

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Réseaux, Télécommunications, Systèmes et Architecture

Présentée et soutenue par :

M. RENAUD SALLANTIN

le lundi 29 septembre 2014

Titre :

OPTIMISATION DE BOUT-EN-BOUT DU DEMARRAGE DES CONNEXIONS TCP

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (I.R.I.T.)

Directeur(s) de Thèse :

M. ANDRE LUC BEYLOT

M. EMMANUEL CHAPUT

Rapporteurs :

M. KONSTANTIN AVRACHENKOV, INRIA SOPHIA ANTIPOLIS

M. VANIA CONAN, THALES

Membre(s) du jury :

M. ANDRZEJ DUDA, INP GRENOBLE, Président

M. ANDRE LUC BEYLOT, INP TOULOUSE, Membre

M. CÉDRIC BAUDOIN, THALES ALENIA SPACE, Membre

M. EMMANUEL CHAPUT, INP TOULOUSE, Membre

M. EMMANUEL DUBOIS, CENTRE NATIONAL D'ETUDES SPATIALES CNES, Membre

M. MARCELO DIAS DE AMORIM, UNIVERSITE PARIS 6, Membre

Remerciements

Je tiens tout d'abord à remercier le directeur de cette thèse, M.André-Luc Beylot, pour avoir cru en moi et être venu me chercher jusqu'en Amérique, puis pour m'avoir guidé, encouragé, poussé pendant presque trois ans tout en me laissant une grande liberté et en me faisant l'honneur de me déléguer plusieurs responsabilités dont j'espère avoir été à la hauteur.

Mes remerciements vont également à M.Emmanuel Chaput, qui a su m'accorder sa confiance et me challenger par ses piques régulières, mais également me guider tout au long de ces trois années.

Au moment de faire le bilan de cette thèse, je souhaite les remercier très sincèrement. Leur investissement personnel et professionnel remarquable a permis la réalisation et le succès de cette thèse.

Je remercie également MM. Emmanuel Dubois et Patrick Gélard pour m'avoir soutenu et permis notamment de tirer le meilleur des ressources inégalées du CNES et ainsi asseoir la crédibilité scientifique de mes travaux. Quand à MM. Cédric Baudoin et Fabrice Arnal, leur implication remarquable tout au long de la thèse m'a permis de structurer, faire avancer et positionner mon travail. L'intérêt qu'ils ont porté à ma recherche, nos nombreuses discussions et les moyens mis à ma disposition par Thales Alenia Space sont à l'origine d'avancées majeures et notamment de l'effort de standardisation.

Je remercie par ailleurs tous ceux sans qui cette thèse ne serait pas ce qu'elle est. Je pense ici en particulier à M.Patrice Raveneau, MM. Julien Fasson, Carlos Aguilar et Benoit Escrig ainsi que les autres membres de l'équipe IRT, et notamment la relève, Nesrine, Aziz, J-B et Guillaume.

Je remercie de plus M.Andrzej Duda pour avoir accepté de présider mon jury, ainsi que MM. Vania Conan et Konstantin Avratchenkov qui ont acceptés d'être les rapporteurs de cette thèse et M.Marcelo Dias de Amorim pour m'avoir fait l'honneur de participer au Jury. Ils ont également contribué par leurs remarques et suggestions à améliorer la qualité de ce mémoire, et je leur en suis très reconnaissant.

Pour son soutien au quotidien, sa confiance sans faille, et l'amour qu'elle me porte, je remercie celle qui est devenue ma femme au cours de cette thèse, Mme Ségolène Sallantin. Il semblerait qu'ensemble nous arrivions à bout des défis les plus fous.

Nos familles m'ont énormément donné et ont notamment beaucoup cru en moi. Je souhaite leur dire que la fierté que je pouvais lire dans leurs yeux tout au long de

la thèse, m'a porté dans cet ambitieux projet et protégé d'un monde extérieur qui continue de croire qu'une thèse est un échappatoire pour éternel étudiant.

Finalement, mes amis qui répondent au doux surnom de "gros" ont une part considérable dans la réussite de cette thèse. Certains m'ont devancé, d'autres m'ont inspiré, mais tous ont contribué à me rendre heureux et à m'apporter l'équilibre nécessaire.

En conclusion, je souhaite à tous les futurs doctorants de jouir d'un environnement professionnel et personnel aussi remarquable que le mien. Auquel cas, j'en suis sur, le succès ne pourra qu'être au bout.

Résumé

Le concept de réseau global est sans aucun doute la notion fondatrice de la prochaine génération de systèmes de communication. Dans un futur proche, un utilisateur devra être en mesure de conserver une excellente qualité de communication quel que soit l'endroit où il se trouve ou se déplace, sans jamais s'apercevoir des changements technologiques sous-jacents. En tant que marché de niche des communications, faire partie de cette dynamique est crucial pour les communications par satellite.

Ces dernières années, de nombreuses études ont néanmoins démontré que l'intégration du segment satellite dans un contexte global est fortement handicapée par ses caractéristiques intrinsèques. En effet, des protocoles essentiels tels que TCP qui influent fortement sur la performance moyenne pâtissent de la durée du RTT et sont inadaptés aux communications par satellite.

La communauté satellite a donc recours depuis les années 2000 à des solutions spécifiques sous la forme de TCP-PEPs (Performance Enhancement Proxy). Celles-ci offrent de très bonnes performances mais marginalisent le lien satellite en brisant le concept essentiel de communication de bout-en-bout, compliquant ainsi son intégration au sein d'un système multi-technologies.

Aussi, à défaut de trouver une solution conciliant intégrabilité et efficacité, le satellite risque-t-il de se trouver exclu de cet ambitieux projet, et ce malgré ses nombreux atouts.

Dans un premier temps, nous avons étudié les comportements des dernières évolutions de TCP initiées par les principaux systèmes d'exploitation. Ceux-ci nous ont permis de constater que des solutions bout-en-bout pertinentes, bien qu'initialement pensées pour optimiser les réseaux terrestres, peuvent offrir des performances similaires aux TCP-PEPs dans un environnement satellite. Toutefois, ces améliorations concernent essentiellement les connexions longues. Les faibles performances des solutions de bout-en-bout pour les connexions courtes, majoritaires dans l'Internet, continuent donc à justifier l'utilisation de TCP-PEPs.

Forts de ce constat, nous nous sommes intéressés plus spécifiquement à l'optimisation des protocoles de transport bout-en-bout pour les connexions courtes et avons proposé un mécanisme appelé Initial Spreading qui permet une amélioration significative des performances quel que soit le contexte.

Son concept simple vise à affranchir les débuts de connexion de leur dépendance au RTT en émettant, dès l'établissement de la connexion, une grande quantité de segments de données. L'attention est portée sur les conséquences d'un tel envoi dans un réseau congestionné, et alors que des solutions telles que la RFC 6928, proposée par Google, voient leurs performances fortement détériorées dans un tel environnement, notre mécanisme utilise un étalement précis et régulé des premiers segments envoyés qui lui permet d'atteindre des performances remarquables.

De nombreuses simulations avec NS2 ont tout d'abord validé l'intérêt et l'efficacité de notre mécanisme.
Un modèle mathématique des connexions courtes TCP a ensuite permis de corroborer ces résultats et de comprendre précisément les conséquences relatives à l'émission d'une rafale de segments (ou burst) sur la performance d'une connexion.
Finalement, nous avons implanté l'Initial Spreading dans le noyau Linux afin de le tester dans des réseaux terrestres et satellite et montrer le bien-fondé de notre proposition en environnement réel.

Nous avons alors pu affiner l'Initial Spreading au travers de nos évaluations et finalement présenter à l'IETF, sous la forme d'un "Internet Draft", un mécanisme à même d'améliorer de façon significative les performances des connexions TCP courtes indépendamment du contexte considéré et de l'état du réseau.

Abstract

Undoubtedly, the idea of global network is the founding concept of the next generation of communication systems. In the future, a user should therefore be able to maintain an excellent quality of communication regardless of where he is or moves without noticing the underlying technology changes. As a niche market of communications, be part of this dynamic is crucial for satellite communications.

However, in the recent years, many studies have shown that the integration of the satellite segment in a global context was complicated by the different characteristics of considered technologies. In fact, some of the most important protocols such as TCP that warranty the quality of a communication are strongly suffering from the RTT duration and so are inadequate for the satellite link.

Since the 2000s, satellite community therefore deployed specific solutions in the form of TCP-PEP. They offer very good performance, but marginalize the satellite link from the others by breaking the essential concept of end-to-end communication, and then make its integration among other technologies difficult.

Also, if a solution that enables a better integrability and efficiency is not found, the satellite may be excluded from this ambitious project, despite its numerous strengths.

We first conducted an extensive set of studies regarding the behavior of TCP latest flavors initiated by the main Operating Systems. They suggest that relevant end-to-end solutions, designed to fit terrestrial networks, can eventually offer similar performance in a satellite environment than the TCP-PEP solution. However, these optimisations only improve long-lived connections. The poor performance of short-lived connections, that are a majority in the Internet, continues therefore to justify the use of TCP-PEP.

Consequently, we focused on improving end-to-end transport protocols for short-lived connections and proposed a mechanism called Initial Spreading that allows significant performance improvements regardless of the context.

Its simple concept aims to overcome the RTT dependence that strongly damages the short-lived flows by emitting a large amount of data segments just after the connection establishment. We pay close attention to the consequences of releasing a large group of segments (burst) in a congested network. So while solutions such as RFC 6928, proposed by Google, see their performance sharply deteriorated in such

an environment, our mechanism ensures very good performance using an accurate and regulated spreading of the first sent segments.

Many simulations in NS2 first allowed us to validate the usefulness and scope of our mechanism.

A mathematical model of short-lived TCP connections then allowed us to corroborate these results and to understand in an accurate way the consequences of the transmission of bursts of segments on the average performance of a communication.

Finally, we implemented the Initial Spreading in the Linux kernel in order to test its behavior and efficiency in terrestrial and satellite networks and show the merits of our proposal in a real environment.

All these evaluations allowed us to refine our mechanism to significantly improve the performance of short-lived TCP connections regardless of the context in question and the state of the network. We finally submitted our proposal to the IETF in the form of an “Internet Draft”.

Table des matières

Résumé iii

Abstract v

Table des matières vii

Table des figures xi

Liste des tableaux xv

Liste des acronymes xvii

1 Introduction 1

2 Les protocoles de transport fiabilisés face aux réseaux à longue distance 7

2.1 Évaluation des protocoles Transmission Control Protocol (TCP)s dans le contexte satellite . . . 7

2.1.1 Description de TCP . . . 7

2.1.1.1 Établissement de connexion . . . 8

2.1.1.2 Le début de connexion . . . 9

2.1.1.3 La reprise sur erreur . . . 9

2.1.1.4 L'état stable . . . 11

2.1.1.5 New Reno : l'algorithme de Congestion Avoidance (CA) traditionnel 11

2.1.1.6 TCP Hybla : une solution pour les longs Round Trip Time (RTT)s 12

2.1.1.7 Compound TCP (CTCP) et TCP Cubic (Cubic) : les nouveaux TCPs . . . 13

2.1.1.8 Comparaison des différents TCPs . . . 14

2.2 Les TCP Performance Enhancement Proxy (TCP-PEP)s : une solution dédiée au satellite . . . 15

2.2.1 Le principe des TCP-PEPs . . . 15

2.2.2 Avantages et inconvénients des TCP-PEPs . . . 16

2.3 Les TCP-PEPs faces aux solutions bout-en-bout . . . 18

2.3.1 Le cas des connexions courtes . . . 19
2.3.2 Le cas des connexions longues . . . 19
2.4 Problématique . . . 19

3 Les pistes d'amélioration de TCP dans le cas des connexions courtes 21
3.1 Des solutions apportées à chaque état de TCP . . . 21
3.1.1 L'établissement de connexion . . . 21
3.1.2 La reprise sur erreur . . . 22
3.1.3 L'état de début de connexion . . . 23
3.2 Les bursts : acteurs majeurs des performances de TCP . . . 27
3.2.1 les bursts et TCP . . . 27
3.3 Le Pacing : un mécanisme de prévention des bursts . . . 28
3.3.1 Principe et objectifs du Pacing . . . 28
3.3.2 Inconvénients du Pacing . . . 28
3.4 Conclusion de l'état de l'art . . . 31

4 Initial Spreading : concept et premières simulations 33
4.1 Association du Pacing et d'une Initial Window (IW) accrue . . . 33
4.2 Présentation de l'Initial Spreading . . . 35
4.3 Validation Empirique par simulation . . . 37
4.3.1 Banc de tests utilisé . . . 37
4.3.2 Performance pour les flux courts . . . 38
4.3.3 Performance pour les flux longs . . . 42
4.3.4 Équité de l'Initial Spreading . . . 44
4.3.5 Choix de la taille de l'IW . . . 44
4.4 Conclusion de ces premières simulations . . . 46

5 Validation analytique de l'Initial Spreading 47
5.1 Introduction . . . 47
5.2 Étude des bursts . . . 47
5.2.1 Définition des bursts . . . 48
5.2.2 Étude empirique des conséquences des bursts sur un réseau congestionné . . . 49
5.2.3 Modélisation des bursts . . . 51
5.3 Modèle analytique pour les connexions courtes TCP . . . 52
5.3.1 Hypothèses . . . 52
5.3.2 Description du modèle . . . 54
5.3.3 Initialisation . . . 56
5.3.4 Généralisation : $\forall n \geq i \geq 2$ . . . 58
5.3.5 Généralisation : $\forall i \geq n$ . . . 62

5.4 Validation . . . 64
5.4.1 Évaluation du modèle . . . 65
5.4.2 Validation de l'Initial Spreading . . . 66
5.5 Conclusion . . . 67

