de bout-en-bout du flux τ_i est égal à : $\sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max}$. Par les propriétés 8.7.1 et 8.7.5, si les flux sont ordonnancés FIFO, alors : $\forall h \in [1, q], R_{max_i}^{ordo(h)} \leq \sum_{j=1}^n C_j^h$. ■

8.8 DISCUSSION

8.8.1 Annulation de gigue ou seau à jetons ?

Comme cela a été souligné dans la section 8.6, le temps de réponse pire cas de bout-en-bout d'un flux τ_i suivant une ligne \mathcal{L}_i composée de q nœuds est égal à : $R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q R_{max_i}^{ordo(h)} + (q-1) \cdot L_{max}$, quelque soit la technique de remise en forme utilisée (annulation de gigue ou seau à jetons). De plus, si les flux sont ordonnancés FIFO, alors la garantie sur le temps de réponse de bout-en-bout de τ_i est la même pour les deux techniques étudiées de remise en forme et est égale à : $R_{max_i}^{\mathcal{L}_i} = \sum_{h=1}^q \sum_{j=1}^n C_j^h + (q-1) \cdot L_{max}$.

Puisque les deux techniques fournissent la même garantie déterministe en terme de temps de réponse de bout-en-bout, l'une n'est pas plus appropriée que l'autre lorsque les besoins de QoS ne concernent que ce paramètre. Cela dit, la technique du seau à jetons ne nécessite pas d'avoir des horloges ε -synchronisées, contrairement à la technique de l'annulation de gigue (voir section 8.5.1).

Par contre, en comparant les résultats donnés par les propriétés 8.7.4 et 8.7.6, il apparaît que la technique de l'annulation de gigue est toujours plus appropriée que celle du seau à jetons lorsque la gigue de bout-en-bout est le paramètre important. En effet, la garantie sur la gigue pire cas de bout-en-bout du flux τ_i est égale à $\sum_{j=1}^n C_j^q$ lorsque la technique de remise en forme utilisée est celle de l'annulation de gigue, et à $\sum_{h=1}^q \sum_{j \neq i}^n C_j^h + (q-1) \cdot (L_{max} - L_{min})$ lorsque la technique utilisée est celle du seau à jetons. La différence est égale à $\sum_{h=1}^{q-1} \sum_{j \neq i}^n C_j^h + (q-1) \cdot (L_{max} - L_{min})$, augmentant avec le nombre de nœuds visités.

8.8.2 Avec ou sans remise en forme ?

8.8.2.a Même ligne de diffusion

Nous commençons par nous intéresser au cas où l'ensemble des flux suivent la même ligne de diffusion dans le réseau. De même que dans la section précédente, nous montrons que la technique la plus appropriée dépend des besoins de QoS. En effet, si le temps de réponse de bout-en-bout est le paramètre important, alors il est préférable de ne pas remettre en forme le trafic. En revanche, si la gigue de bout-en-bout est le paramètre important, la technique de l'annulation de gigue fournit une meilleure garantie.

Lorsque les flux sont ordonnancés FIFO et aucune remise en forme du trafic n'est effectuée, le temps de réponse de bout-en-bout d'un flux τ_i est borné par : $\sum_{j=1}^n C_j^{slow} + \sum_{h=1}^q \max_{j=1..n, h \neq slow} (C_j^h) + (q-1) \cdot L_{max}$, où *slow* est le nœud le plus lent parmi les q nœuds visités de la ligne de diffusion \mathcal{L} . Lorsque le trafic est remis en forme, quelque soit la technique utilisée, le temps de réponse pire cas de bout-en-bout est égal à : $\sum_{h=1}^q \sum_{j=1}^n C_j^h + (q-1) \cdot L_{max}$ sur toute ligne de diffusion \mathcal{L} composée de q nœuds. La différence est donc égale à : $\sum_{h=1}^q \sum_{j=1}^n (C_j^h - \max_{j=1..n} (C_j^h))$, quantité qui augmente avec le nombre de nœuds visités. Les techniques étudiées de remise en forme sont donc contre-performantes, en terme de temps de réponse de bout-en-bout, lorsque les flux suivent tous la même ligne de diffusion, et ce quelque soit le nombre de nœuds visités et le nombre de flux.

Concernant la gigue de bout-en-bout, la technique de l'annulation de gigue fournit la meilleure garantie, mais il est intéressant de souligner qu'il est préférable de ne pas remettre en forme le trafic plutôt que d'utiliser la technique du seau à jetons. En effet, pour un flux quelconque τ_i , le temps de réponse minimum de bout-en-

bout de τ_i est le même dans les deux cas alors que le temps de réponse maximum de bout-en-bout est plus petit lorsque le trafic n'est pas remis en forme. Cela dit, il peut être utile, voire indispensable d'annuler les rafales de paquets d'un flux sur les différents nœuds visités (voir section 8.4).

8.8.2.b Cas général

Dans le cas général, c'est-à-dire le cas où les flux suivent des lignes de diffusion différentes, la propriété 8.7.2 étend les résultats connus en contexte monoprocesseur pour l'ordonnancement FIFO. Lorsque le trafic n'est pas remis en forme, les bornes sur le temps de réponse de bout-en-bout et la gigue de bout-en-bout ont été calculées en utilisant l'approche holistique. Ces bornes ne sont donc pas nécessairement atteintes. Les garanties déterministes fournies à un flux quelconque τ_i sur le temps de réponse et la gigue de bout-en-bout dépendent non seulement du nombre de nœuds visités par τ_i mais également de la gigue des flux croisant τ_i . Par conséquent, les techniques étudiées de remise en forme sont appropriées non seulement lorsque τ_i suit une ligne de diffusion composée d'un grand nombre de nœuds, mais également lorsque les giges des flux croisant τ_i sont grandes.

Supposons, par exemple, que le flux τ_i visite un unique nœud h dans le réseau. Si ce nœud n'applique aucune technique de remise en forme, le temps de réponse du flux τ_i sur le nœud h est plus grand que : $\sum_{j=1}^n (1 + \lfloor J_{in_j^h}/T_j \rfloor) \cdot C_j^h$ (voir propriété 5.6.1). Ce délai augmente si les giges des autres flux visitant le nœud h augmentent. La technique de l'annulation de gigue, en annulant la gigue en entrée de l'ordonnanceur du nœud h des flux visitant ce nœud, réduit le temps de réponse pire cas de τ_i à : $\sum_{j=1}^n C_j^h$. La technique du seau à jetons, en annulant les rafales de paquets des flux visitant le nœud h , permet d'obtenir le même résultat puisque ces flux arrivent sans gigue d'activation du point de vue de l'ordonnanceur du nœud.

8.9 CONCLUSION

La variation des délais réseau et des durées de séjour dans les nœuds conduit à une distorsion du trafic : le délai minimum d'inter-arrivée entre deux paquets successifs d'un même flux n'est donc plus respecté. Afin d'éviter ce phénomène, des techniques de remise en forme du trafic ont été introduites. Dans ce chapitre, nous nous sommes intéressés à trois techniques : aucune remise en forme, remise en forme par annulation de gigue et remise en forme par seau à jetons. Nous avons établi des résultats, indépendamment de la politique d'ordonnancement.

Nous avons alors calculé une borne sur le temps de réponse de bout-en-bout et une borne sur la gigue de bout-en-bout pouvant être garanties, de manière déterministe, aux flux lorsqu'ils sont ordonnancés FIFO, avec et sans remise en forme. Nous avons finalement discuté le mérite relatif de chacune des techniques étudiées.

CHAPITRE 9

Exemples d'applications

9.1	Introduction	148
9.2	Architecture DiffServ	148
9.2.1	Modèle DiffServ	148
9.2.2	MPLS	148
9.2.3	Solution proposée	150
9.2.4	Contrôle d'admission	151
9.2.5	Exemple	153
9.2.6	Discussion	154
9.3	Autres architectures	155
9.3.1	Architecture IntServ	155
9.3.2	Architecture hybride	155
9.4	Conclusion	155

9.1 INTRODUCTION

Dans ce chapitre, nous proposons une solution basée sur DiffServ et MPLS pour fournir à des applications temps-réel des garanties déterministes sur le temps de réponse et la gigue de bout-en-bout. Dans ce cadre, nous montrons la mise en œuvre d'un contrôle d'admission en nous basant sur les résultats établis dans le chapitre 7. Nous montrons ensuite comment étendre ces résultats à d'autres architectures de QoS (architecture IntServ et architecture hybride).