6 Implantation de l'Initial Spreading 69
6.1 Implantation de l'Initial Spreading . . . 70
6.1.1 Implantation fondée sur l'utilisation de jiffies . . . 70
6.1.2 Limite de cette implantation . . . 72
6.2 Premiers enseignements des expérimentations en environnement réel . . . 73
6.2.1 Comportement du mécanisme dans le cas avec congestion . . . 73
6.2.2 Comportement du mécanisme proposé dans le cas sans congestion . . . 75
6.2.3 Discussion sur $T_{Spreading}$ . . . 76
6.2.4 Considérations autour de l'Initial Spreading . . . 77
6.2.5 Synchronisation entre connexions due aux jiffies . . . 79
6.3 Évolution de l'implantation et nouvel algorithme de l'Initial Spreading . . . 80
6.3.1 Implantation dans le noyau fondée sur FQ/Pacing . . . 80
6.3.2 Algorithme final de l'Initial Spreading . . . 82

7 Évaluation de l'Initial Spreading en environnement réel 85
7.1 Validation sur réseau filaire . . . 85
7.1.1 Résultats expérimentaux . . . 85
7.1.2 Initial Spreading : un outil de lutte contre le bufferbloat . . . 88
7.2 Validation sur réseau satellite . . . 89
7.2.1 Impact de la couche MAC sur l'Initial Spreading . . . 90
7.2.2 Configuration de l'Initial Spreading . . . 94
7.2.3 Conclusion . . . 95
7.3 L'Initial Spreading face aux TCP-PEPs . . . 96
7.3.1 Comparaison dans des environnements non-congestionnés . . . 96
7.3.2 Comparaison dans des environnements congestionnés . . . 97
7.4 Quel $T_{Spreading}$ envisager? . . . 98

8 Conclusion et Perspectives 101
8.1 Conclusion . . . 101
8.1.1 Poids des bursts sur la performance moyenne . . . 102
8.1.2 Initial Spreading, une réponse aux faiblesses de TCP . . . 102
8.2 Perspectives . . . 103
8.2.1 Perspectives à court terme . . . 103
8.2.2 Perspectives à plus long terme . . . 104

Liste des communications 105

Conférences internationales avec comité de lecture **105**

Contributions aux instances de standardisation **105**

Articles Soumis **105**

Bibliographie **107**

Table des figures

1.1 Comparaison des performances des TCPs de bout-en-bout et des TCP-PEPs . . 2

2.1 Principes généraux de TCP sous forme de diagramme d'états simplifié 8
2.2 Établissement de connexion entre 2 utilisateurs 9
2.3 Topologie simple utilisée pour l'évaluation des différents algorithmes de contrôle de congestion 11
2.4 Évolution de TCP New Reno 12
2.5 Évolution de Cubic 14
2.6 Les deux différentes architectures des TCP-PEPs 15
2.7 Premiers échanges de segments illustrant l'utilisation d'un seul TCP-PEP 16
2.8 Comparaison des solutions TCP-PEPs, de TCP New Reno et de Cubic 18

3.1 Quick Start : Demande du débit disponible 24
3.2 Topologie "Dumbbell" utilisée dans nos simulations 26
3.3 Comparaison des durées de transmission des flux en fonction de l'IW 26
3.4 Durée normalisée de la transmission de différentes tailles de flux avec et sans Pacing. Figure tirée de [1] 29
3.5 Durée de la transmission de flux de 300 segments en fonction du pourcentage de flux utilisant le Pacing. Figure tirée de [1] 30
3.6 Récapitulatif des différents protocoles ou mécanismes de la couche Transport considérés 32

4.1 Topologie simple utilisée pour l'évaluation de l'impact des différents mécanismes sur l'évolution de la file d'attente 33
4.2 Nombre de segments ajoutés dans le buffer par un flux en Slow Start 34
4.3 Illustration des différents temps utilisés dans le manuscrit 35
4.4 Chronogramme représentant la transmission de 12 segments avec une IW de 4 segments en utilisant différents mécanismes 36
4.5 Évolution des files d'attente pour une IW de 10 segments, avec et sans Pacing et avec Initial Spreading 37

4.6 La topologie Dumbbell retenue dans la plupart des simulations et expériences . . 38
4.7 Chronogramme comparant la transmission de 10 segments sans congestion pour différents mécanismes 39
4.8 Durée de délivrance moyenne pour une IW de 10 segments avec et sans Initial Spreading 40
4.9 Conséquences de l'augmentation de l'IW sans Initial Spreading 41
4.10 Comparaison de la durée moyenne de délivrance pour différentes IW avec et sans Initial Spreading 42
4.11 Conséquences des effets de la synchronisation sur les flux longs avec et sans Initial Spreading 43
4.12 constat de l'équité de l'Initial Spreading 44

5.1 Probabilité de transmettre l'intégralité de l'IW en fonction de sa taille avec et sans burst 49
5.2 Probabilité d'avoir X des segments du burst qui font suite au premier segment perdu correctement transmis ($X \in 1, 2, 3$) en fonction de la place de la perte . . . 50
5.3 Cas où 2 des 7 segments "suivants" sont correctement transmis 51
5.4 Illustration des différents bursts considérés par le modèle 54
5.5 Correspondance entre le modèle et un scénario de transmission réelle des segments 55
5.6 Un des scénarios possibles pour transmettre D_4^4 55
5.7 $\bar{T}\left(D_2^2\right)$, les différents scénarios possibles 57
5.8 Diagramme d'états pour D_2^2 58
5.9 $\bar{T}\left(S_2^2\right)$, les différents scénarios possibles 59
5.10 $\bar{T}\left(S_5^5\right)$, perte du cinquième segment 60
5.11 $\bar{T}\left(S_5^5\right)$, perte de 2 segments : déclenchement du Fast Retransmit après réception de 3 Duplicated Acknowledgment (DupACK)s 61
5.12 $\bar{T}\left(B_4^3\right)$, avec perte des segments 2 et 4 63
5.13 Comparaison entre le modèle et les simulations pour la transmission de flux courts avec IW = 1 dans différents environnements 65
5.14 Comparaison entre le modèle et les simulations pour la transmission de flux courts avec IW = 1 et IW = 10 avec et sans Initial Spreading 67

6.1 Fonctionnement de la pile TCP dans le noyau Linux 72
6.2 Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 42ms 74
6.3 Conséquences d'une mesure erronée du RTT durant l'établissement de la connexion sur la transmission de 20 segments 74
6.4 Conséquences de l'utilisation de jiffy dans le choix de T_{max} 78
6.5 Différence entre simulation et implantation réelle dans le cas de connexions en parallèle 80

7.1 Topologie du réseau retenue pour nos expérimentations 86
7.2 Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 42ms 86
7.3 Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 252ms 87
7.4 Conséquences de la taille du buffer sur les performances avec et sans Initial Spreading 89
7.5 Topologie du réseau employée pour nos expérimentations avec satellite 90
7.6 Encapsulations successives des paquets IP 91
7.7 Différents ordonnancements niveau MAC 92
7.8 Évolution de la file d'attente du routeur dans les cas d'un réseau filaire et satellite 93
7.9 Comparaison des temps moyens de délivrance pour des connexions courtes avec et sans Initial Spreading dans un réseau satellite ayant un délai de 280ms 95
7.10 Comparaison des 3 solutions existantes (TCP-PEP, RFC 3390, RFC 6928) avec l'Initial Spreading dans un réseau non congestionné 97
7.11 Comparaison des 3 solutions existantes (TCP-PEP, RFC 3390, RFC 6928) avec l'Initial Spreading dans un réseau congestionné 98

Liste des tableaux

3.1 Nombre de RTTs nécessaires à la transmission d'une connexion courte 24

4.1 Gains atteints par l'utilisation d'Initial Spreading pour l'émission de 10 segments 45

5.1 Tableau récapitulatif des états et paramètres utilisés dans notre modèle 56

6.1 Conséquences du choix de T_{max} pour un goulot d'étranglement avec un débit réel de 6 Mb/s 78

6.2 Conséquences des jiffies sur le T_{max} avec $HZ = 300Hz$ 79

Liste des acronymes

ACK Accusé de Réception

AWND Advertised Window

BBFrame Base Band Frame

CA Congestion Avoidance

CTCP Compound TCP

Cubic TCP Cubic

CWND Congestion Window

Del Ack accusés de réception retardés

DupACK Duplicated Acknowledgment

DVB Digital Video Broadcasting

DVB-S Digital Video Broadcasting - Satellite

GSE Generic Stream Encapsulation

HTTP HyperText Transfer Protocol

IP Internet Protocol

IPsec Internet Protocol Security

IETF Internet Engineering Task Force

IW Initial Window

MSS Maximum Segment Size

NIC Network Interface Card

NS2 Network Simulation 2

OS Systèmes d'exploitation

OSEG Outstanding Segment

QdS Qualité de Service

RTO Retransmission Time Out

RTT Round Trip Time

RWND Receiver Advertised Window

SACK Selective Acknowledgment

ssthresh Slow Start Threshold

TCP Transmission Control Protocol

TCP-PEP TCP Performance Enhancement Proxy

UDP User Datagram protocol

XCP eXplicit Control Protocol

1 Introduction

Le concept de réseau global est sans aucun doute la notion fondatrice de la prochaine génération de systèmes de communication. Dans un futur proche, un utilisateur devra être en mesure de conserver une excellente qualité de communication quel que soit l'endroit où il se trouve ou se déplace, sans jamais s'apercevoir des changements technologiques sous-jacents (WiFi, réseaux mobiles, réseaux satellite,...). En tant que marché de niche des communications, faire partie intégrante de cette dynamique est cruciale pour les communications par satellite.

Ces dernières années, de nombreuses études ont néanmoins démontré que les caractéristiques intrinsèques des différentes technologies compliquent l'intégration du segment satellite dans un contexte global. Nombre de protocoles régissant la performance de la communication perdent en effet en efficacité dès lors qu'un segment satellite est emprunté.

Aussi, malgré ses nombreux atouts et notamment sa propension à couvrir à moindre coût des zones étendues et inaccessibles, le satellite risque-t-il de se trouver exclu de cet ambitieux objectif et contraint à jouer les seconds rôles.

Les raisons des singularités protocolaires des communications par satellite

Un des exemples flagrants de cette singularité concerne plus spécifiquement les protocoles de bout en bout. On pense alors particulièrement aux protocoles de transport.

En effet, les contraintes des communications par satellite et notamment le très long délai rendent inadéquats les protocoles de transport classiques tels que TCP New Reno. Ces derniers, conçus pour assurer un service en mode connecté efficace pour les réseaux filaires à faible temps de propagation, subissent de plein fouet l'augmentation du Round Trip Time (RTT), c'est-à-dire du temps compris entre l'émission d'une donnée et la réception de son Accusé de Réception (ACK). Ce RTT supérieur à 0.5 seconde, plus de 10 fois supérieur aux RTTs terrestres classiques, affecte lourdement un grand nombre de leurs mécanismes tels que l'établissement de connexion, le contrôle de flux ou encore la reprise sur erreur, et dégrade de façon significative leur efficacité.

La communauté "satellite" a donc eu recours dans les années 2000 à des solutions dédiées. Ces solutions appelées TCP Performance Enhancement proxy (TCP-PEP) permettent d'atteindre un bon niveau de performance en scindant notamment la connexion TCP de bout-en-bout afin

d'isoler le segment satellite et de cacher ses spécificités à l'utilisateur. Les TCP-PEPs sont donc devenus rapidement la solution de référence et sont très largement utilisés par l'ensemble des opérateurs de télécommunication par satellite.

Pourtant, les TCP-PEPs et la rupture du principe de bout-en-bout qu'ils induisent sont à l'origine de la difficulté à utiliser conjointement les communications par satellite et les autres types de communications.

Quelle solution envisager?

Afin de permettre l'intégration du satellite dans la mouvance actuelle des systèmes de communication, toute nouvelle proposition pour la couche transport visant à en améliorer les performances sur le segment satellite doit être, dans le pire des cas, transparente pour le reste du réseau.

Deux axes principaux de recherche se présentent alors :

- Poursuivre la piste des solutions spécifiques "satellites" de type TCP-PEPs en s'appliquant à ce que leur utilisation ne perturbe pas le fonctionnement et l'efficacité des protocoles de bout-en-bout actuels.
- Viser une optimisation des performances des TCPs "terrestres" classiques, en s'appuyant sur l'étude des évolutions récentes apportées à certaines de leurs fonctions qui permettent, comme nous le montrerons par la suite, d'approcher par moment les performances des TCP-PEPs dans un contexte satellite.

Le point de départ de cette thèse a donc consisté à comparer les solutions "satellite" avec les solutions "terrestres", observant notamment les récentes et pertinentes évolutions de TCPs que sont TCP-Compound et TCP-Cubic, afin de juger de l'écart réel de performance [2].

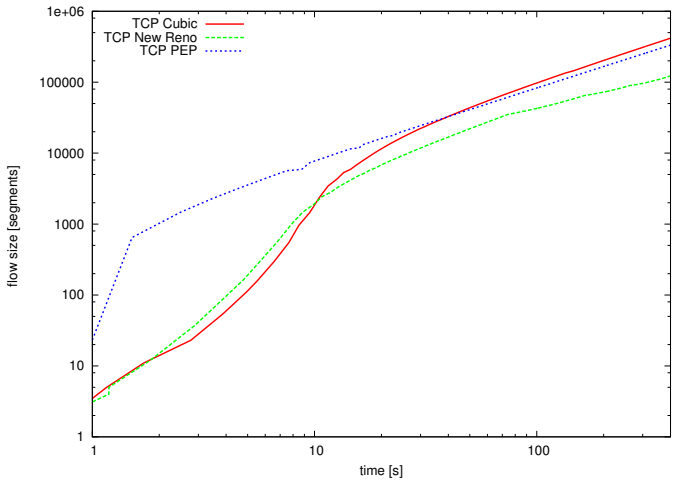

FIGURE 1.1 – Comparaison des performances des TCPs de bout-en-bout et des TCP-PEPs

Nous avons alors montré que ces évolutions de TCP permettaient de se passer de TCP-PEPs pour la transmission de certaines catégories de connexions, mais n'apportaient pas encore une réponse complète. Nous détaillerons ces résultats plus loin dans ce manuscrit mais à titre d'illustration, la Figure 1.1 compare le temps nécessaire à la transmission d'une certaine quantité de données, exprimée en nombre de segments, dans le cas d'une utilisation de TCP-PEPs, de TCP New Reno, et de TCP Cubic.

Nous pouvons directement observer que :

- dans le cas de connexions longues, les nouveaux TCPs atteignent des résultats similaires à ceux des solutions TCP-PEPs et surclassent les anciens TCPs ;
- dans le cas des connexions courtes, aucun progrès significatif ne semble avoir été fait, et l'écart de performance entre les solutions TCP-PEPs et les solutions terrestres est inchangé et demeure significatif.

Les enseignements sont nombreux et nous offrent une réelle base de travail. En particulier, bien que cela ne soit pour l'instant vrai que dans le cas des connexions longues, les évolutions protocolaires non dédiées aux satellites semblent être en mesure de rivaliser avec les solutions spécifiques de type TCP-PEPs. Une généralisation de cette tendance permettrait à terme de se passer de ces solutions spécifiques et donc de ne plus différencier d'un point de vue transport le segment satellite des autres segments.

Cette possibilité aux retombées particulièrement intéressantes pour l'intégration des satellites dans le projet de nouveaux systèmes de communication ne sera néanmoins possible qu'à travers une optimisation des performances des TCPs classiques pour les connexions courtes.

Une faiblesse plus générale : les débuts de connexion

Nous avons alors procédé à un état de l'art des solutions visant à améliorer les débuts de connexion. Ce dernier a révélé que les mauvaises performances des TCPs classiques pour les connexions courtes n'étaient pas l'apanage des communications par satellite. Contrairement au traitement des connexions longues, le début de connexion a peu évolué, et se trouve être désormais une faiblesse majeure des solutions classiques.

Pour autant, les travaux sur l'amélioration des débuts de connexion ont été nombreux, mais aucun des mécanismes proposés n'a réussi à obtenir l'assentiment général, étant tous reconnus ou pressentis comme coupables de la dégradation des performances du réseau dans certains cas de figure et en particulier dans les réseaux très chargés.

Une des idées en vogue, défendue par Google, et déjà implantée dans les systèmes d'exploitation récents illustre parfaitement ce cas de figure. Elle consiste à augmenter la taille de la fenêtre initiale de congestion (IW), et donc à émettre dès l'établissement de la connexion l'équivalent d'une dizaine de segments sous la forme d'une unique rafale (ou burst). Cette solution agressive permet notamment une réduction considérable du délai moyen nécessaire à la transmission de connexions courtes lorsque le réseau n'est pas congestionné, mais voit ses performances fortement diminuées dans le cas contraire : tous les segments du burst ayant de plus grands risques d'être

perdus.

Initial Spreading : une solution de bout-en-bout

Forts des constats précédents, nous avons élargi notre objectif et proposé une solution qui améliore de façon générale les performances des connexions courtes, indépendamment du support. Nous verrons que cela permet de combler les lacunes des solutions TCPs dans le contexte satellite et donc de s'affranchir des TCP-PEPs.

Notre proposition, intitulée Initial Spreading, vise a émettre dès le premier RTT la totalité de la connexion courte en prenant garde de minimiser son impact sur le réseau traversé. Pour ce faire, elle exploite 2 mécanismes *a priori* antagonistes dans leurs traitements des bursts : l'augmentation de la taille de la fenêtre initiale et le mécanisme d'espacement ou Pacing, dont l'objectif est de supprimer autant que possible les bursts.

Pour valider notre proposition, nous avons tout d'abord effectué un nombre important de simulations qui a confirmé l'intérêt de l'Initial Spreading quel que soit le scénario envisagé [3], montrant des bénéfices significatifs face à l'ensemble des solutions existantes.

Puis, conscients des lacunes des outils utilisés pour simuler le comportement de TCP dans des milieux congestionnés, nous avons proposé un modèle mathématique fondé sur l'existence d'un lien bas débit engorgé qui correspond bien à notre environnement satellitaire mais aussi à la plupart des situations rencontrées dans l'Internet. Fondé sur une appréhension fine des bursts, une modélisation à l'aide d'un automate probabiliste temporisé nous permet d'obtenir le délai moyen en fonction de la taille du flux, du RTT et du pourcentage de pertes, validant ainsi nos simulations [4] mais également nos hypothèses sur les conséquences des émissions groupées.

Finalement, nous avons implanté l'Initial Spreading dans le noyau linux. Cette phase était de notre point de vue indispensable pour montrer le bien-fondé et la faisabilité de notre proposition en environnement réel. L'implantation a pu ensuite être testée dans des environnements simples en suivant les exemples pris pour la simulation et le modèle mathématique, dans le cadre de l'émulateur OpenSand et sur de vrais satellites, ainsi que sur des réseaux filaires traditionnels. Alors que les expérimentations sur satellite (réel ou émulé) nous ont permis d'observer la difficulté de mise en œuvre de l'Initial Spreading ainsi que l'impact de la couche accès Digital Video Broadcasting (DVB) sur notre mécanisme, les expérimentations sur des réseaux terrestres ont, quant à elle, conforté nos résultats précédents et validé tout l'intérêt d'utiliser l'Initial Spreading.

Au regard des résultats obtenus, nous avons proposé notre travail à l'IETF [5] afin de statuer sur son originalité, son attractivité et la faisabilité de son implantation auprès des différents acteurs du monde du Transport. Nos nombreuses discussions avec les membres de l'IETF, outre avoir confirmé l'intérêt et les attentes que pouvait susciter l'Initial Spreading, nous ont également aidé à optimiser son implantation en utilisant les dernières évolutions logicielles du noyau Linux et notamment en adaptant le dernier projet de Google visant à développer les capacités "réseaux" du noyau.

Cela nous a alors permis d'affiner l'algorithme de notre mécanisme en optimisant la relation

entre les besoins théoriques et les contraintes réelles auxquelles il devra faire face. Notre nouvelle version de l'Initial Spreading résout notamment le problème des interactions avec les couches basses du satellite. Nous avons ainsi pu établir que l'utilisation d'Initial Spreading en complément des nouvelles versions de TCP offre une telle amélioration des performances dans le cas des connexions courtes, et ce quel que soit le support, qu'elle permet notamment :

- de se passer des TCP-PEPs ;
- d'améliorer significativement les performances du réseau de façon générale.

2 Les protocoles de transport fiabilisés face aux réseaux à longue distance

Depuis plus de 20 ans, les communications par satellite souffrent, dans le contexte d'une utilisation Internet, de l'inadéquation des protocoles utilisés. Cela est d'autant plus remarquable pour les protocoles de bout-en-bout de la couche transport qui pâtissent fortement de la durée du RTT [6].

TCP est le protocole de transport le plus largement employé. Pensé et conçu pour les réseaux filaires, ses performances se trouvent fortement amoindries dans un tel environnement.

Dans les prochaines sections, nous détaillerons les différents mécanismes de TCP susceptibles d'être dégradés par le contexte satellite et évaluerons notamment leurs coûts en termes de RTTs.

2.1 Évaluation des protocoles TCPs dans le contexte satellite

2.1.1 Description de TCP

TCP est un protocole de transport de bout-en-bout fonctionnant en mode connecté dont l'objectif est de transmettre de manière fiable une information entre deux utilisateurs. Il utilise le principe des fenêtres glissantes pour contrôler la performance de bout-en-bout. Le débit d'une connexion est donc directement lié à la taille de sa fenêtre, cette dernière représentant la quantité d'information qu'il peut émettre sans avoir à attendre de nouvel accusé de réception (ACK). La fenêtre est appelée Initial Window (IW) lorsqu'elle est égale au nombre de segments que l'émetteur peut envoyer dès l'établissement de la connexion, et Congestion Window (CWND) dès lors que les accusés de réception ont fait évoluer l'IW. Par ailleurs, à tout moment, la taille de la fenêtre de l'émetteur est égale au minimum de CWND et de Advertised Window (AWND), cette dernière étant la capacité de la fenêtre de réception (Receiver Advertised Window (RWND)) transmise par le récepteur.

N'ayant pas ou peu d'informations sur la teneur du réseau emprunté, notamment en termes de type de liens, débit disponible ou congestion éventuelle, TCP ne peut compter que sur les

événements qu'ils interprètent comme des notifications de congestion pour établir le débit de la connexion et assurer aux utilisateurs le meilleur débit instantané possible.

Le protocole TCP est spécifié par une machine à états ; le passage de l'un à l'autre est le résultat du traitement des événements de congestion via le processus de reprise sur erreur.

Entre l'établissement et la fermeture de la connexion, le mode opératoire de TCP est séparé en deux états principaux permettant d'assurer le contrôle de flux et de congestion :

– le début de connexion ;
– l'état stable.

La Figure 2.1 est une représentation simplifiée du diagramme d'états de TCP insistant sur les changements d'états qui ont lieu entre l'ouverture et la fermeture d'une connexion

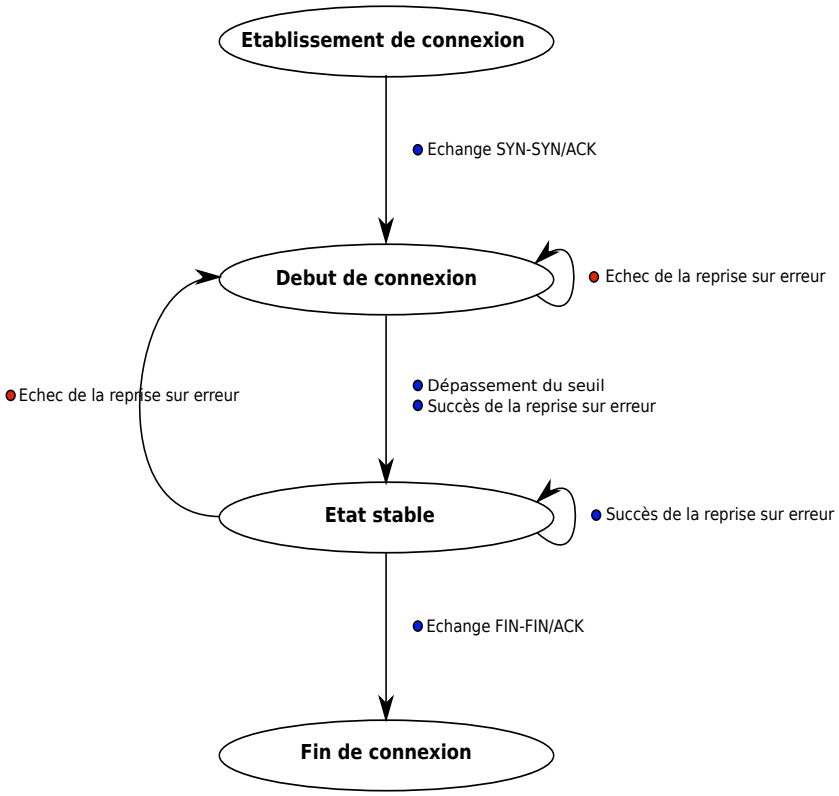

FIGURE 2.1 – Principes généraux de TCP sous forme de diagramme d'états simplifié

2.1.1.1 Établissement de connexion

L'établissement de connexion est à l'initiative de l'émetteur. Celui-ci initie un échange de segments, sollicitant dans un premier temps l'ouverture de la connexion en envoyant un segment SYN. Le récepteur confirmera la prise en compte de la volonté de l'émetteur en accusant réception du segment reçu et en demandant à son tour l'ouverture de la connexion via l'émission d'un

segment SYN/ACK. L'émetteur pourra alors entériner l'ouverture de connexion en validant le segment reçu, mettant ainsi un terme à l'échange dit de poignée de main qui aura alors duré 1 RTT pour le client. La Figure 2.2 illustre l'établissement de la connexion.

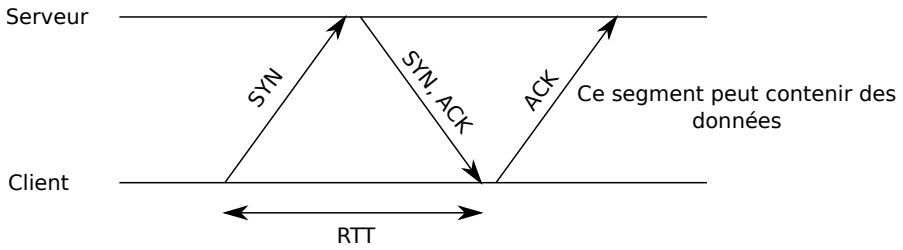

FIGURE 2.2 – Établissement de connexion entre 2 utilisateurs

2.1.1.2 Le début de connexion

La connexion est établie et le RTT connu, pour autant aucun autre indicateur de l'état du réseau n'a été échangé. TCP utilise cet état et notamment le mécanisme du "Slow-Start" pour sonder le réseau : il commence par émettre une petite quantité de données puis augmente rapidement sa capacité d'émission jusqu'à la détection d'un événement de congestion ou au dépassement d'une variable appelée Slow Start Threshold (ssthresh), qui détermine si la connexion doit continuer en Slow Start ou passer à l'état stable.