9.2 ARCHITECTURE DIFFSERV

Dans cette section, nous proposons une solution pour fournir des garanties déterministes de bout-en-bout à des applications temps-réel dans un domaine DiffServ. Plus précisément, nous montrons comment garantir aux applications à fortes contraintes temporelles que le temps de réponse (respectivement la gigue) pire cas de bout-en-bout de leurs paquets n'excédera pas D (respectivement J), avec D et J deux paramètres du domaine DiffServ. Notre solution est basée sur la combinaison de DiffServ et MPLS (*MultiProtocol Label Switching*), avec l'établissement d'un contrôle d'admission appliquant les résultats obtenus dans le chapitre 7. Nous utilisons dans cette section les notations présentées dans la section 4.3.2, page 51.

Avant de décrire cette solution, nous rappelons le principe de l'architecture DiffServ en insistant sur l'absence de garanties quantitatives pour les applications temps-réel, puis nous présentons le protocole MPLS.

9.2.1 Modèle DiffServ

Le modèle de QoS DiffServ a été présenté dans la section 3.4, page 37. Dans une telle architecture, le trafic est réparti dans des classes de service dont le nombre est limité. En particulier, la classe EF (*Expedited Forwarding*) a été proposée pour les applications nécessitant de faibles délais (temps de réponse et gigue de bout-en-bout) et un faible taux de pertes des paquets. Cette classe (la plus prioritaire) est donc bien adaptée pour les applications temps-réel. Cependant, la définition du service EF, donnée dans [55], n'est que qualitative : faibles délais et faibles taux de pertes. Aucune garantie quantitative n'est proposée. De plus, il a été montré dans [77] que la gigue de bout-en-bout du trafic EF pouvait être élevée dans le cas de grands réseaux.

L'utilisation de l'algorithme d'ordonnancement FIFO au sein de la classe EF a été discuté dans [78], mais les bornes obtenues sur les délais ne sont valides que pour un faible volume de trafic EF. Par ailleurs, un contrôle d'admission *hybride* a été proposé dans [79], basé sur (i) une remise en forme du trafic EF dans les nœuds en bordure du domaine et (ii) une allocation de ressources effectuée à partir de mesures réalisées dans le réseau. Cette approche conduit à un meilleur taux d'utilisation, mais ne fournit pas de garanties déterministes.

Nous proposons un contrôle d'admission déterministe basé sur le calcul d'une borne du temps de réponse de bout-en-bout pour la classe EF. Ceci permet d'offrir des garanties quantitatives déterministes à la classe EF.

9.2.2 MPLS

Dans cette solution, nous avons retenu MPLS car il permet de (i) fixer le chemin parcouru par un flux et (ii) indiquer la classe de service à laquelle appartient le flux lorsque MPLS est couplé à DiffServ. MPLS repose sur le principe suivant. Chaque LSR (*Label Switching Router*), c'est-à-dire chaque routeur (ou commutateur) capable de commuter des paquets en fonction des labels qu'ils contiennent, partitionne l'ensemble de tous les

paquets qu'il peut avoir à transmettre en un nombre fini de sous-ensembles disjoints, appelés FEC (*Forwarding Equivalence Classes*). Une FEC est donc un ensemble de paquets partageant les mêmes caractéristiques pour leur transport. A chaque FEC est ensuite associé un *label*¹ (une étiquette), qui est un identificateur relativement court, de longueur fixée, utilisé pour la transmission des paquets. Un chemin étiqueté, ou LSP (*Label Switched Path*), est ainsi constitué pour chaque FEC.

Contrairement au routage traditionnel, où la décision d'acheminement des paquets est prise en fonction des adresses des réseaux de destination, MPLS permet de constituer des FEC selon de nombreux critères. Le routeur d'entrée peut ainsi utiliser toute information qu'il a sur un paquet pour l'affecter à une FEC. Par exemple, les paquets arrivant sur différents ports peuvent être assignés à différentes FEC. La granularité d'une FEC est donc une caractéristique importante puisqu'elle permet de rendre le système de routage extensible et fonctionnel, la complexité du mécanisme d'assignation n'ayant pas d'impact sur les routeurs de cœur. Ce principe permet donc de simplifier les traitements effectués dans le cœur du réseau en reportant les traitements complexes sur les nœuds en bordure, dans le même esprit que l'architecture DiffServ.

En effet, lorsqu'un paquet entre dans le réseau MPLS, un label lui est affecté en fonction de la FEC à laquelle il appartient. A l'intérieur du réseau, les routeurs n'analysent plus l'en-tête IP du paquet ; ils transmettent le paquet grâce à son label. Le routeur de sortie retire le label du paquet et le transmet vers sa destination finale selon les techniques classiques de routage. Un label a la structure présentée à la figure 9.1.



FIG. 9.1: Structure d'un label

Dans cette structure, nous avons :

- un bit *Stack*, indiquant si le label est le seul dans la pile² ;
- un champ *Exp* (*Experimental*), dont l'usage n'a pas été entièrement défini lors de la standardisation de l'en-tête contenant le(s) label(s).
- un octet TTL, permettant d'éviter les problèmes de boucles dans les réseaux MPLS.

MPLS permet des avancées intéressantes dans les domaines du *traffic-engineering* [81] et de la QoS [82], en proposant des services qui ne sont pas possibles avec les techniques classiques de routage IP. Associé à DiffServ, MPLS permet d'attribuer un label à chaque classe de service, offrant ainsi un contrôle très fin des ressources utilisées et des chemins empruntés [83]. Cela répond à un problème essentiel du modèle DiffServ. En effet, l'utilisation de MPLS dans une architecture DiffServ rend possible l'acheminement des flux ayant des caractéristiques de QoS voisines par un LSP spécifique. Plus précisément, si le DSCP³ est codé dans le champ EXP (voir figure 9.2), un label permet alors à un nœud de connaître le prochain nœud à qui envoyer le paquet, mais également le traitement à appliquer en terme de QoS. Les flux suivant un même LSP sont affectés à différentes classes de service dans les nœuds. On parle alors de E-LSP. Cependant, le nombre de

¹Le label est associé localement (sur chaque nœud du réseau) à une FEC déterminée.

²Un paquet peut contenir plusieurs labels, on parle alors de pile de labels [80]. L'empilement des labels permet d'utiliser MPLS de façon hiérarchique et offre ainsi une meilleure extensibilité du routage dans l'Internet.

³Présenté dans la section 3.4, le DSCP (*Differentiated Services CodePoint*) indique aux routeurs du domaine l'agrégat auquel le paquet appartient et donc le traitement à appliquer.

classes doit être limité à huit afin de pouvoir coder le DSCP (six bits) dans le champ EXP (trois bits). S'il y a plus de huit classes de service, il est possible de définir plusieurs LSP pour une même destination, chaque LSP correspondant à une classe de service. On parle alors de L-LSP.

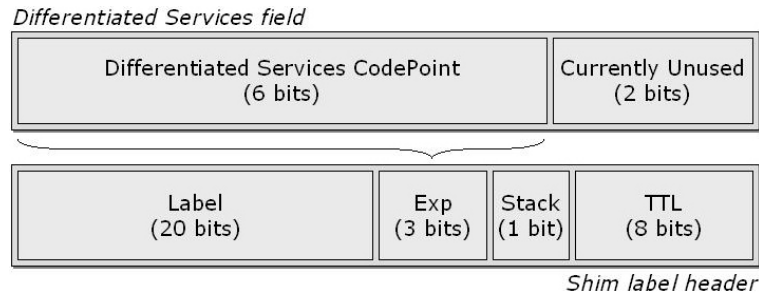


FIG. 9.2: Combinaison de DiffServ et MPLS

9.2.3 Solution proposée

Afin de garantir des bornes déterministes sur les temps de réponse et les giges de bout-en-bout d'applications temps-réel, nous proposons une solution basée sur une combinaison DiffServ/MPLS, où les flux sont ordonnancés FP/DP*. Nous considérons qu'à chaque classe est associée une priorité fixe et les paquets d'une même classe sont traités selon un ordonnancement de type DP*. Nous supposons que la classe EF a la plus forte priorité et que l'ordonnancement appliqué au sein de la classe EF est FIFO*. Nous considérons également que les nœuds gèrent d'autres classes de service de type AF (*Assured Forwarding*) et *best-effort*. Dans [65, 84], nous avons montré comment déterminer le temps de réponse de bout-en-bout lorsque les flux sont ordonnancés FP/FIFO.