Durant le Slow Start, TCP commencera par émettre le contenu de l'IW, dont la taille est communément fixée à 3 segments par la RFC 3390 [7], avant d'attendre les accusés de réception des segments envoyés. Chaque ACK permettra alors à l'émetteur d'augmenter d'une unité la taille de sa fenêtre de congestion, c'est-à-dire, du nombre potentiel de segments qu'il peut envoyer sans attendre de nouvel accusé. L'émetteur doublera ainsi sa capacité d'émission à chaque RTT.

Le Slow Start permet ainsi d'éviter une surcharge du réseau qui serait préjudiciable non seulement à la performance individuelle de la connexion mais également à la performance collective du réseau. Cependant, il n'en reste pas moins trop conservateur dans le cas des transmissions de connexions courtes et inadapté aux réseaux à long RTT de par sa dépendance directe à ce paramètre. Par exemple, la transmission d'une connexion de 10 segments, représentative des objets HTTP, nécessitera au minimum 3 RTTs et donc plus d'1.5 seconde dans un environnement satellitaire.

2.1.1.3 La reprise sur erreur

L'expiration d'une temporisation sur la non-réception d'un ACK a longtemps été le seul moyen de détecter une perte. Cette temporisation appelée Retransmission Time Out (RTO) est enclenchée à l'établissement de la connexion et remise à zéro chaque fois que la transmission d'un segment est correctement validée. Sa durée importante, fixée à 3 secondes par la RFC 2988 [8], permet d'être certain que le segment attendu est bien perdu mais également de s'assurer que

le réseau a eu le temps nécessaire pour sortir d'un état potentiel de congestion. L'expiration du Timer entraîne alors irrémédiablement le retour dans l'état précédent avec une IW égale à 1.

Avec ce seul mécanisme de détection, aucune distinction n'est par exemple possible entre les pertes dues à des erreurs sur le support et les pertes dues à une congestion majeure du réseau. Toute perte résulte en une même limitation du débit de la connexion.

Depuis New Reno, la reprise sur erreur de TCP a été affinée en ajoutant des mécanismes qui lui permettent de distinguer les raisons d'une perte afin de la traiter plus efficacement.

Ces mécanismes reposent sur un des principes de TCP qui veut que le récepteur envoie à chaque réception de segment l'ACK du dernier segment reçu de façon ordonnée. Ainsi, la réception d'un segment ne faisant pas numériquement suite au précédent entraîne l'émission d'un DupACK, c'est-à-dire d'un ACK en tout point équivalent au dernier envoyé.

Fast Retransmit

Fast Retransmit utilise la réception de ces DupACKs pour statuer sur la raison de la perte. La réception de 3 DupACKs, nombre jugé suffisant pour considérer que les segments n'arriveront pas de façon désordonnée et que la perte n'était pas due à une congestion majeure du réseau mais à un événement isolé, entraîne la retransmission instantanée du prochain segment attendu par le récepteur et la transition vers l'état stable.

Fast Recovery

Ce passage à l'état stable se fait via une réduction significative de CWND afin de prévenir une potentielle surcharge du réseau. Fast Recovery [9, 10] est alors utilisé pour diminuer l'effet de cette réduction de la fenêtre : chacun des DupACKs reçu incrémentera la taille de CWND d'une unité et ce jusqu'à la réception de l'accusé du segment retransmis.

Ainsi lorsqu'un segment est perdu, TCP réagit de 2 façons possibles :

- soit il est en mesure de récupérer la perte en utilisant Fast Retransmit, auquel cas il entrera dans l'état stable ;
- soit il ne peut pas, et dans ce cas, il recommence en Slow Start après expiration du RTO.

Selective Acknowledgment (SACK)

Par ailleurs, il n'est pas rare que TCP subisse des pertes multiples dans une même fenêtre, ce qui a pour conséquence de détériorer de façon importante sa performance. En effet, sans mécanisme particulier, la réception des DupACKs permet à TCP de ne récupérer qu'une perte par RTT.

Le mécanisme SACK [11] associé à une politique de retransmission sélective aide à la récupération de pertes multiples. Le récepteur TCP ne se contente plus de renvoyer un DupACK accusant réception du dernier segment arrivé dans l'ordre, mais envoie également à l'émetteur les segments qu'il a correctement reçus dans un champ SACK. L'émetteur peut alors retransmettre les données

manquantes dès réception des trois DupACKs signalant le ou les mêmes segments manquants.

Dans tous les cas, le temps minimal d'attente pour récupérer une perte est d'1 RTT, et peut atteindre plusieurs secondes si les mécanismes de reprise sur erreur ne sont pas applicables. Il est par ailleurs intéressant de noter que la valeur du RTO n'est pas fonction du RTT et sera proportionnellement plus pénalisante pour les connexions à RTT court que pour celles à RTT long.

2.1.1.4 L'état stable

L'objectif de cet état est de permettre à la connexion d'évoluer à un débit élevé, sachant que des événements de congestion ont été détéctés [12, 13]. TCP entre dans cet état suite au dépassement du ssthresh ou au succès du mécanisme de Fast Retransmit.

Cet état géré par des algorithmes d'évitement de congestion (CA) sujets à de nombreuses évolutions [14, 15] est essentiel dans les performances de TCP. Ainsi, par abus de langage, les différentes versions de TCPs portent généralement le nom de leur algorithme de CA. Récemment, les Systèmes d'Exploitation tels que Windows et Linux ont proposé et adopté leur propre version appelée respectivement CTCP [16] et Cubic [17], celles-ci viennent concurrencer TCP New Reno, la version historique, mais également des solutions plus spécifiques telles que TCP Hybla [18] qui ciblent les réseaux à long RTTs et notamment les communications par satellite.

La Figure 2.3 présente le banc de test utilisé pour observer et comparer les différents CA. Le lien satellite est émulé par Opensand [19], une plateforme d'émulation des systèmes de télécommunications par satellite. Nous avons utilisé le logiciel *iperf* entre les 2 machines Linux d'extrémité pour engendrer le trafic.

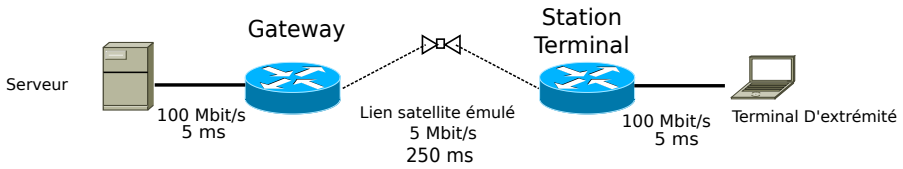

FIGURE 2.3 – Topologie simple utilisée pour l'évaluation des différents algorithmes de contrôle de congestion

2.1.1.5 New Reno : l'algorithme de CA traditionnel

L'algorithme de CA de New Reno [20] est très simple, et bien que son utilisation s'amenuise, il reste néanmoins déployé dans un grand nombre de machines.

Suite au Fast Retransmit, TCP réduit sa CWND et donc son débit de moitié, et va accroître lentement ce dernier en incrémentant d'une unité la taille de sa fenêtre à chaque RTT, et ce, jusqu'à la prochaine détection d'un événement de congestion.

TCP New Reno est très conservateur : il aura en effet besoin de $\frac{CWND}{2}$ RTTs pour récupérer le débit avant réduction ($CWND$ représentant la taille de fenêtre atteinte alors). Ses performances seront donc totalement dépendantes du RTT avec une accélération du débit inversement proportionnelle à la durée de ce dernier.

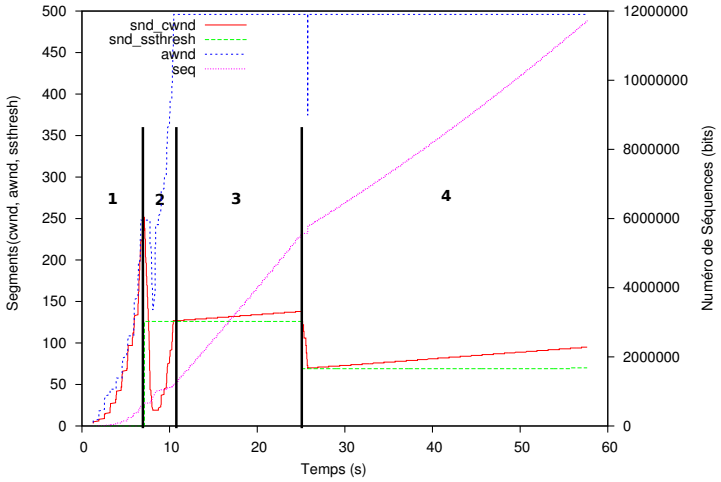

FIGURE 2.4 – Évolution de TCP New Reno

La Figure 2.4 montre l'évolution d'une connexion utilisant New Reno sur un lien satellite dans un environnement sans aucune congestion. Cette courbe traçant les résultats obtenus pour une seule connexion est représentative de TCP New Reno et illustre parfaitement les différentes phases de la connexion :

- Phase 1 : Suite à l'établissement de la connexion, le Slow Start débute et la CWND est rapidement augmentée jusqu'à la première perte.
- Phase 2 : La première perte n'ayant pu être récupérée, la connexion repart en Slow Start quelques secondes plus tard avec une IW unitaire, après avoir au préalable fixé la taille de ssthresh à la moitié de la fenêtre atteinte en Phase 1.
- Phase 3 : Dès lors que la taille de le CWND atteint la valeur de ssthresh, TCP entre dans l'état stable et utilise l'algorithme de contrôle de congestion de New Reno pour augmenter la taille de sa fenêtre.
- Phase 4: Une nouvelle perte a lieu, mais cette dernière peut être récupérée par l'algorithme de Fast Retransmit. Ainsi, TCP se contente de réduire sa CWND avant de repartir en état stable.

2.1.1.6 TCP Hybla : une solution pour les longs RTTs

En réaction aux faiblesses des versions de TCPs telles que Reno et New Reno lorsqu'elles se trouvent confrontées à un long RTT, TCP Hybla [18] a été proposé pour rendre TCP indépendant au RTT durant la phase de contrôle de congestion et gommer ainsi les différences de performances

de TCPs qui lui sont dues. Son objectif est en effet de permettre aux entités d'extrémité le même débit d'émission instantané que celui d'un lien de référence ayant quelques millisecondes de RTT, et ce quelque soit le RTT réel de la connexion.

Pour ce faire, il utilise un ensemble de procédures qui inclut notamment une augmentation plus rapide de la taille de la CWND visant à compenser la réduction du débit provoquée par la longueur du RTT et un espacement des segments pour atténuer les conséquences sur le réseau de l'émission d'un nombre de segments plus important.

L'utilisation d'Hybla a montré de réelles améliorations du débit pour les communications par satellite[21].

Néanmoins nous noterons deux défauts qui ont réduit son intérêt pour la communauté "réseau" :

- Le RTT de référence est considéré égal à un RTT terrestre classique. Dès lors que le RTT mesuré est inférieur ou égal à ce RTT de référence, Hybla n'a aucune conséquence. Ainsi, Hybla est au mieux transparent pour les réseaux terrestres et ne permet donc pas d'améliorations motivant son implantation dans les nouveaux noyaux.
- Les auteurs suggèrent l'espacement pour contrecarrer les effets liés à l'émission d'une grande CWND dans le réseau. Nous verrons dans la partie 3.4 que cette solution fait débat et peut avoir de sérieuses contreparties.

2.1.1.7 CTCP et Cubic : les nouveaux TCPs

Bien qu'elles soient moins affectées que les communications par satellite, les communications terrestres subissent également la faiblesse des performances de New Reno. Ainsi, conscients des conséquences en termes de performance du caractère trop conservateur de New Reno, les nouveaux algorithmes de CA tels que Cubic [17] et Compound [16, 22] vont tâcher d'approcher le plus rapidement possible le débit maximal avant congestion, en considérant la taille de la fenêtre atteinte lors de la détection de la perte comme un point pivot essentiel.

Ainsi les deux mécanismes coupent la progression linéaire de New Reno pour proposer une évolution en trois étapes :

- Étape 1 : la CWND est tout d'abord réduite à 80% de la valeur précédente afin de prévenir une éventuelle congestion, puis les deux algorithmes augmentent rapidement la taille de leur fenêtre de façon à sonder de nouveau le réseau. Cette phase a pour but de ne plus souffrir de pertes isolées et permet de retrouver rapidement un débit élevé dans un tel cas de figure.
- Étape 2 : durant cette phase, les deux mécanismes font l'hypothèse que la conservation de leur débit actuel leur permet un partage efficace et équitable de la bande passante avec les autres utilisateurs. Le passage de la première phase à la deuxième est différent pour les deux mécanismes. Ainsi, Cubic utilise un seuil statique égal à la valeur atteinte par la CWND au moment de la détection de la perte, tandis que CTCP utilise un indicateur dynamique. CTCP observe l'évolution de la mesure du RTT. Jugeant qu'une augmentation

significative de cette durée est révélatrice d'un encombrement du réseau et donc à terme d'une congestion, il décide alors de stopper la phase 1 pour passer à la phase 2.

– Étape 3 : après un certain temps sans détection de congestion, les deux mécanismes vont tester si l'évolution constante et dynamique du réseau leur est favorable. Ils sondent donc à nouveau le débit disponible en augmentant leur CWND. Cubic le fait de façon très agressive tandis que CTCP utilise la croissance linéaire de TCP New Reno.

Parmi les nombreux avantages de ces nouveaux algorithmes, leur indépendance vis-à-vis du RTT est à souligner et laisse présager des résultats très intéressants pour les communications par satellite. Pour y parvenir, CTCP fonde l'évolution de sa CWND à la fois sur la détection des pertes ("loss-based"), mais également sur l'évolution de la mesure du RTT ("delay-based"). Ce dernier point lui permet de faire évoluer sa fenêtre indépendamment du RTT. Cubic, tout comme New Reno, est uniquement fondé sur la détection des pertes, mais, à la différence de ce dernier, il mesure l'évolution du temps entre les pertes pour modifier sa CWND et s'affranchir ainsi de la dépendance vis-à-vis du RTT.