Une implémentation de notre solution consiste à coder la priorité dynamique d'un paquet dans une partie de son label. Par exemple, le champ *label* de 20 bits peut être décomposé en :

- son label proprement dit sur 10 bits ;
- sa priorité dynamique sur 10 bits.

Notons que la priorité fixe d'un paquet est contenue dans le champ EXP du label. Un paquet appartient à la classe de service correspondant à sa priorité fixe.

La borne établie dans le chapitre 7 sur le temps de réponse pire cas de bout-en-bout d'un flux permet de dériver un contrôle d'admission. Il est alors possible de proposer des garanties déterministes sur les temps de réponse de bout-en-bout des flux EF dans le domaine DiffServ considéré. Ainsi, lorsqu'un flux entre dans le réseau, le DSCP se trouvant dans son en-tête IP détermine le label qui sera attribué à chacun de ses paquets. Si ce flux appartient à la classe EF, alors le contrôle d'admission est appelé. Si le flux est accepté, le domaine garantit un temps de réponse de bout-en-bout inférieur ou égal à D et une gigue de bout-en-bout inférieure ou égale à J . MPLS est alors utilisé pour la transmission des paquets.

Par ailleurs, les nœuds de sortie doivent s'assurer que la gigue de bout-en-bout d'un flux EF reste bornée par J . Si tel n'est pas le cas, le flux est remis en forme pour rester conforme. En effet, soient τ_i un flux de la classe EF suivant une ligne de diffusion \mathcal{L}_i dans le domaine et $R_i^{\mathcal{L}_i}$ son temps de réponse pire cas de bout-en-bout. Pour respecter la gigue de bout-en-bout maximale acceptable, le dernier nœud du domaine visité par

τ_i doit garder chacun de ses paquets jusqu'à obtenir un temps de réponse pire cas de bout-en-bout compris dans l'intervalle $[R_i^{\mathcal{L}_i} - J, R_i^{\mathcal{L}_i}]$.

9.2.4 Contrôle d'admission

Nous reprenons la propriété 7.3.5 établie page 119 et la maximisons afin d'avoir une faible complexité de calcul pour obtenir une borne sur le temps de réponse de bout-en-bout d'un flux EF. Nous déterminons ainsi la propriété suivante, dans laquelle nous notons $j \in EF$ si le flux τ_j , $j \in [1, n]$, appartient à la classe EF.

Propriété 9.2.1 Soit un flux quelconque τ_i de la classe EF suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds, numérotés de 1 à q . Si $\sum_{j \in EF, j \neq i} U_j^{slow_j} < 1$, alors le temps de réponse pire cas de bout-en-bout du flux τ_i , noté $R_i^{1,q}$, est inférieur ou égal à :

$$\sum_{\tau_j \in EF} \left(1 + \frac{S_{max_j}^{first_j} + J_j}{T_j} \right) \cdot C_j^{slow_j} + \sum_{\substack{h=1 \\ h \neq slow}}^q \max_{\tau_j \in EF} \{C_j^h\} + \sum_{h=1}^q \max \left(0; \max_{\tau_j \notin EF} C_j^h - 1 \right) + (q-1) \cdot Lmax.$$

Démonstration Par maximisation de l'expression de $R_i^{1,q}$ pour l'ordonnancement FP/FIFO* en environnement distribué (voir propriété 7.6.1), avec $hp_i = 0$ et pour tout $j \in EF$:

$$\left\lceil \frac{\min \left(t; W_i^{last_j}(t) - S_{min_j}^{last_j} \right) - M_i^{first_j}(t) + S_{max_j}^{first_j} + J_j}{T_j} \right\rceil^+ \leq \frac{t + S_{max_j}^{first_j} + J_j}{T_j}. \quad \blacksquare$$

A partir de cette borne sur le temps de réponse pire cas de bout-en-bout d'un flux EF, nous dérivons un contrôle d'admission pour la classe EF. Quand un nouveau flux EF arrive dans le domaine DiffServ, le contrôle d'admission doit décider s'il peut être accepté. Pour cela, le contrôle d'admission vérifie deux points, à savoir :

- l'acceptation du nouveau flux ne remettra pas en cause les garanties offertes aux flux déjà établis ;
- le temps de réponse de bout-en-bout du nouveau flux n'excédera pas D , l'échéance du domaine.

La conception d'un contrôle d'admission en-ligne pour la classe EF est une tâche complexe dans le sens où l'arrivée d'un nouveau flux EF ne doit pas conduire à recalculer les temps de réponse pire cas de bout-en-bout de tous les flux déjà acceptés. En effet, soit τ_j un flux déjà accepté. L'acceptation d'un nouveau flux τ_i entraîne l'augmentation de la durée de séjour de τ_j si τ_i croise τ_j dans le domaine. Par conséquent, la gigue de τ_j augmente également. L'impact de τ_j sur les flux qu'il croise devient donc plus important, ce qui augmente la durée de séjour de ces autres flux. Et ainsi de suite.

Pour éviter cet effet en cascade, nous définissons pour chaque flux EF τ_j suivant une ligne \mathcal{L}_j :

- d_j^h , la durée de séjour maximale garantie⁴ pour un paquet quelconque de τ_i dans le nœud h .

⁴La décomposition de l'échéance du domaine D en échéances intermédiaires est un problème général (voir par exemple [72]) qui est en dehors de la problématique traitée dans cette thèse.

Le contrôle d'admission utilise alors cette durée de séjour maximale garantie pour maximiser l'impact d'un flux EF quelconque sur un autre flux EF. Ainsi, lors de l'arrivée d'un nouveau flux τ_i pour la classe EF souhaitant suivre une ligne de diffusion \mathcal{L}_i , le contrôle d'admission vérifie que les quatre conditions présentées dans le tableau 9.1 sont vérifiées⁵.

TAB. 9.1: Conditions à vérifier pour l'acceptation d'un nouveau flux τ_i dans la classe EF

Condition de charge locale	$\forall h \in \mathcal{L}_i, U_{j \in EF}^h \leq 1$
Condition de charge distribuée	$\sum_{j \in EF, j \neq i} U_j^{slow_j} < 1$
Condition sur la durée de séjour	<ul style="list-style-type: none"> • $\forall h \in \mathcal{L}_i, R_i^h \leq d_i^h$ • $\forall \text{flux } \tau_j, j \in EF, \text{ suivant } \mathcal{L}_j, \forall h \in \mathcal{L}_j \cap \mathcal{L}_i, R_j^h \leq d_j^h$
Condition sur le temps de réponse de bout-en-bout	<ul style="list-style-type: none"> • $R_i^{\mathcal{L}_i} \leq D$ • $\forall \text{flux } \tau_j, j \in EF, \text{ suivant } \mathcal{L}_j \text{ avec } \mathcal{L}_j \cap \mathcal{L}_i \neq \emptyset, R_j^{\mathcal{L}_i} \leq D$

Nous précisons maintenant chacune de ces conditions.

Condition de charge locale

Lorsqu'un nouveau flux EF τ_i arrive dans le domaine DiffServ, la première condition à vérifier par le contrôle d'admission est que le facteur d'utilisation de chacun des nœuds visités par τ_i reste inférieur à 1 (voir remarque 3.2.1, page 26).

Condition de charge distribuée

Cette condition est nécessaire pour appliquer la propriété 9.2.1. Elle a été démontrée lors de l'étude du temps de réponse pire cas de bout-en-bout en environnement distribué dans le chapitre 7, page 117.

Condition sur la durée de séjour

Si les conditions de charge sont respectées, le contrôle d'admission vérifie que sur tout nœud $h \in \mathcal{L}_i$, la durée de séjour de chaque flux τ_j de la classe EF sur ce nœud reste inférieure ou égale à d_j^h . Si nous notons $J_{in_j}^h$ la gigue du flux τ_j en entrée du nœud h , alors d'après la propriété 5.6.1 établie page 67, nous avons pour tout flux $\tau_i, i \in EF$:

$$R_i^h \leq \sum_{j \in EF} \left(1 + \frac{J_{in_j}^h}{T_j} \right) \cdot C_j^h + \max_{j \notin EF} \left(0; \max \{ C_j^h \} - 1 \right) \quad (\text{avec } C_j^h = 0 \text{ si } \tau_j \text{ ne visite pas le nœud } h).$$

Pour tout flux τ_i suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds numérotés de 1 à q , nous avons : $J_{in_i}^h = S_{max_i}^h - S_{min_i}^h \leq \left(\sum_{h'=1}^{h-1} d_i^{h'} + (h-1) \cdot L_{max} \right) - \left(\sum_{h'=1}^{h-1} C_j^{h'} + (h-1) \cdot L_{min} \right)$.

⁵Pour tout flux τ_j suivant une ligne de diffusion \mathcal{L}_j , nous notons R_j^h sa durée de séjour maximale sur le nœud h et $R_j^{\mathcal{L}_j}$ son temps de réponse pire cas de bout-en-bout.

Condition sur le temps de réponse de bout-en-bout

Pour garantir un temps de réponse pire cas de bout-en-bout inférieur ou égal à l'échéance du domaine à tout flux de la classe EF, le contrôle d'admission doit vérifier que le temps de réponse de bout-en-bout de τ_i , mais aussi celui de chaque flux EF τ_j suivant une ligne de diffusion \mathcal{L}_j avec $\mathcal{L}_j \cap \mathcal{L}_i \neq \emptyset$, sont bornés par D , soit : $R_i^{\mathcal{L}_i} \leq D$ et $\forall j \in EF$ tel que $\mathcal{L}_j \cap \mathcal{L}_i \neq \emptyset$, $R_j^{\mathcal{L}_j} \leq D$.

Pour calculer le temps de réponse pire cas de bout-en-bout d'un flux de la classe EF, nous avons établi la propriété 9.2.1. Dans la borne proposée, le terme $S_{max_j}^{first_j}$ doit être évalué pour tout flux τ_j , $j \in EF$. Ce terme représente le temps maximum mis par un paquet de τ_j depuis son instant de génération sur son nœud source pour arriver sur le nœud $first_j$. Pour tout flux τ_i de la classe EF suivant une ligne de diffusion \mathcal{L}_i composée de q nœuds numérotés de 1 à q , nous avons :

$$S_{max_i}^h = \sum_{h'=1}^{h-1} R_i^{h'} + (h-1) \cdot L_{max} \leq \sum_{h'=1}^{h-1} d_i^{h'} + (h-1) \cdot L_{max}.$$

Remarque 9.2.1 *Lorsqu'un nouveau flux de la classe EF arrive dans le domaine DiffServ, le contrôle d'admission en-ligne présenté ci-dessus n'implique que le nouveau flux EF et les flux EF le croisant.*

Nous illustrons par un exemple le fonctionnement, étape par étape, du contrôle d'admission.

9.2.5 Exemple

Nous donnons ici un exemple de borne sur le temps de réponse de bout-en-bout établie dans la sous-section précédente. Nous considérons que trois flux, τ_1 , τ_2 and τ_3 , ont déjà été acceptés dans la classe EF. Comme le montre la figure 9.3 :

- τ_1 visite les nœuds 1, 2, 3 et 4 ;
- τ_2 visite les nœuds 5, 2, 3 et 6 ;
- τ_3 visite les nœuds 7, 2 et 8.

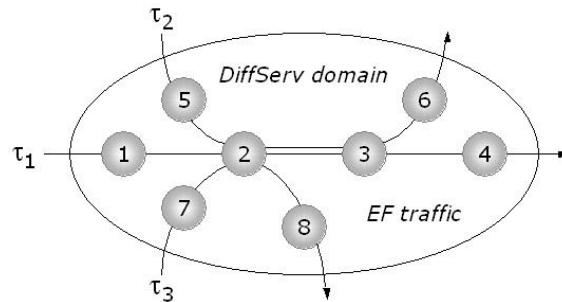


FIG. 9.3: Trafic EF dans le domaine DiffServ

Nous supposons qu'un nouveau flux, τ_4 , souhaite entrer dans le domaine et parcourir les nœuds 1, 2, 3 et 4.

Les hypothèses sont les suivantes. L'échéance du domaine est $D = 100$. Tous les flux τ_i , $i = 1..4$, ont une période $T_i = 10$, arrivent sans gigue dans le domaine et ont une durée de séjour maximale garantie égale

à 10 dans les routeurs d'entrée, 25 dans les routeurs au cœur du domaine et 40 dans les routeurs de sortie. De plus, le temps de traitement d'un paquet quelconque est égal à 3 dans les routeurs d'entrée et de sortie, 2 dans les routeurs au cœur du domaine. Enfin, $L_{max} = L_{min} = 1$.

1. Le contrôle d'admission vérifie que sur tout nœud $h = 1..4$, $U_{j \in EF}^h \leq 1$.

Puisque $U_{j \in EF}^1 = U_{j \in EF}^3 = U_{j \in EF}^4 = 0.6$ et $U_{j \in EF}^2 = 0.8$, Cette condition est satisfaite.

2. La condition de charge distribuée est également satisfaite. En effet :

$$\sum_{j \in EF, j \neq i} U_j^{slow_j} = \frac{C_4^1}{T_4} + \frac{C_2^2}{T_2} + \frac{C_3^2}{T_3} = 0,3 + 0,2 + 0,2 = 0,7 < 1.$$

3. Le contrôle d'admission doit alors vérifier que les durées de séjour des flux EF sur les nœuds communs à τ_4 restent bornées par leurs échéances intermédiaires. Cette condition est vérifiée. Par exemple, les durées de séjour maximales du flux τ_2 sur les nœuds 2 et 3 sont respectivement 15 et 25.

4. Enfin, la condition sur le temps de réponse de bout-en-bout doit être vérifiée par le contrôle d'admission. Par exemple, le temps de réponse pire cas de bout-en-bout du flux τ_4 est inférieur ou égal à :

$$\underbrace{\sum_{j \in EF} \left(1 + \frac{Smax_j^{first_j} + J_j}{T_j} \right) \cdot C_j^{slow_j}}_{C_1^1 + (1 + \frac{11}{10}) \cdot C_2^2 + (1 + \frac{11}{10}) \cdot C_3^2 + C_4^1 = 14,4} + \underbrace{\sum_{\substack{h=1 \\ h \neq slow}}^4 \max_{j \in EF} \{C_j^h\}}_7 + \underbrace{\sum_{h=1}^4 \max \left(0; \max_{j \notin EF} \{C_j^h - 1\} \right)}_6 + \underbrace{(q-1) \cdot L_{max}}_3.$$

La borne sur le temps de réponse pire cas de bout-en-bout du flux τ_4 est donc égale à 30 (le temps étant supposé discret).

Le temps de réponse pire cas de bout-en-bout exact (obtenu par un outil développé dans le cadre de cette thèse et présenté page 2.4) est égal à 26. Nous surestimons donc la valeur exacte d'environ 15%. Cela dit, il est important de remarquer que si les flux τ_2 et τ_3 sont gênés par de nouveaux flux ne croisant pas τ_4 , la valeur exacte du temps de réponse pire cas de bout-en-bout de τ_4 augmentera alors que la borne calculée restera inchangée. Par exemple, si deux nouveaux flux EF souhaitent visiter respectivement les nœuds 5 et 7, avec une période égale à 20 et un temps de traitement maximum égal à 7, ils seront acceptés par le contrôle d'admission. La valeur exacte du temps de réponse pire cas de bout-en-bout sera alors égale à 29, alors que la borne calculée précédemment restera égale à 30. La surestimation n'est plus que de l'ordre de 3%. La borne ainsi calculée pour le temps de réponse de τ_4 reste valide tant que les échéances intermédiaires des flux croisant τ_4 sont vérifiées. Par ailleurs, il est également important de souligner que la charge obtenue dans notre exemple pour la classe EF atteint 80%.

9.2.6 Discussion

9.2.6.a Classes et priorités fixes

Nous avons considéré pour l'établissement du contrôle d'admission qu'à chaque classe correspondait une priorité fixe. Tous les flux d'une classe étaient donc traités DP* puisqu'ils partageaient la même priorité fixe. Il est possible de généraliser en permettant à des flux ayant des degrés d'importance différents (donc des priorités fixes différentes) d'appartenir à une même classe de service. La seule exigence à respecter est que les

priorités fixes attribuées à la classe EF soient plus grandes que celles attribuées aux classes AF, elles-mêmes plus grandes que celles attribuées à la classe *best-effort*.