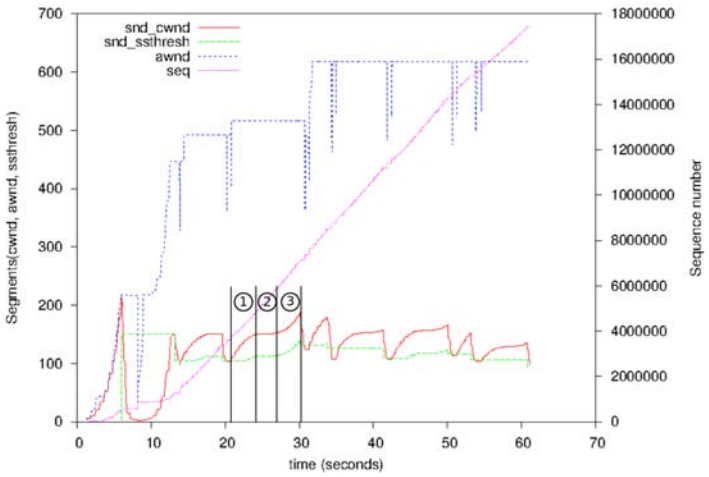

FIGURE 2.5 – Évolution de Cubic

Obtenue de la même façon que la Figure 2.4, la Figure 2.5 illustre les 3 phases de l'évolution de l'algorithme de congestion d'une connexion utilisant Cubic sur un lien satellite dans un environnement sans congestion.

2.1.1.8 Comparaison des différents TCPs

Les Figure 2.4 et Figure 2.5 permettent d'observer les différences entre ces deux protocoles dans un environnement similaire et soulignent notamment les progrès accomplis sur l'état stable. Ainsi les nouveaux algorithmes de congestion, s'affranchissant de leur dépendance au RTT dont l'impact sur les communications par satellite est non négligeable, permettent une amélioration significative des performances de TCP dans cet état. La CWND moyenne avec Cubic est de 130

segments contre 100 pour TCP New Reno, ce qui signifie une amélioration du débit de 30% avec Cubic.

En revanche, les deux TCPs n'ayant modifié leur comportement que dans l'état stable ont des performances identiques pour les débuts de connexions, les reprises sur erreur ou encore l'établissement de connexion. Ces phases ayant une durée proportionnelle au RTT, les performances pour les connexions courtes continueront à pâtir fortement de sa valeur, quel que soit le CA utilisé.

2.2 Les TCP-PEPs : une solution dédiée au satellite

Face à l'inadéquation des solutions classiques et notamment de TCP New Reno, la communauté satellite a mis au point une solution spécifique, les TCP-PEPs [23], dont le but est d'utiliser de façon optimale la bande passante disponible, et d'apporter ainsi une solution aux dégradations liées aux propriétés inhérentes du segment satellite, en particulier leur long RTT [2].

2.2.1 Le principe des TCP-PEPs

Le concept général des TCP-PEPs est d'isoler le lien satellite des autres parties du réseau, en coupant la connexion de bout-en-bout en plusieurs connexions.

Pour ce faire, les TCP-PEPs utilisent deux architectures differentes :

- une solution distribuée qui utilise une combinaison de TCP-PEP serveur et TCP-PEP client, placés aux deux extrémités du lien satellite ;
- une solution intégrée qui utilise un unique TCP-PEP placé entre la Gateway et le serveur.

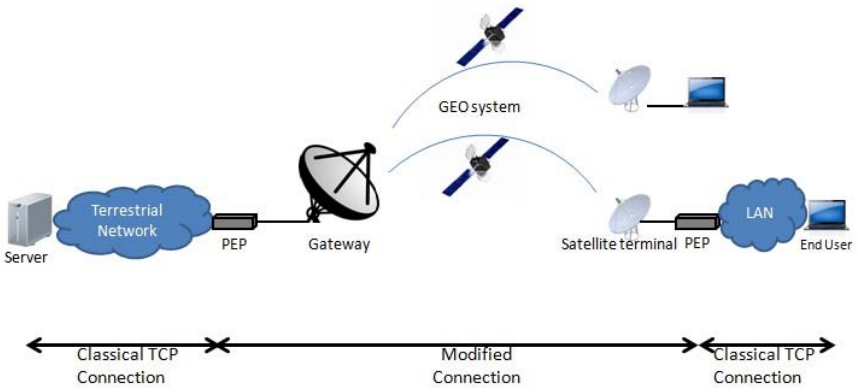

FIGURE 2.6 – Les deux différentes architectures des TCP-PEPs

La Figure 2.6 illustre ces différentes architectures. L'insertion de TCP-PEPs dans le réseau revient donc à scinder en deux ou trois la connexion TCP initiale. En effet, les TCP-PEPs se comportent comme un utilisateur TCP et accusent réception des segments reçus provenant de l'émetteur initial, avant de les émettre à nouveau vers le destinataire à travers une nouvelle connexion dédiée. La Figure 2.7 illustre les premiers échanges de segments qui ont lieu lorsqu'un TCP-PEP de type intégré est utilisé.

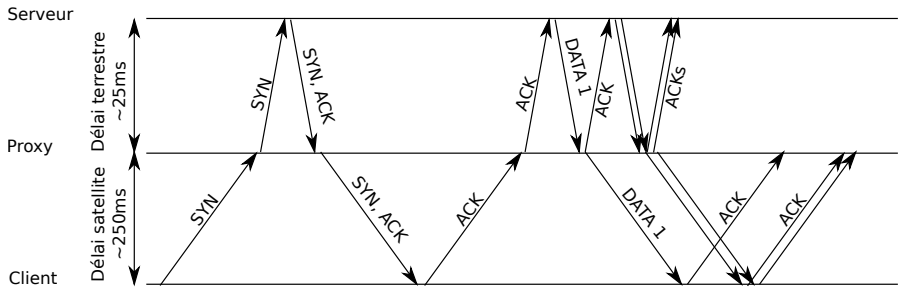

FIGURE 2.7 – Premiers échanges de segments illustrant l'utilisation d'un seul TCP-PEP

Dans la suite de l'étude, nous considérerons des TCP-PEPs distribués car ceux-ci sont plus représentatifs du contexte qui nous intéresse : l'insertion du lien satellite dans un contexte multi-technologies. Les principes exposés restent néanmoins valables pour les deux solutions.

2.2.2 Avantages et inconvénients des TCP-PEPs

Les avantages des TCP-PEPs

L'intérêt principal des TCP-PEPs réside dans l'amélioration significative de la performance de la connexion via l'ajout d'un mécanisme a priori peu intrusif. En effet, l'utilisation de TCP-PEPs ne nécessite pas de modification des piles des utilisateurs et leurs apparaît comme transparente. L'amélioration des performances est quant à elle une conséquence directe de la séparation en plusieurs connexions au rôle différent [24, 25].

Ainsi, en se plaçant entre l'émetteur et le lien satellite, le TCP-PEP permet de passer outre le long RTT en accusant réception des segments émis avant qu'ils n'empruntent le lien satellite. Le TCP-PEP peut ainsi solliciter l'émetteur pour recevoir les segments au débit qu'il souhaite sans être contraint par la longueur du RTT.

Une fois en possession des segments à retransmettre vers le récepteur, le TCP-PEP peut le faire de façon optimale en utilisant la connexion qui s'occupe spécifiquement du segment satellite et permet donc d'appréhender au mieux ce lien et ses contraintes en utilisant des protocoles adaptés. De nombreuses optimisations des TCP-PEPs existent d'ailleurs, visant à améliorer les communications entre les TCP-PEPs en modifiant la couche transport ou ses interactions avec les couches inférieures. Ainsi, les différents fournisseurs de TCP-PEPs améliorent le rendement de la connexion entre les TCP-PEPs en utilisant des modifications du protocole TCP, ou en se fondant sur des protocoles de transport différents tels que des versions modifiées d'User Datagram

protocol (UDP), ou encore eXplicit Control Protocol (XCP) [26] mais peuvent également utiliser des mécanismes d'interconnexion permettant notamment la gestion de la CWND en utilisant des informations fournies par la couche 2 [27].

Finalement, l'utilisation de TCP-PEPs permet d'augmenter le débit de la connexion de la même façon qu'une connexion locale et donc de remplir quasiment instantanément le débit du satellite alloué à l'émetteur en garantissant une forte Qualité de Service (QdS).

Les inconvénients des TCP-PEPs

Finalement, l'utilisation des TCP-PEPs s'avére très intrusive. En effet, l'amélioration des performances rendue possible par les TCP-PEPs en tronçonnant la connexion en plusieurs connexions ayant des émetteurs et des destinataires différents, se fait au détriment du principe de connexion de bout-en-bout [28], un concept fondamental des communications actuelles [28]. En brisant ce principe largement exploité par les protocoles et services des autres couches, les TCP-PEPs détériorent ou empêchent leur bon fonctionnement.

De nombreux protocoles nécessaires à la communication sont ainsi affectés. Ceux assurant la sécurité sont probablement parmi les plus touchés. En effet, afin de couper la connexion pour en ouvrir de nouvelles, les TCP-PEPs ont besoin d'accéder aux informations du segment telles que les adresses Internet Protocol (IP) ou certains champs TCP qui sont généralement chiffrés et connus uniquement des utilisateurs d'extrémité. Cela ne peut se faire qu'au travers d'une violation d'Internet Protocol Security (IPsec), le protocole de sécurité le plus largement employé. Ainsi, bien que des solutions proposant une utilisation conjointe d'une nouvelle version plus sûre du protocole IPsec et de TCP-PEPs soient régulièrement avancées [29, 30], les résultats continuent de montrer qu'utiliser les TCP-PEPs ne permet une amélioration des performances qu'au détriment de la sécurité.

Parallèlement à ces problèmes de sécurité, les solutions TCP-PEPs affectent également les protocoles de mobilité tels que Mobile IP [31], notamment dans le cas des réseaux hybrides satellite et terrestre avec des terminaux dual-mode, situation classique représentative des nouvelles tendances des télécommunications. Ainsi, dans le cas de la mobilité d'un des utilisateurs qui nécessiterait un changement du chemin, et notamment le passage d'un chemin avec lien satellite et donc des TCP-PEPs à un chemin sans (ou réciproquement), le TCP-PEP ou l'utilisateur final recevra des segments TCPs dont les entêtes ré-écrites par les TCP-PEPs ne correspondent pas à ce qu'ils attendent. Ces segments sont alors irrémédiablement supprimés pour des raisons de sécurité. La connexion TCP ne peut donc pas poursuivre de façon transparente et doit être à nouveau initiée au prix d'une importante dégradation de la QdS.

Par ailleurs, utiliser des TCP-PEPs ajoute un coût important pour les opérateurs qui doivent non seulement les installer, mais également les maintenir et les actualiser afin que leur utilisation n'aille pas à l'encontre de nouvelles améliorations protocolaires. Ce problème se retrouve d'ailleurs de façon plus générale dans l'ajout de dispositifs intermédiaires dans les réseaux. De nombreuses études ont ainsi été menées montrant que l'insertion de telles boîtes noires dans le réseau réduit

la marge d'optimisation des protocoles de bout-en-bout en annihilant leurs actions [32]. Dans le cas des TCP-PEPs par exemple, nous sommes en droit de nous interroger sur leurs réactions face à une entête TCP modifiée par l'utilisation d'une nouvelle option.

En conclusion, les TCP-PEPs ont été conçus alors que les performances des TCPs classiques ne permettaient pas d'utiliser de façon satisfaisante les satellites, et ont parfaitement joué leur rôle. Néanmoins, l'évolution des besoins et la volonté récente d'intégrer la technologie satellite dans un ensemble de technologies diverses, ne sont pas compatibles avec les solutions actuelles. Il est ainsi nécessaire de faire évoluer les solutions-types TCP-PEPs pour qu'elles n'entravent plus le fonctionnement des protocoles de bout-en-bout, ou d'améliorer les performances des solutions de bout-en-bouts telles que TCP.

Dans la partie suivante, nous mesurons les progrès accomplis par les nouvelles versions de TCP depuis que les TCP-PEPs sont devenus la solution prioritaire et incontestée du marché. Notre objectif est de pouvoir privilégier l'une des solutions précédentes.

2.3 Les TCP-PEPs faces aux solutions bout-en-bout

La Figure 2.8 représente le nombre de segments envoyés en fonction du temps pour des connexions évoluant dans un environnement satellitaire non-congestionné et pour différents protocoles de transport. Elle a été obtenue en utilisant le même banc de test que celui représenté par la Figure 2.3. Des TCP-PEPs commercialisés ont été ajoutés quand cela était nécessaire.