9.2.6.b Garanties apportées aux autres classes

Nous avons vu dans les sous-sections précédentes comment borner le temps de réponse de bout-en-bout d'un flux de la classe EF. Il est également possible d'utiliser les résultats établis dans cette thèse pour borner le temps de réponse de bout-en-bout d'un flux d'une autre classe (par exemple AF ou *best-effort*), sous réserve que les flux soient ordonnancés FP/DP*. Un contrôle d'admission similaire à celui présenté ci-dessus pourra être dérivé. Pour cela, il suffit de reprendre la propriété 7.3.5 donnant une borne sur le temps de réponse pire cas de bout-en-bout d'un flux, en notant l'existence de flux plus prioritaires que les flux considérés (ce qui n'était pas le cas pour la classe EF).

9.3 AUTRES ARCHITECTURES

9.3.1 Architecture IntServ

Le modèle de QoS IntServ a été présenté dans la section 3.4.1, page 37. Dans une telle architecture, la qualité de service est gérée par flux. L'admission d'un nouveau flux est effectuée lors de la phase de réservation des ressources. Les résultats établis dans cette thèse peuvent être appliqués dès lors que les flux sont ordonnancés FP/DP* dans les nœuds traversés. Dans ce cas, la priorité fixe d'un flux est fonction de son degré d'importance ; sa priorité dynamique est elle fonction de son paramètre temporel. Le contrôle d'admission s'appuiera alors sur la borne obtenue sur le temps de réponse de bout-en-bout (voir propriété 7.3.5).

9.3.2 Architecture hybride

Dans une architecture hybride, la QoS est gérée soit par classe (type DiffServ), soit par flux (type IntServ). Il est également possible dans une telle architecture d'offrir des garanties déterministes sur le temps de réponse et la gigue de bout-en-bout des flux. En effet, les résultats établis dans cette thèse peuvent s'appliquer dès lors que les flux sont ordonnancés FP/DP* dans chacun des nœuds visités. Dans cette architecture, gérer la QoS par flux peut revenir, du point de vue du calcul des temps de réponse, à gérer la QoS par classe, en considérant que ce flux est le seul à appartenir à cette classe.

9.4 CONCLUSION

Dans ce chapitre, nous avons proposé une solution basée sur DiffServ et MPLS pour fournir à des applications temps-réel des garanties déterministes sur le temps de réponse et la gigue de bout-en-bout. Dans ce cadre, nous avons montré la mise en œuvre d'un contrôle d'admission en nous basant sur les résultats établis dans le chapitre 7. Afin de réduire la complexité du contrôle d'admission, permettant ainsi une exécution en ligne, nous avons majoré l'impact des flux croisant le flux demandant son admission.

Nous avons ensuite montré comment étendre ces résultats à d'autres architectures de QoS (architecture IntServ et architecture hybride).

Quatrième partie

Conclusion

CHAPITRE 10

Conclusion et perspectives

10.1 Conclusion	160
10.2 Perspectives	161

10.1 CONCLUSION

Dans cette thèse, nous nous sommes intéressés à des applications ayant des contraintes fortes en termes de temps de réponse et de gigue de bout-en-bout dans un réseau de type Internet. En effet, les nouvelles applications telles que la téléphonie sur IP, la vidéo à la demande ou encore les jeux interactifs distribués ont besoins de garanties temporelles de QoS pour fonctionner correctement. A partir de deux paramètres de QoS spécifiés par l'utilisateur, nous avons proposé une solution pour garantir de manière déterministe des bornes sur le temps de réponse et la gigue de bout-en-bout du flux considéré. Les deux paramètres sont les suivants :

- un degré d'importance, représentant la criticité du flux du point de vue utilisateur ;
- un paramètre temporel, utilisé pour départager les flux de même degré d'importance.

Notre solution consiste en un modèle d'ordonnancement à base de priorités fixes (FP) combiné à un ordonnancement à base de priorité dynamiques (DP). La priorité fixe est le critère principal d'ordonnancement, la priorité dynamique ne servant qu'à départager les flux de même priorité fixe. Ainsi, un paquet entrant dans le réseau se voit assigner la priorité fixe du flux auquel il appartient et une priorité dynamique. Il est ensuite ordonné dans le réseau selon ses priorités, attribuées sur son nœud d'entrée. Nous avons noté cet ordonnancement FP/DP*. Ce principe correspond à la tendance actuelle qui consiste à reporter les traitements complexes dans les nœuds d'entrée ou de sortie du réseau, afin d'améliorer les performances des routeurs au cœur du réseau.

Nous avons alors réalisé une analyse pire cas pour établir des bornes mathématiquement calculables sur les temps de réponse et les giges de bout-en-bout de n flux coexistant dans un réseau. Pour cela, nous avons adopté une approche dite "par trajectoire", qui ne considère que des scénarios possibles.

L'établissement de ces bornes a été effectué successivement dans trois configurations de complexité croissante :

- cas monoprocesseur ;
- cas d'une simple ligne de diffusion : tous les flux suivent la même séquence de nœuds ;
- cas général distribué : les flux suivent des lignes quelconques.

Puis, nous avons appliqué nos résultats à deux ordonnancements FP/DP* particuliers : FP/FIFO* et FP/EDF*. En considérant les cas où (i) les flux ont tous une priorité fixe différente et (ii) les flux ont tous la même priorité fixe, nous avons également établis des résultats pour les ordonnancements FP, FIFO* et EDF*. Nous avons alors montré que nos résultats améliorent les résultats existants en contexte monoprocesseur pour les ordonnancements FIFO et EDF et que nous apportons de nouveaux résultats en environnement distribué.

Pour chaque configuration étudiée (monoprocesseur, ligne de diffusion et cas général distribué), les résultats obtenus avec l'approche par trajectoire améliorent significativement les résultats obtenus par l'approche holistique. Par ailleurs, nous avons évalué la précision de nos résultats par des exemples numériques en les comparant avec les valeurs exactes, fournies par un outil de validation que nous avons développé.

Ensuite, nous avons analysé l'impact d'une remise en forme d'un flux sur son temps de réponse et sa gigue de bout-en-bout. Pour cela, nous avons considéré deux techniques de remise en forme : l'annulation de gigue et le seau à jetons. Après avoir établi un ensemble de propriétés portant notamment sur le délai introduit par un module de remise en forme et le temps de réponse pire cas de bout-en-bout en résultant, nous avons discuté (i) l'intérêt d'une remise en forme et (ii) des mérites respectifs des deux techniques de remise en forme étudiées.

Enfin, nous avons présenté une application immédiate de nos résultats en proposant une solution qui permet de pallier l'absence de garanties déterministes du service EF dans le modèle DiffServ. En effet, dans une architecture combinant DiffServ et MPLS, nous avons dérivé un contrôle d'admission à partir des résultats établis dans cette thèse, permettant de garantir de manière déterministe les temps de réponse et les giges de bout-en-bout des flux de la classe EF. Cette classe a été développée pour des applications temps-réel. Nous avons ensuite montré comment généraliser à d'autres architectures de QoS (architecture IntServ et architecture hybride).

10.2 PERSPECTIVES

A l'issue de cette thèse, plusieurs perspectives pourraient être investiguées. Elles concernent quatre domaines spécifiques, à savoir :

- **Ordonnancement**

Les résultats établis dans cette thèse permettent de calculer des temps de réponse pire cas en contexte monoprocesseur, dans le cas d'une simple ligne de diffusion et en environnement distribué, pour des ordonnancements de type FP/DP^{*}¹. Dans ce cadre, trois extensions seraient intéressantes :

- *Comparaison de FP/FIFO^{*} et FP/EDF^{*}*

Deux ordonnancements FP/DP^{*} particuliers ont été étudiés : FP/FIFO^{*} et FP/EDF^{*}. Nous avons montré qu'en monoprocesseur, FP/EDF domine FP/FIFO lorsque les flux partageant la même priorité fixe ont même temps de traitement maximum. Nous aimerions identifier l'ensemble des cas pour lesquels cette propriété est vérifiée. Nous aimerions également comparer FP/FIFO^{*} et FP/EDF^{*} dans le cas d'une ligne de diffusion, puis dans le cas général distribué.

- *Attribution d'une échéance locale pour FP/EDF^{*} et FP/EDF*

Dans le cadre de cette thèse, nous n'avons pas développé de méthode d'attribution d'une échéance locale à partir d'une échéance de bout-en-bout. Ce problème a été formellement étudié dans [85]. La méthode la plus simple que nous avons adoptée pour les exemples est la répartition équitable : l'échéance locale est égale à l'échéance de bout-en-bout divisée par le nombre de nœuds visités. Il serait intéressant de comparer cette méthode avec d'autres méthodes, telles que l'attribution proportionnelle à la charge.