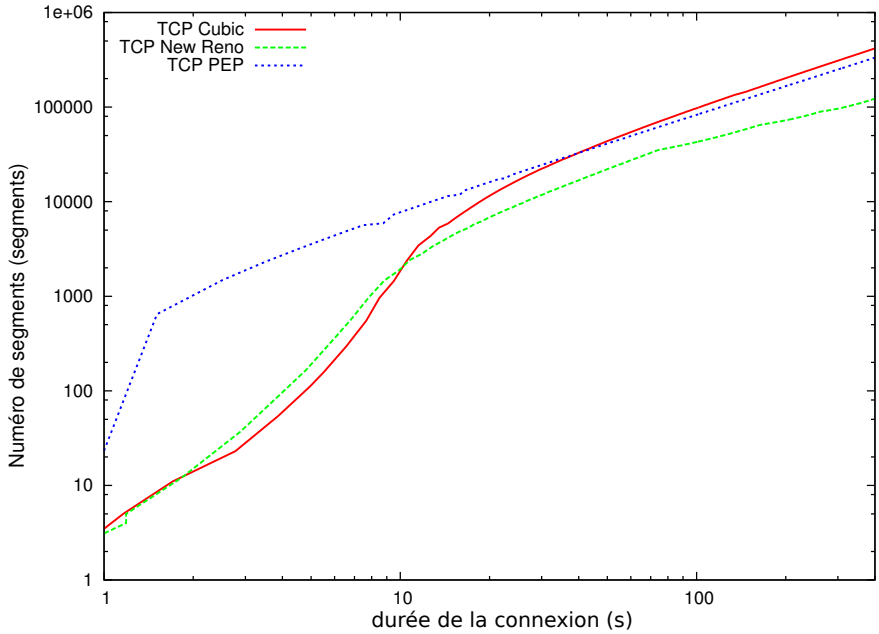

FIGURE 2.8 – Comparaison des solutions TCP-PEPs, de TCP New Reno et de Cubic

2.3.1 Le cas des connexions courtes

La transmission d'une connexion courte, c'est-à-dire d'une connexion dont la taille est inférieure ou égale à une dizaine de segments, se fait avec le protocole TCP dans son état de début de connexion. Ainsi, à l'exception d'une perte récupérée grâce au Fast Retransmit qui provoquerait le passage à l'état stable, le slow start est le seul mécanisme utilisé. Les nouvelles versions de TCP n'offrant pas d'améliorations notables, TCP New Reno et Cubic sont affectés de la même façon par le long RTT.

La Figure 2.8 illustre la différence notable de performances entre les solutions de type TCP-PEPs et les solutions de bout-en-bout. Ainsi, les TCP-PEPs permettant à la connexion de ne pas pâtir des contraintes propres aux satellites et d'atteindre très rapidement son débit maximal offrent un début de connexion beaucoup plus efficace que ces autres solutions qui voient l'augmentation de leur débit restreinte par la durée du RTT.

2.3.2 Le cas des connexions longues

Dans le cas des connexions longues, le protocole TCP évolue vraisemblablement dans son état stable suite aux pertes rencontrées en fin de Slow Start. La différence dans l'efficacité des algorithmes de contrôle de congestion s'exprime ici pleinement [33].

La Figure 2.8 montre que les TCP-PEPs continuent d'assurer un débit quasi-maximal, mais également que les nouveaux TCPs représentés ici par Cubic, permettent des performances similaires en étant beaucoup plus réactifs aux pertes, et beaucoup plus aggressifs dans leur CA. Ces deux solutions surpassent nettement TCP New Reno qui reste très sévèrement affecté par la croissance du débit inversement proportionnelle au RTT.

En conclusion, alors que les performances des premiers TCPs représentés ici par New Reno justifiaient le développement de nouvelles solutions, les améliorations récentes apportées aux algorithmes de contrôle de congestion ont montré qu'il est possible pour les solutions de bout-en-bout de rivaliser avec les solutions spécifiques. Néanmoins, tant que la phase de début de connexion ne sera pas elle aussi optimisée, la différence de performance dans le cas des connexions courtes continuera à justifier l'utilisation des solutions de type TCP-PEPs.

2.4 Problématique

Des études récentes [34, 35] menées à grande échelle sur des données réelles ont montré que les connexions courtes représentent 90% des communications Internet. Ainsi, transmettre de façon optimale ces connexions n'est pas un enjeu propre à la communauté satellite mais bien un enjeu majeur des télécommunications.

Le Slow Start est un mécanisme volontairement conservateur, conçu non pas pour transmettre une connexion courte mais bien dans le but de sonder le réseau et éviter de créer des perturbations

majeures. Cependant son fort coût en nombre de RTTs dégrade les performances pour l'ensemble des technologies.

Dans la suite de ce manuscrit, nous allons désormais considérer l'amélioration des performances pour les connexions courtes, indépendamment du support. Trouver une solution à ce problème devrait nous permettre d'une part d'améliorer les performances de l'ensemble des connexions TCPs, et d'autre part de répondre à notre problématique initiale : l'amélioration des protocoles de transport pour les réseaux satellite.

3 Les pistes d'amélioration de TCP dans le cas des connexions courtes

Dans le chapitre précédent, nous avons souligné le besoin d'amélioration des performances de TCP dans le cas du traitement des connexions courtes. Ainsi, alors que les nouvelles versions proposent des algorithmes de contrôle de congestion de plus en plus performants, les performances de TCP pour les connexions courtes souffrent toujours du caractère chronophage de ses autres mécanismes.

De nombreuses études ont été réalisées ou sont en attente de validation par les instances de normalisation afin de proposer des solutions plus en adéquation avec l'évolution des besoins et des tendances d'utilisation de l'Internet [36]. De fait, ces dernières ciblent l'efficacité de la transmission des connexions de faible taille.

Dans la suite de ce chapitre, nous détaillerons les solutions proposées pour améliorer les mécanismes de TCP autres que le CA et donnerons le gain éventuel en termes de RTT. Nous analyserons également les possibilités d'amélioration encore inexploitées.

3.1 Des solutions apportées à chaque état de TCP

3.1.1 L'établissement de connexion

Contrairement à la fin de connexion qui se trouve sollicitée lorsque l'émetteur n'a plus de données à émettre et n'influe donc pas sur le temps d'émission des données utiles, l'ouverture de connexion n'est qu'une étape préalable mais toutefois indispensable à la transmission des données par TCP. De nombreux chercheurs s'interrogent sur la possibilité d'utiliser cet échange de poignée de main pour émettre les premières données, et ainsi économiser un RTT.

TCP Fast open

Depuis la RFC 793 [37], il est possible d'ajouter des données au segment SYN. Ces données arrivent au récepteur qui doit toutefois attendre l'accusé de réception du segment SYN, qu'il a émis en retour, avant de transmettre les données aux applications. Cette attente est justifiée par la nécessité de parer à une potentielle réception dupliquée du segment SYN mais également

par la nécessité d'empêcher les éventuelles attaques dites de Spoofing durant lesquelles un pirate tente d'usurper l'adresse IP d'un autre utilisateur afin de pouvoir faire passer des paquets sur un réseau sans que ceux-ci ne soient interceptés par le système de filtrage (pare-feu). Ainsi, bien que le récepteur soit déjà en possession de l'information, celui-ci doit attendre 1 RTT avant de pouvoir l'exploiter et l'acquitter, et donc valider le passage à l'état de début de connexion.

TCP Fast Open [38, 39] est une solution proposée pour économiser ce RTT dans le cas où deux utilisateurs, ayant déjà préalablement établi et fermé des connexions entre eux, ouvrent une nouvelle connexion. D'après des analyses à grande échelle de données réelles entreprises par Google, ce cas de figure se retrouve dans 35 % des nouvelles connexions.

TCP Fast Open agit en 2 temps :

- lors du premier établissement de connexion, l'émetteur sollicite l'émission d'un jeton unique via l'utilisation d'une option TCP ;
- lors des établissements des connexions suivantes, l'émetteur joint à son segment SYN le jeton et les données à transmettre. Le récepteur est alors assuré de la provenance des données et peut les utiliser sans attendre un RTT.

Ce mécanisme controversé permet l'économie d'1 RTT et annihile donc complètement l'impact temporel de l'établissement de connexion dans 35% des cas. La validation de l'identité de l'émetteur étant nécessaire afin de se prémunir d'éventuelles attaques, arriver à un gain similaire pour le reste des connexions est un problème autrement plus complexe.

3.1.2 La reprise sur erreur

Améliorer la reprise sur erreur est particulièrement important dans le cas des connexions courtes. En effet, alors que chaque connexion courte a un impact limité voire quasiment nul sur la performance globale du réseau, sa performance individuelle est extrêmement dégradée par les pertes et les mécanismes de prévention de la congestion.

La faible quantité de segments transmis réduit notablement les chances de succès du Fast Retransmit et augmente donc les risques de devoir attendre un RTO de 3 secondes avant de pouvoir récupérer une perte [8].

Réduction du RTO

La première solution envisagée pour réduire cet impact est la diminution du RTO. Le RTO continue ainsi d'être calculé en fonction du RTT, mais sa valeur minimale et initiale n'est plus considérée égale à 3 secondes mais à 1 seconde. Ce nouvel RTO est suffisant pour que le réseau puisse se remettre d'une congestion et permet une meilleure prise en compte du RTT mesuré.

Les conséquences de cette étude ont été validées de façon empirique et cette optimisation fait maintenant l'objet d'une RFC [40].

Early Retransmit

Le mécanisme d'Early Retransmit fait partie d'une seconde catégorie de propositions qui visent

non plus à réduire l'impact de la non-détection d'une perte, mais à améliorer sa détection.

Early Retransmit [41] cible notamment les CWNDs faibles et est donc parfaitement enclin à améliorer les performances des connexions courtes. Il autorise la réduction, sous certaines conditions, du nombre de DupACKs nécessaires au déclenchement du Fast Retransmit et permet donc de récupérer une perte et de rentrer dans l'état stable dans de nombreux cas qui auraient alors nécessité l'attente d'un RTO et un retour dans l'état initial.

Early Retransmit utilise le nombre de segments préalablement envoyés n'ayant pas été validés ($OSEG$). Lorsque ce nombre est inférieur à 4 et qu'il n'y a plus de segment en attente d'émission, c'est-à-dire, dès lors que le nombre de DupACKs potentiels n'est pas suffisant pour récupérer une perte éventuelle via les mécanismes classiques, TCP active l'Early Retransmit et réduit le nombre de DupACKs nécessaires à $OSEG - 1$ segments.

Ainsi la perte de l'avant-dernier segment d'une connexion pourra être récupérée via le Fast Retransmit si l'accusé du dernier segment est correctement reçu. Ce mécanisme permet donc des gains significatifs dans le cas des connexions courtes, limitant considérablement les cas où l'attente du RTO est nécessaire.

Tail Loss Probe

Le mécanisme de Tail loss Probe [42] est complémentaire du mécanisme d'Early Retransmit. En effet, Tail Loss Probe permet de couvrir certains des cas où l'utilisation d'Early Retransmit est inefficace, et notamment le cas de la perte du dernier segment envoyé qui résulte systématiquement en l'attente d'un RTO et au retour à l'état initial.

Tail Loss Probe est un mécanisme implanté au niveau de l'émetteur qui permet à une connexion ne recevant pas d'ACK ni de DupACK pendant un certains temps d'émettre à nouveau le dernier segment dont elle a reçu l'ACK. La réception de cet ACK sous forme d'un DupACK avant l'expiration du RTO permet le déclenchement des mécanismes de reprise sur erreur et donc la poursuite dans l'état stable.

Ne pas attendre un RTO induit un gain de temps conditionné par la taille du RTT mais l'amélioration des performances associée à la poursuite en CA reste quoi qu'il en soit très significative.

3.1.3 L'état de début de connexion

Lorsqu'il fait suite à l'établissement de connexion, l'état de début de connexion consiste en l'utilisation d'un Slow Start initié avec une IW égale à 3 [7]. En revanche, dans le cas d'un retour à l'état de début de connexion causé par l'expiration d'un RTO, le Slow Start est lancé avec une IW de 1 segment.

Le Slow Start n'a pas pour objectif d'être performant dans la transmission de connexion courte et privilégie un sondage progressif du réseau à une performance immédiate. Il est adapté aux connexions longues en leur permettant une augmentation de débit progressive, ciblant un

partage équitable de la bande passante entre les différents utilisateurs. Ainsi, la transmission des premiers segments de la connexion est très coûteuse en nombre de RTTs.

La Table 4.1 montre le nombre de RTTs nécessaires à la transmission d'une connexion courte en fonction de la taille de l'IW considérée.

taille du flux	IW=1	IW=3	IW=10
1 segment	1	1	1
3 segments	3	2	1
10 segments	4	3	1

TABLEAU 3.1 – Nombre de RTTs nécessaires à la transmission d'une connexion courte

De nombreux mécanismes de démarrage rapide de TCP ont donc été proposés pour améliorer l'efficacité des débuts de connexion et réduire la durée des connexions courtes [43, 44]. Comme nous allons le voir, ces derniers, bien qu'ils soient efficaces dans la tâche qui leur est dévolue, présentent tous des inconvénients majeurs qui restreignent ou empêchent leur utilisation.

Quick Start

Quick Start [45] utilise des informations explicites retournées par les routeurs rencontrés pour adapter son débit et optimiser l'émission des données sans utiliser le Slow Start.

L'émetteur envoie une requête Quick Start dans une option IP contenant le débit d'émission souhaité. Chaque routeur parcouru peut alors soit accepter la requête s'il juge son débit disponible supérieur à celui demandé, soit la modifier en indiquant son débit disponible maximal, soit encore la rejeter ou l'ignorer s'il n'a pas connaissance de Quick Start. Le résultat de la requête est ensuite renvoyé à l'émetteur dans une option TCP et un contrôle est fait pour s'assurer que chaque routeur a effectivement répondu à la question. Le résultat d'une requête ignorée par l'un des routeurs ne peut pas être utilisé par l'émetteur.

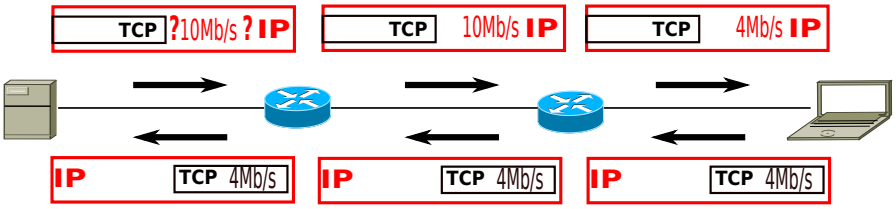

FIGURE 3.1 – Quick Start : Demande du débit disponible

La Figure 3.1 illustre la demande de débit initial de Quick Start et montre le cas où le second routeur ne peut pas écouler le débit demandé.