- *Comparaison de FP/DP^{*} et FP/DP*

Pour une méthode d'attribution d'échéances fixée, nous souhaiterions comparer en environnement distribué les temps de réponse pire cas obtenus avec (i) un ordonnancement de type FP/DP^{*} et (ii) un ordonnancement de type FP/DP et identifier les cas pour lesquels l'un des deux ordonnancements domine l'autre.

Par ailleurs, tout au long de cette thèse, nous avons confronté les résultats obtenus par analyse pire cas avec ceux fournis par un outil donnant la solution exhaustive à un problème d'ordonnancement donné. Nous nous sommes rapidement heurtés à un problème de complexité exponentielle en pire cas.

- *Complexité de calcul de l'outil de validation*

L'outil de validation que nous avons développé dans le cadre de cette thèse génère tous les scénarios d'activations possibles et mémorise le temps de réponse pire cas pour chacun des flux. Cet outil nous a ainsi permis de valider l'ensemble de nos résultats et d'évaluer leur précision. Malheureusement, la complexité de calcul interdit l'utilisation de cette approche exhaustive dans le cadre d'un nombre élevé de flux. Pour y remédier, il serait nécessaire de ne tester que les scénarios pire cas parmi l'ensemble des scénarios possibles. Pour ce faire, il est nécessaire d'établir les propriétés permettant de caractériser les scénarios pire cas.

¹Dans le cas monoprocesseur, un ordonnancement FP/DP^{*} équivaut à un ordonnancement FP/DP.

- **Network Calculus**

Nous avons présenté dans l'état de l'art de cette thèse l'approche Network Calculus et les résultats de base associés. Les résultats que nous avons établis l'ont été avec une approche par trajectoire selon une analyse pire cas, offrant l'avantage d'un formalisme plus concret. Il serait intéressant d'étudier comment nos résultats pourraient être intégrés dans l'approche Network Calculus. En effet, le modèle sporadique peut être considéré comme un cas particulier de courbe d'arrivée.

- **Garanties probabilistes**

La thématique de cette thèse porte sur les garanties déterministes dans un réseau accordées à des flux ayant des contraintes temporelles. Pour ce faire, nous avons réalisé une analyse pire cas sur le temps de réponse de bout-en-bout. Un contrôle d'admission a alors été dérivé. Mais offrir une garantie déterministe peut coûter cher, surtout lorsque le pire cas ne se produit que très rarement. Dans ces conditions, des garanties probabilistes peuvent s'avérer suffisantes. Plus généralement, il est intéressant de connaître le rapport entre la probabilité de garantie d'un temps de réponse et son coût (inversement proportionnel au nombre de flux maximum acceptables). De même, la distribution des temps de réponse peut fournir des informations pertinentes pour l'administrateur du réseau [86].

- **QoS dans les réseaux sans-fil**

Nous nous sommes intéressés à la gestion de la qualité de service dans les réseaux filaires. Ce problème se pose avec plus d'acuité dans les réseaux sans-fil. En effet, ceux-ci se distinguent par leur très grande dynamique (changements de topologie très fréquents, dûs en partie à la mobilité) et par la limitation des ressources (bande passante, énergie). Dans ce contexte, il est illusoire de vouloir fournir une garantie déterministe. Cela dit, il est recommandé d'utiliser des techniques d'ordonnancement prenant en compte les exigences de QoS fournies par l'utilisateur, d'où l'intérêt d'un ordonnancement de type FP/DP. En effet, cette solution permet de caractériser un besoin de QoS suivant deux paramètres simples : un degré d'importance et un paramètre temporel. Notons que le contrôle d'admission à mettre en œuvre est plus complexe que celui présenté dans cette thèse, en raison des interférences des différents flux.

INDEX

Index

A

- annulation de gigue, 135
- approche
 - exhaustive 22
 - holistique 33
 - Network Calculus 33
 - par trajectoire 82

C

- contrôle d'admission
 - définition 42
 - exemple dans DiffServ/MPLS 151

D

- degré d'importance, 17
- DiffServ
 - architecture 40
 - classe AF 42
 - classe EF 42
 - contrôle d'admission 148
- distribué
 - définition 21
 - notations 51
 - nouveaux résultats 107

E

- Echéance, 16
 - absolue 20
 - intermédiaire 151
 - relative 20
- EDF
 - définition 20
 - nouveaux résultats
 - distribué, 126
 - ligne de diffusion, 102
 - monoprocasseur, 72
 - résultats connus 30
- effet non-préemptif, 27

F

- FIFO
 - définition 20
 - nouveaux résultats
 - distribué, 124
 - ligne de diffusion, 98
 - monoprocasseur, 68
 - résultats connus 29
- flux
 - définition 14
 - sporadicité 16
- FP
 - définition 19
 - nouveaux résultats
 - distribué, 120
 - ligne de diffusion, 94
 - monoprocasseur, 64
 - résultats connus 28

G

- garantie
 - déterministe 5
 - temporelle 15
- gigue
 - activation 16
 - annulation 135
 - définition 15

I

- instant
 - arrivée 16
 - démarrage au plus tard 49
 - génération 16
 - oisif 49
- IntServ
 - architecture 37
 - service contrôlé 39
 - service garanti 39

L

- label, 148
- ligne de diffusion
 - définition 21
 - notations 50
 - nouveaux résultats 81

M

- monoprocasseur
 - définition 21
 - notations 49
 - nouveaux résultats 53
 - résultats connus 26
- MPLS, 148

O

- ordonnancement
 - définition 17
 - DP : priorités dynamiques 20
 - EDF 20
 - FIFO 20
 - FP : priorités fixes 19
 - FP/DP 18
 - FP/DP* 18
 - oisif 17
 - préemptif 17
 - priorités généralisées 18
- outil de validation, 22

P

- période active, 49
- paramètre temporel, 17
- priorité dynamique, 17
- priorité fixe, 17
- priorité généralisée, 18

Q

- qualité de service (QoS)
 - définition 5
 - dimension temporelle 15
 - paramètres 5

R

- remise en forme, 131
 - annulation de gigue 135
 - seau à jetons 136
- RSVP, 38

S

- seau à jetons, 136

T

- temps de réponse, 15

Glossaire

Annulation de gigue Technique de remise en forme.

Application temps-réel Application ayant des contraintes fortes en termes de délai de bout-en-bout.

Approche holistique Approche considérant le scénario pire cas dans chacun des nœuds visités par un flux.

Approche Network Calculus Approche considérant des courbes d'arrivée et de service.

Approche par trajectoire Approche considérant les scénarios pire cas pouvant se produire.

Contrôle d'admission Mécanisme permettant de s'assurer du bon respect des garanties contractées.

Degré d'importance Paramètre représentant la criticité du flux du point de vue utilisateur.

DiffServ Architecture de QoS pour la différenciation de services, utilisant un nombre limité de classes.

Echéance Temps de réponse maximal acceptable.

EDF Algorithme d'ordonnancement traitant les paquets dans l'ordre de leur échéance absolue.

EF Classe de service la plus prioritaire dans le domaine DiffServ.

DP Algorithme d'ordonnancement traitant les paquets dans l'ordre de leur priorité dynamique.

Faisabilité Respect de l'échéance d'un flux ou d'un ensemble de flux.

FIFO Algorithme d'ordonnancement traitant les paquets dans l'ordre de leur arrivée.

Flux Séquence de paquets ayant des caractéristiques communes.

FP Algorithme d'ordonnancement traitant les paquets dans l'ordre de leur priorité fixe.

FP/DP Ordonnancement FP (priorités fixes) combiné à un ordonnancement DP (priorités dynamiques).

Garantie déterministe Respect sans faute des contraintes imposées par le service demandé.

Gigue Différence entre les temps de réponse maximum et minimum.

IntServ Architecture de QoS basée sur la réservation des ressources.

Ligne de diffusion Séquence des nœuds visités par un flux dans le réseau.

MPLS Technologie permettant de commuter les paquets en fonction des labels qu'ils contiennent.

Ordonnancement Algorithme déterminant le prochain paquet à traiter parmi ceux en attente.

Paramètre temporel Paramètre utilisé pour départager des flux de même degré d'importance.

Préemption Le traitement d'un paquet peut être suspendu pour traiter un paquet plus prioritaire.

Priorité dynamique Priorité utilisée pour départager des flux de même priorité fixe.

Priorité fixe Priorité associée au degré d'importance du flux. Critère principal d'ordonnancement.

Priorité généralisée Priorité d'un paquet représentant ses priorités fixes et dynamiques.

Qualité de service Capacité d'un réseau à fournir un service adapté aux besoins spécifiques des applications.

Région d'ordonnançabilité Paramètres pour lesquels un ensemble de flux est faisable.

Remise en forme Mécanisme permettant de rendre un flux conforme à sa spécification ou à ses exigences.

Seau à jetons Technique de remise en forme.

Temps de réponse de bout-en-bout Temps mis par un paquet pour traverser le réseau.

Liste des publications

REVUES INTERNATIONALES

- S. Martin, P. Minet, L. George. *Deterministic end-to-end guarantees for real-time applications in a DiffServ-MPLS domain*. LNCS (Lecture Notes in Computer Science), Springer-Verlag. To be published in 2004.
- S. Martin, P. Minet, L. George. *End-to-end response time with Fixed Priority scheduling : trajectory approach versus holistic approach*. IJCS (International Journal of Communication Systems), Wiley. In revision process.

CONFÉRENCES INTERNATIONALES

- S. Martin, P. Minet, L. George. Non-preemptive Fixed Priority scheduling with Earliest Deadline First: The distributed case. *SERA'04*, Los Angeles, USA, May 2004.
- S. Martin, P. Minet, L. George. Deterministic QoS guarantee for the EF class with and without traffic shaping. *ICN'04*, Pointe-à-Pitre, France, March 2004.
- S. Martin, P. Minet, L. George. A DiffServ-MPLS solution for real-time applications with deterministic end-to-end guarantees. *SERA'03*, San Francisco, USA, June 2003.
- L. George, S. Martin, P. Minet. End-to-end real-time guarantees for the EF traffics in the DiffServ model. *SNPD'02*, Madrid, Spain, June 2002.
- L. George, S. Martin, P. Minet. Ordonnancement FIFO et garantie déterministe de qualité de service en environnement distribué. *RTS'02*, Paris, France, March 2002. Prix de la meilleure communication.

RAPPORTS DE RECHERCHE

- S. Martin, P. Minet, L. George. FP/EDF, a non-preemptive scheduling combining fixed priorities and deadlines : uniprocessor and distributed cases. *INRIA Research Report*, No. 5112, February 2004.
- S. Martin, P. Minet, L. George. Fixed Priority scheduling with FIFO arbitration for the same priority: uniprocessor and distributed cases. *INRIA Research Report*, No. 5051, December 2003.
- S. Martin, P. Minet, L. George. Real-time end-to-end guarantees for the EF class with and without traffic shaping. *INRIA Research Report*, No. 4980, October 2003.
- S. Martin, P. Minet, L. George. A DiffServ-MPLS solution offering real-time end-to-end guarantees. *INRIA Research Report*, No. 4831, May 2003.

Bibliographie

- [1] R. Guérin and V. P  ris. Quality-of-service in packet networks : basic mechanisms and directions. *Computer Networks*, Vol. 31, No. 3, pp. 169-179, February 1999.
- [2] X. Xiao and L. M. Ni. Internet QoS : A big picture. *IEEE Network*, Vol. 13, No. 2, pp. 8-18, April 1999.
- [3] W. Zhao, D. Olshefski, and H. Schulzrinne. Internet quality of service : An overview. *Technical Report CUCS-003-00*, Columbia University, February 2000.
- [4] J. Forgie. ST - A proposed Internet stream protocol. *Internet experimental notes, IEN-119*, September 1979.
- [5] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture : an Overview. *RFC 1633*, June 1994.
- [6] L. Zhang, S. Deering, S. Estrin, et al. RSVP : A new Resource ReServation Protocol. *IEEE Network*, Vol. 7, No. 5, pp. 8-18, September 1993.
- [7] S. Blake, D. Black, M. Carlson, et al. An Architecture for Differentiated Services. *RFC 2475*, December 1998.
- [8] P. Ferguson and G. Huston. *Quality of Service - Delivering QoS on the Internet and in Corporate Networks*. Wiley, 1998.
- [9] J. L. M  lin. *Qualit   de Service sur IP*. Eyrolles, 2001.
- [10] P. Goyal, A. Greenberg, C. Kalmanek, et al. Integration of call signaling and resource management for IP telephony. *IEEE Network*, Vol. 13(3), pp. 24-32, May-June 1999.
- [11] International Telecommunication Union (ITU). Transmission systems and media, general recommendation on the transmission quality for an entire international telephone connection ; one-way transmission time. *Recommendation G.114*, Geneva, Switzerland, March 1993.
- [12] D. Miras. Network QoS needs of advanced internet applications : A survey. *Internet2 QoS working document*, November 2002.
- [13] G. Armitage. *Quality of service in IP networks : Foundations for a multi-service Internet*. Macmillan Technical Publishing, September 2000.
- [14] N. Bouillot. The auditory consistency in distributed music performance : A conductor based synchronization. *ISDM*, Vol. 13, pp. 129-137, 2004.
- [15] K. Skedd. Consumer Oriented Video-On-Demand Via IP Networks. *InStat Report, No. IN020022MB*, Scottsdale, USA, June 2002.
- [16] D. Marinca. Conception et dimensionnement d'un syst  me multim  dia hautes performances. *Ph.D. thesis*, University of Versailles St-Quentin, 2003.
- [17] B. Fouquet. *Gestion de la qualit   de service*. Eyrolles, 2000.

- [18] K. Jeffay, D. F. Stanat, and C. U. Martel. On non-preemptive scheduling of periodic and sporadic tasks. *IEEE Real-Time Systems Symposium*, pp. 129-139, San Antonio, USA, December 1991.
- [19] S. Shenker and J. Wroclawski. General characterization parameters for integrated service network elements. *RFC 2215*, September 1997.
- [20] L. George, D. Marinca, and P. Minet. A solution for a deterministic QoS in multimedia systems. *International Journal on Computer and Information Science*, Vol. 1, No. 3, September 2001.
- [21] S. Shenker, C. Partridge, and R. Guérin. Specification of Guaranteed quality of service. *RFC 2212*, September 1997.
- [22] F. Cottet, J. Delacroix, and Z. Mammeri. *Scheduling in real-time systems*. John Wiley & Sons, October 2002.
- [23] A. K. Mok. Fundamental design problems for the hard real-time environments. *MIT Ph.D. Dissertation*, May 1983.
- [24] K. H. Kim and M. Naghibzadeh. Prevention of task overruns in real-time non-preemptive multiprogramming systems. *Proceedings of the 1980 international symposium on Computer performance modelling, measurement and evaluation*, pp. 267-276, 1980.
- [25] D. B. Stewart and P. K. Khosla. Real-time scheduling of sensor-based control systems. *8th IEEE workshop on real-time operating systems*, Atlanta, USA, May 1991.
- [26] S. Martin, P. Minet, and L. George. A DiffServ-MPLS solution for real-time applications with deterministic end-to-end guarantees. *SERA'03*, San Francisco, USA, June 2003.
- [27] K. Tindell, A. Burns, and A. J. Wellings. Analysis of hard real-time communications. *Real-Time Systems*, Vol. 9, pp. 147-171, 1995.
- [28] J. Liu. *Real-time systems*. Prentice Hall, 2000.
- [29] L. George, S. Kamoun, and P. Minet. First come first served : some results for real-time scheduling. *PDCS'2001*, Dallas, USA, August 2001.
- [30] L. C. Liu and J. W. Wayland. Scheduling algorithms for multi-programming in a hard real time environment. *Journal of ACM*, pp. 46-61, January 1973.
- [31] S. Baruah, R. Howell, and L. Rosier. Algorithms and complexity concerning the preemptive scheduling of periodic real-time tasks on one processor. *Real-Time Systems*, Vol. 2, pp. 301-324, 1990.
- [32] K. Tindell, A. Burns, and A. J. Wellings. Calculating Controller Area Network (CAN) message response times. *IFAC DCCS'94*, Toledo, Spain, 1994.
- [33] L. George, N. Rivierre, and M. Spuri. Preemptive and non-preemptive scheduling real-time uniprocessor scheduling. *INRIA Research Report*, No. 2966, September 1996.
- [34] K. Tindell, H. Hansson, and A. J. Wellings. Analysing real-time communications : Controller Area Networks (CAN). *Real-Time Systems*, pp. 259-263, 1994.
- [35] L. George. Ordonnancement en-ligne temps réel critique dans les systèmes distribués. *Ph.D. thesis*, University of Versailles St-Quentin, 1998.
- [36] D. L. Levine, C. D. Gill, and D. C. Schmidt. Dynamic scheduling strategies for avionics mission computing. *IEEE ACM Transactions on Networking*, 17th IEEE/AIAA Digital Avionics Systems Conference, Seattle, USA, November 1998.
- [37] M. González Harbour and J. C. Palencia. Response time analysis for tasks scheduled under EDF within Fixed Priorities. *IEEE Real-Time Systems Symposium*, Cancun, Mexico, December 2003.