Ce mécanisme d'interaction entre les couches 3 et 4 nécessite donc la compréhension et l'approbation de l'ensemble des routeurs traversés, sinon le Slow Start classique doit être utilisé. Cette contrainte réduit fortement les chances de déploiement et de succès de Quick Start.

Jump Start

Jump Start utilise une estimation des données prêtes à être envoyées ainsi que du RTT afin de disperser de façon régulière les segments tout le long de ce dernier sans utiliser le Slow Start [46].

Ce mécanisme particulièrement efficace dans les réseaux sans congestion s'avère trop agressif dans les environnements congestionnés. Emettre un nombre trop important de segments dans un réseau congestionné augmente en effet la congestion et cause de sévères dégradations de performance. Jump Start est alors responsable de performances pires que le Slow Start classique.

Augmentation de l'IW

L'augmentation de la taille de la CWND est un sujet récurrent [7, 47, 48], intimement lié à l'évolution des tendances d'utilisation de l'Internet. Ainsi, en 2002, répondant au constat que la majorité des objets WEB avaient alors une taille inférieure a 4kB, la RFC 3390 préconisa de cesser l'utilisation d'une IW d'1 segment, pour passer à une IW de 3 segments. La tendance étant à l'augmentation de la taille moyenne de l'objet WEB, la RFC 6928 [49] préconise maintenant l'adoption d'une IW de 10 segments.

Contrairement aux mécanismes précédemment cités, l'augmentation de l'IW ne remplace pas l'utilisation du Slow Start. L'objectif est de permettre la transmission la plus rapide possible des connexions courtes (connexions de taille inférieure ou égale à 10 segments et 15kB), en augmentant le nombre de segments pouvant être émis dès l'établissement de la connexion. Le Slow Start reprend ensuite son cours normal lors de la réception des premiers ACKs.

De nombreuses études [35] ont vanté les mérites de cette légère modification qui permet l'émission en 1 RTT de l'ensemble de la connexion courte et donc une économie pouvant aller jusqu'à 2 RTTs (cf. laTable 4.1) dans 90% des émissions de requêtes HyperText Transfer Protocol (HTTP), ce qui, dans l'exemple des communications par satellite, équivaut à un gain de temps de plus d'1 seconde. Dans un environnement non congestionné, émettre l'ensemble de la connexion de façon continue est sans conteste la solution la plus performante.

Néanmoins, de nombreux chercheurs s'inquiètent des conséquences d'un tel mécanisme dans certains scénarios et notamment dans les cas congestionnés. A la différence de Jump Start, ce n'est pas tant la réaction du réseau face à la quantité de données envoyées qui pose question, celle-ci demeurant faible, mais la façon avec laquelle il va être capable d'absorber le "burst" initial, c'est-à-dire l'émission à un débit plus élevé que le débit réel du réseau d'un nombre important de segments. Beaucoup préconisent donc la conservation d'une IW de 3 segments [50].

Afin de mettre en évidence l'impact réel de la taille de l'IW sur un réseau congestionné, nous avons réalisé des simulations avec Network Simulation 2 (NS2) utilisant une topologie de réseau Dumbbell simple semblable à celle présentée par la Figure 3.2 avec un lien à 5 Mb/s constituant un goulot d'étranglement et un RTT de 500ms avec un taux de perte de 5%. Nous présenterons plus en détail notre banc de tests dans la partie suivante. 13 serveurs et utilisateurs terminaux avec un débit d'émission de 20 Mb/s sont utilisés pour engendrer de la congestion et réaliser nos

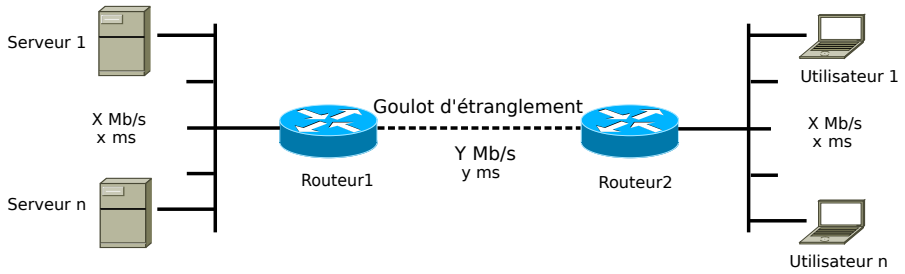

FIGURE 3.2 – Topologie "Dumbbell" utilisée dans nos simulations

tests.

La Figure 3.3 compare le temps de délivrance, c'est-à-dire le temps nécessaire à la transmission de la totalité d'un flux sans tenir compte de l'ouverture de connexion, pour 4 tailles d'IW. Sans nous attarder sur l'allure atypique des courbes pour une IW égale à 6 et 10 segments, nous pouvons d'ores et déjà constater que dans ce milieu congestionné, le temps de délivrance des flux courts est meilleur avec une IW de 6 segments plutôt qu'avec une IW de 10 segments, justifiant les craintes sur le bien-fondé de l'augmentation de l'IW.

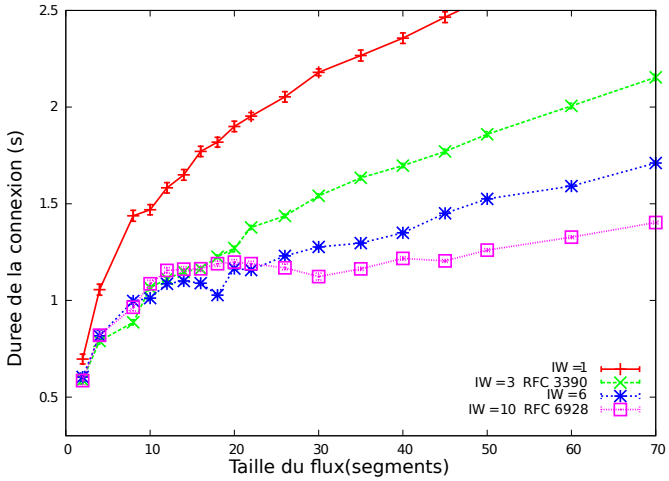

FIGURE 3.3 – Comparaison des durées de transmission des flux en fonction de l'IW

En conclusion, de nombreuses solutions ont été proposées pour palier la latence excessive introduite par TCP durant les débuts de connexions. Néanmoins, les mécanismes de démarrage rapide censés permettre la transmission efficace des connexions courtes pendant la phase de début de connexion ne donnent pas satisfaction et les espoirs suscités par l'augmentation de l'IW restent déçus.

Dans la partie suivante, nous analyserons plus finement la relation entre les bursts et le

comportement de TCP afin de comprendre la dégradation de la performance résultant de l'augmentation de l'IW.

3.2 Les bursts : acteurs majeurs des performances de TCP

La notion de rafale ou burst est très communément employée pour justifier des performances d'une connexion TCP. On parle d'ailleurs de trafic sporadique ou "bursty traffic" pour qualifier le comportement du trafic TCP entre deux utilisateurs.

3.2.1 les bursts et TCP

Les échanges entre extrémités TCP, conditionnés par le rythme des accusés de réception, évoluent en créant et utilisant des bursts de nature et aux conséquences différentes. Nous allons notamment regarder le cas de deux types de bursts différents, ceux créés par l'émission de l'IW et ceux dus à l'émission des CWNDs.

Cas de l'émission de CWND en Slow Start

Durant le Slow Start, l'émetteur reçoit les accusés de réception au rythme du goulot d'étranglement, c'est-à-dire au rythme du lien à plus faible débit, et transmet à son propre débit deux nouveaux segments pour chaque ACK reçu dans l'ordre. En considérant le cas de figure raisonnable où le débit du goulot d'étranglement est au moins deux fois plus faible que celui de l'émetteur, les nouveaux segments émis le sont avec un débit deux fois supérieur au lien à plus faible débit. Ainsi, le routeur de ce lien reçoit 2 segments dans le temps qui lui est nécessaire pour en traiter un seul. Il doit donc ajouter à la file d'attente de son buffer le second segment reçu afin de pouvoir finir de traiter le premier. Une CWND de W segments est donc responsable d'une augmentation maximale de la file d'attente du buffer de $\frac{W}{2}$ segments.

Par ailleurs, l'émission de la CWND n'occupant pas l'ensemble du RTT, le buffer aura le temps de se vider avant de recevoir les nouveaux segments autorisés par les accusés de réceptions.

Cas de l'émission de l'IW

Dans le cas de l'émission de l'IW, les segments sont tous émis au débit de l'émetteur. Ainsi, en fonction de la différence entre le débit d'émission et le débit limitant, l'émission de W segments peut ajouter jusqu'à $W - 1$ segments à la file d'attente du buffer quand ce nombre ne peut dépasser $\frac{W}{2}$ segments lors de la transmission régulière de CWND.

L'augmentation de l'IW et donc du burst initial sont ainsi directement corrélées à l'impact effectif de la connexion sur le réseau. Dans un réseau congestionné, le buffer du routeur du goulot d'étranglement peut atteindre un fort taux d'occupation, l'augmentation de l'IW et donc du nombre de segments qui grossissent sa file d'attente accroissent la probabilité de rejet des segments et donc la probabilité de dégradation de la performance, comme illustrée par la Figure 3.3.

3.3 Le Pacing : un mécanisme de prévention des bursts

Dans les années 2000, le Pacing a été proposé pour contrer les effets indésirables des bursts sur les performances des réseaux [51, 52, 53].

3.3.1 Principe et objectifs du Pacing

Le principal objectif du Pacing est d'éviter l'émission de bursts autant que possible en étalant régulièrement l'émission des segments sur le RTT. Un calcul est fait chaque RTT fondé sur la taille de l'IW et de la CWND pour déterminer l'espacement entre deux émissions, et ce jusqu'à ce que cet espacement soit inférieur au temps d'émission d'un segment.

Le Pacing vise à augmenter le débit de la connexion en minimisant l'impact des segments envoyés sur le routeur du lien qui crée le goulot d'étranglement, ce qui aura pour conséquence de limiter les pertes isolées dues aux bursts. La transmission du flux est étalée et n'engendre pas de surcharge momentanée de la capacité de traitement du routeur. Dans l'idéal, les performances individuelles sont améliorées par la réduction du taux de perte et les performances générales du réseau le sont par un meilleur partage des buffers entre les différents flux en compétition.

3.3.2 Inconvénients du Pacing

Une étude a néanmoins montré que ce qui faisait en théorie la force du Pacing, était en fait la cause d'inconvénients majeurs [1]. Cette étude a sonné le glas des ambitions de ce mécanisme. Il nous semble important de revenir ici sur ces principaux inconvénients.

Synchronisation des flux

Dans la partie précédente nous avons montré l'évolution de la transmission d'un flux TCP et notamment souligné le rôle prépondérant des bursts dans les pertes de segments. Ces bursts peuvent avoir des conséquences préjudiciables sur les performances comme nous avons pu le voir avec la Figure 3.3 mais sont paradoxalement nécessaires au bon fonctionnement de TCP. Ainsi, TCP et notamment ses dernières versions utilisent les pertes pour étalonner leur débit et partager le débit de façon équitable. En étalant et lissant le trafic, le Pacing réduit les pertes au minimum et donc les indications de congestion, ce qui permet à chaque connexion d'augmenter son débit. Finalement, le Pacing ne fait que retarder la détéction d'événement de congestion jusqu'à ce que le réseau soit complètement saturé. La capacité d'accueil du buffer du lien bottleneck se retrouve donc dépassée par les débits atteints par chaque connexion et tous les nouveaux segments se retrouvent rejetés.

Ceci a de fortes implications :

– Les pertes successives dans un même flux dues à la saturation du réseau vont empêcher le déclenchement des mécanismes de reprise sur erreur et entraîner l'attente du RTO, pénalisant ainsi les débits individuels.

– Les pertes de segments touchant des flux multiples vont engendrer une synchronisation des flux : tous les flux réduiront en effet drastiquement leur débit simultanément. Le réseau passera donc d'un état de surcharge à un état de sous-exploitation et les flux auront tendance à repartir simultanément dans le même état.

La Figure 3.4 illustre les effets du Pacing à travers les performances de flux de différente taille. Afin de pouvoir comparer l'évolution des conséquences du Pacing selon la taille des flux transmis, les résultats des simulations ont été normalisés. Pour cela, la durée moyenne de transmission de ces flux est divisée par une estimation de la durée idéale de délivrance d'une même taille de flux, c'est-à-dire du temps nécessaire à une connexion pour transmettre la totalité de son flux en évoluant en slow-start avec une IW de 3 segments jusqu'à atteindre et conserver une CWND permettant un partage équitable de la bande passante entre les différents utilisateurs.

NS2 est utilisé pour simuler une topologie "Dumbbell" comme illustrée par la Figure 3.2 où, cette fois, 20 flux de taille variable utilisant TCP New RENO sont émis en parallèle entre 20 émetteurs et récepteurs différents situés de part et d'autre d'un goulot d'étranglement de sorte à créer une congestion. Dès qu'un flux se termine, un nouveau flux est établi entre les deux mêmes nœuds d'extrémité après un temps d'attente moyen d'1 seconde. Le débit du goulot d'étranglement est 25 Mbit/s tandis que le débit des autres liens est égal à 100 Mbit/s. Le RTT moyen est de 100ms et le buffer du routeur d'entrée du goulot d'étranglement est égal à un quart de produit délai-bande passante. Par ailleurs, les flux sont soit tous avec Pacing soit tous sans Pacing, de sorte à n'ajouter aucune incertitude sur l'équité du mécanisme, celle-ci étant vérifiée ultérieurement.