- [38] J. Liebeherr, D. Wrege, and D. Ferrari. Exact admission control for networks with a bounded delay service. *IEEE ACM Transactions on Networking*, Vol. 4, No. 6, pp. 885-890, December 1996.
- [39] V. Sivaraman, F. Chiussi, and M. Gerla. End-to-end statistical delay service under GPS and EDF scheduling : a comparison study. *INFOCOM'01*, Anchorage, USA, April 2001.
- [40] M. Vojnovic and J. Y. le Boudec. Stochastic analysis of some expedited forwarding networks. *INFOCOM'02*, New York, USA, June 2002.
- [41] F. M. Chiussi and V. Sivaraman. Achieving high utilization in guaranteed services networks using Early-Deadline-First scheduling. *IWQoS'98*, Napo, USA, May 1998.
- [42] K. Tindell and J. Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocessors and Microprogramming, Euromicro Journal*, Vol. 40, 1994.
- [43] J. Y. le Boudec and P. Thiran. *Network calculus : A theory of deterministic queuing systems for the Internet*. LNCS 2050, Springer-Verlag, September 2003.
- [44] C. S. Chang. *Performance guarantees in communication networks*, volume New York. Springer-Verlag, 2000.
- [45] F. Kelly, A. Mauloo, and D. Tan. Rate control for communication networks : Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, No. 49, 1998.
- [46] S. Keshav. *Computer networking : An engineering approach*. Prentice Hall, 1996.
- [47] R. L. Cruz. A calculus for network delay, Part I : Network elements in isolation. *IEEE Transactions on Information Theory*, Vol. 37, No. 1, pp. 114-131, January 1991.
- [48] A. Koubâa and Y. Q. Song. Upper bound evaluation of response time for real-time communication. *LORIA Research Report*, 2002.
- [49] L. Georgiadis, R. Guérin, V. Peris, et al. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, August 1996.
- [50] V. Sivaraman, F. Chiussi, and M. Gerla. Traffic shaping for end-to-end delay guarantees with EDF scheduling. *IWQoS'2000*, Pittsburgh, USA, 2000.
- [51] J. White and J. Crowcroft. Integrated Services in the Internet : State of the Art. *Proceedings of the IEEE*, Vol. 85(12), December 1997.
- [52] J. Wroclawski. Specification of the Controlled-Load network element service. *RFC 2211*, September 1997.
- [53] D. Grossman. New terminology for DiffServ. *Internet draft*, March 2001.
- [54] J. Heinanen, F. Baker, W. Weiss, et al. Assured forwarding PHB group. *RFC 2597*, June 1999.
- [55] V. Jacobson, K. Nichols, and K. Poduri. An expedited forwarding PHB. *RFC 2598*, June 1999.
- [56] R. Braden, L. Zhang, S. Berson, et al. Resource ReSerVation Protocol (RSVP)- version 1 functional specification. *RFC 2205*, September 1997.
- [57] J. Wroclawski. The use of RSVP with IETF Integrated Services. *RFC 2210*, September 1997.
- [58] S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Trans. Networking*, Vol. 1(4), pp. 397-413, 1993.
- [59] M. May, J. Bolot, A. Jean-Marie, et al. Simple performance models of differentiated services shemes for the internet. *IEEE INFOCOM*, New York, USA, March 1999.
- [60] C. Aurrecoechea, A. Campbell, and L. Hauw. A survey of QoS architectures. *Multimedia Systems*, Springer-Verlag, No. 6, pp. 138-151 1998.

- [61] K. Nichols, J. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the Internet. *IETF Diffserv working group*, November 1997.
- [62] R. Guérin, H. Ahmadi, and M. Nagshineh. Equivalent capacity and its Application to bandwidth allocation in high-speed networks. *IEEE JSAC*, Vol. 9, pp. 968-981, September 1991.
- [63] E. Knightly and N. Shroff. Admission control for statistical QoS : Theory and practice. *IEEE Network*, 1999.
- [64] S. Jamin, P. Danzig, and S. Shenker. A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Transactions on Networking*, Vol. 5, No. 1, pp. 56-70, February 1997.
- [65] S. Martin, P. Minet, and L. George. A DiffServ-MPLS solution offering real-time end-to-end guarantees. *INRIA Research Report*, No. 4831, May 2003.
- [66] S. Baruah, A. K. Mok, and L. Rosier. Preemptively scheduling hard real-time sporadic tasks on one processor. *Proceedings of the 11th Real-Time Systems Symposium*, pp. 182-190, 1990.
- [67] S. Martin, P. Minet, and L. George. Non-preemptive Fixed Priority scheduling with FIFO arbitration for the same priority : Uniprocessor and distributed cases. *INRIA Research Report*, No. 5051, December 2003.
- [68] S. Martin, P. Minet, and L. George. FP/EDF, a non-preemptive scheduling combining fixed priorities and deadlines : Uniprocessor and distributed cases. *INRIA Research Report*, No. 5112, February 2004.
- [69] J. Y. le Boudec and P. Thiran. A note on time and space methods in network calculus. *Technical Report*, No. 97/224, Ecole Polytechnique Fédérale de Lausanne, Suisse, April 1997.
- [70] L. George, S. Martin, and P. Minet. Ordonnancement FIFO et garantie déterministe de qualité de service en distribué. *RTS'02*, Paris, France, March 2002.
- [71] S. Martin, P. Minet, and L. George. Non-preemptive Fixed Priority scheduling with Earliest Deadline First : The distributed case. *SERA'04*, Los Angeles, USA, May 2004.
- [72] D. Raz and Y. Shavitt. Optimal partition of QoS requirements with discrete cost functions. *INFOCOM'00*, Tel-Aviv, Israel, March 2000.
- [73] S. Martin, P. Minet, and L. George. Real-time end-to-end guarantees for the EF class with and without traffic shaping. *INRIA Research Report*, No. 4980, October 2003.
- [74] S. Martin, P. Minet, and L. George. Deterministic QoS guarantee for the EF class with and without traffic shaping. *ICN'04*, Pointe-à-Pitre, France, March 2004.
- [75] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks : the multiple node case. *IEEE ACM Transactions on Networking*, Vol. 2, No. 2, April 1994.
- [76] L. George and P. Minet. A FIFO worst case analysis for a hard real-time distributed problem with consistency constraints. *ICDCS'97*, Baltimore, USA, May 1997.
- [77] J. Bennett, K. Benson, A. Charny, et al. Delay jitter bounds and packet scale rate guarantee for Expedited Forwarding. *INFOCOM'01*, Anchorage, USA, April 2001.
- [78] A. Charny and J. Y. le Boudec. Delay bounds in a network with aggregate scheduling. *QoFIS*, Berlin, Germany, October 2000.
- [79] M. Gerla, C. Casetti, S. Lee, et al. Resource allocation and admission control styles on QoS DiffServ networks. *LNCSS, Springer-Verlag*, Roma, Italy, 2001.

- [80] E. Rosen, A. Viswanathan, and R. Callon. MultiProtocol Label Switching architecture. *RFC 3031*, January 2001.
- [81] D. Awduche, J. Malcolm, J. Agogbua, et al. Requirements for traffic engineering over MPLS. *RFC 2702*, September 1999.
- [82] D. Awduche, L. Berger, D. Gan, et al. RSVP-TE : Extensions to RSVP for LSP Tunnels. *Internet draft*, August 2001.
- [83] F. Le Faucheur, L. Wu, S. Davari, et al. MPLS Support of Differentiated Services. *Internet draft*, August 2000.
- [84] S. Martin, P. Minet, and L. George. Deterministic end-to-end guarantees for real-time applications in a DiffServ-MPLS domain. 2004.
- [85] D. Marinca, P. Minet, and L. George. Analysis of deadline assignment methods in distributed real-time systems. *Computer Communications*, Elsevier, To appear, 2004.
- [86] L. Azouz Saidane. Modélisation et étude analytique des performances d’algorithmes d’ordonnancement dans un système transactionnel réparti temps-réel. *Thesis*, Univerity of Tunis, February 2002.