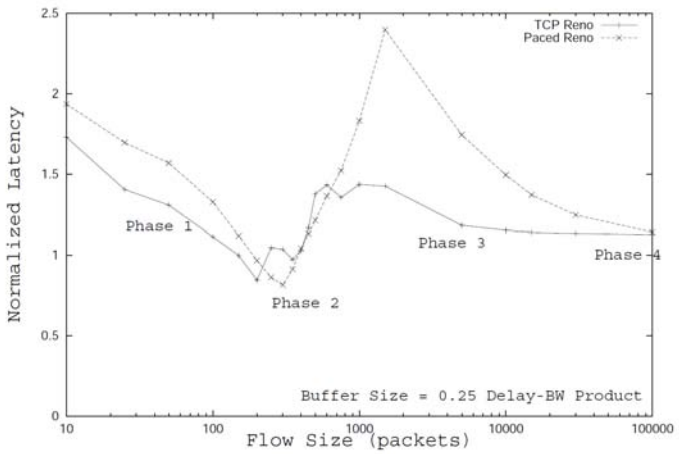

FIGURE 3.4 – Durée normalisée de la transmission de différentes tailles de flux avec et sans Pacing. Figure tirée de [1]

Quatre phases distinctes permettent de constater les effets du Pacing :

1. Phase 1 : Considérant des flux de taille relativement faible, les connexions, qu'elles soient

avec ou sans Pacing, ne subissent pas de pertes et restent majoritairement en Slow Start. La durée légèrement plus importante des flux avec Pacing est due à l'étalement des segments sur le RTT.

2. Phase 2 : Les bursts induits par l'évolution en Slow Start des flux sans Pacing provoquent les premières pertes et causent la sortie du Slow Start pour le passage probable à l'état stable. Les connexions avec Pacing, non sujettes à ces pertes isolées, restent en Slow Start et augmentent leur débit. Le Pacing permet ainsi d'atteindre momentanément de meilleures performances.
3. Phase 3 : Les flux avec Pacing ayant continué à augmenter leur débit saturent complètement le réseau. Ils subissent des pertes nombreuses et simultanées et recommencent tous ensemble en Slow Start après une période d'inactivité correspondant à l'attente d'un ou plusieurs RTOs. La taille des flux à transmettre augmentant, il est probable que la synchronisation engendrée entraîne la répétition de la situation précédente.
4. Phase 4 : La taille très importante du flux à transmettre amoindrit les effets de synchronisation, et les deux solutions tendent vers des performances similaires.

Ainsi en diminuant les pertes isolées et retardant la congestion au maximum, le Pacing réduit sur le long terme l'indépendance des flux et donc leurs performances individuelles et collectives. Les connexions longues sont très significativement touchées : les performances peuvent être deux fois moindres avec que sans Pacing.

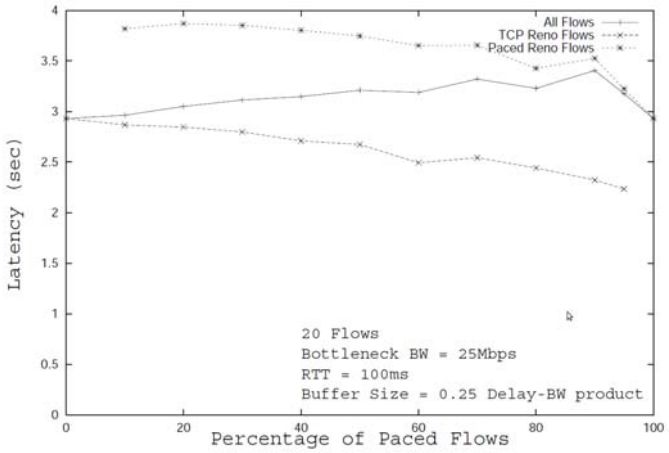

FIGURE 3.5 – Durée de la transmission de flux de 300 segments en fonction du pourcentage de flux utilisant le Pacing. Figure tirée de [1]

Equité du Pacing

La Figure 3.5 montre le comportement des connexions avec et sans Pacing en fonction du

pourcentage de chaque type de flux. Une vingtaine de flux de 300 segments sont émis en continu. Il est évident que les connexions avec Pacing souffrent de la comparaison avec les connexions sans Pacing. Les auteurs de l'article suggèrent qu'en étalant les segments sur la longueur du RTT dans un environnement où les flux sans Pacing ont engendré des bursts, on augmente finalement la probabilité qu'un des segments du flux se retrouve dans une situation de congestion momentanée du buffer et soit donc rejeté.

Dans sa volonté de ne pas subir les bursts, le Pacing a mis de côté le fait que les pertes sont les seuls événements à même de signaler une congestion. Ainsi, en les retardant au maximum, le Pacing prive TCP de son moyen de régulation ce qui s'avère autrement plus préjudiciable pour l'ensemble des connexions que quelques pertes isolées.

3.4 Conclusion de l'état de l'art

L'état de l'art effectué conjointement avec l'observation des différents protocoles et mécanismes dans divers environnements nous a permis de modifier notre objectif initial. En effet, alors qu'auparavant l'amélioration des performances des communications par satellite était notre objectif principal, elle pourra désormais être vue comme une retombée de l'amélioration des performances de TCP dans un contexte général.

TCP, le protocole de transport majoritairement utilisé dans les réseaux, souffre en effet de mauvaises performances dans la transmission de connexions courtes. Dans le cas de l'Internet, ces dernières représentent pourtant 90 % des connexions établies [35], pour un volume d'environ 40% des données échangées. Améliorer ces performances est donc un enjeu majeur de ces dernières années.

Pour autant, les nombreux mécanismes de démarrage rapide de TCP ayant été proposés ont tous des défauts rendant leur démocratisation et leur utilisation à grande échelle très incertaines. L'augmentation de l'IW, mécanisme déja communément utilisé dans les nouveaux Systèmes d'exploitation (OS), souffre notamment de la hausse de la taille des bursts. Cette dernière semble être responsable d'une augmentation de la probabilité de pertes associée à la réduction de l'efficacité des mécanismes de reprise sur erreur, et donc d'une sévère dégradation de la performance moyenne.

Considérant l'impact important de ces bursts sur les performances, une solution appelée Pacing visant à empêcher autant que possible les bursts a donc été étudiée. Cette dernière montre que l'appréhension des bursts n'est pas chose aisée, car autant leurs présences excessives en début de connexion réduisent la performance individuelle des flux courts, autant leur disparition dégrade incontestablement les performances à long terme.

La Figure 3.6 récapitule les différents mécanismes présentés dans cet état de l'art ainsi que leurs domaine d'application respectif.

Dans la suite de ce manuscrit, nous proposons un mécanisme appelé Initial Spreading reposant

sur une étude précise des bursts qui permet de s'affranchir de leurs conséquences et offrent des améliorations significatives des performances dans le cas des connexions courtes.

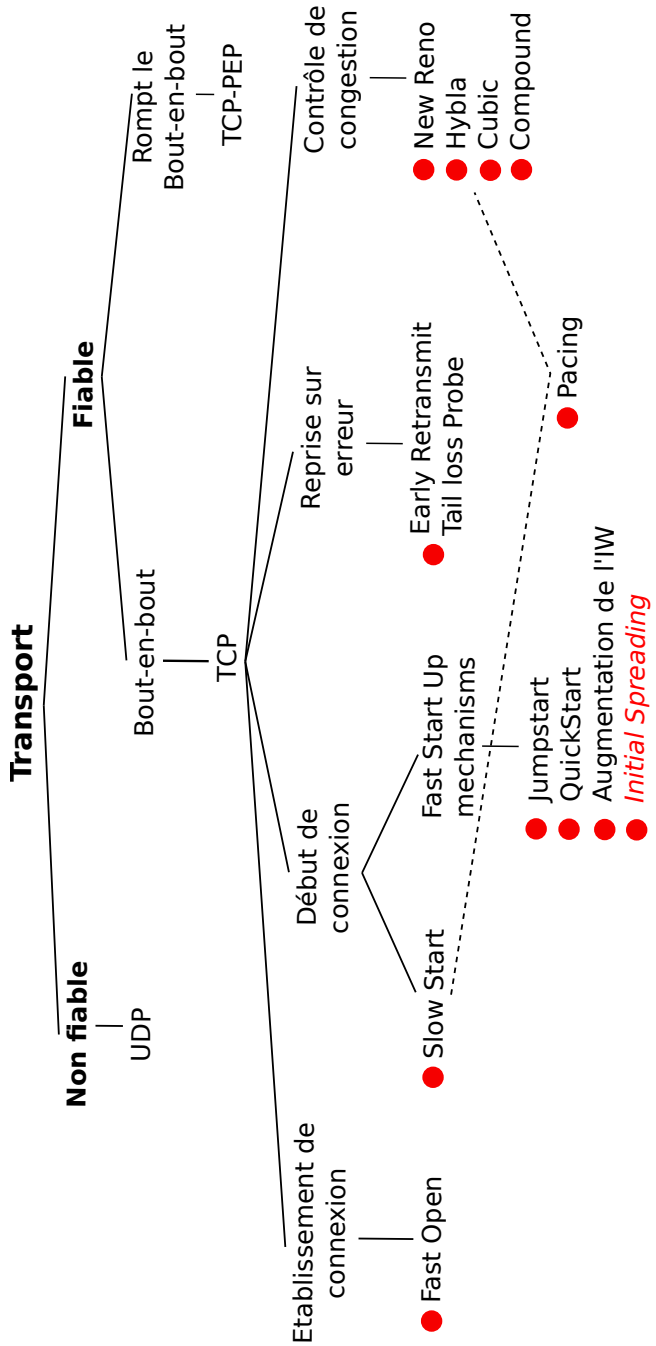

FIGURE 3.6 – Récapitulatif des différents protocoles ou mécanismes de la couche Transport considérés

4 Initial Spreading : concept et premières simulations

4.1 Association du Pacing et d'une IW accrue

Dans le chapitre précédent, nos études du Pacing et de l'augmentation de l'IW ont souligné les conséquences antagonistes des bursts sur la performance d'une connexion TCP. L'augmentation de l'IW souffre ainsi de la présence de bursts tandis que le Pacing souffre de leur absence. Il est donc tout naturel de se demander ce qui résulterait d'une combinaison des deux mécanismes.

Dans un premier temps, nous allons comparer d'un point de vue théorique les conséquences effectives des différents mécanismes sur le routeur d'entrée du goulot d'étranglement.

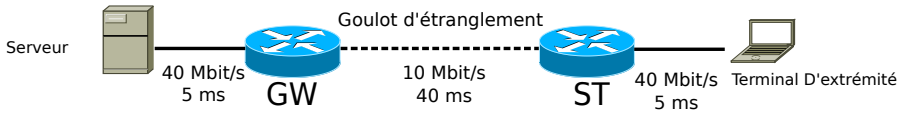

FIGURE 4.1 – Topologie simple utilisée pour l'évaluation de l'impact des différents mécanismes sur l'évolution de la file d'attente

Cette évaluation faite en Matlab est fondée sur une topologie de réseau minimale à 3 liens telle qu'illustrée par la Figure 4.1 : 1 goulot d'étranglement avec un débit à 10 Mbit/s entouré de 2 liens à 40 Mbit/s. Le RTT choisi est de 100ms. Plusieurs cas sont représentés : les cas sans Pacing pour différentes IW, le cas du Pacing tel qu'il a été originellement conçu, c'est-à-dire associé à un Slow Start classique débutant avec une IW d'1 segment et finalement les cas associant Pacing et une IW de grande taille.

La Figure 4.2 illustre l'évolution du nombre maximal de segments devant être ajoutés à la file d'attente avant de pouvoir être traités, à chaque RTT, pour une connexion unique.

Cette figure souligne les conséquences du mécanisme adopté sur l'état du buffer au fur et à mesure qu'augmente la quantité de données envoyée. L'augmentation de la taille de l'IW a des répercussions immédiates sur le nombre de segments à ajouter dans la file d'attente, tandis que l'utilisation du Pacing permet l'émission et le traitement d'un grand nombre de segments sans que ceux-ci n'aient d'incidence directe sur le remplissage du buffer. Son utilisation permet

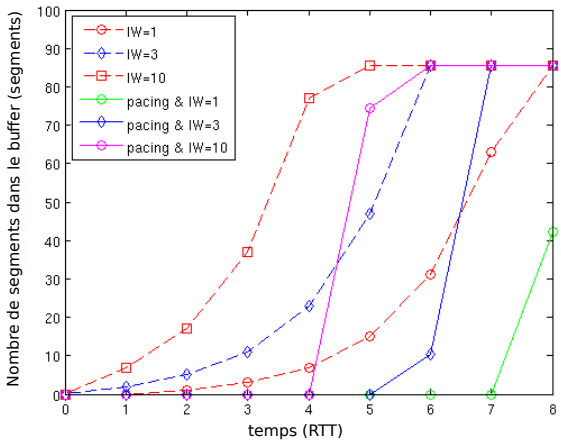

FIGURE 4.2 – Nombre de segments ajoutés dans le buffer par un flux en Slow Start

notamment de compenser une grande IW lorsque les deux mécanismes sont associés.

Dans un réseau congestionné, le buffer d'entrée du goulot d'étranglement a un fort taux d'occupation moyen et est donc très sensible aux mécanismes TCP qui nécessitent une "bufférisation" importante. Par conséquent, une IW de 10 segments, qui modifie l'état du buffer de façon importante en nécessitant l'ajout de plusieurs segments à une file d'attente presque pleine, est susceptible d'engendrer plus de pertes qu'une IW de 3 segments. Augmenter l'IW risque donc de réduire la performance moyenne.

Les scénarios associant le Pacing à l'augmentation de l'IW ne souffrent pas de ces bursts initiaux. Cependant, ces solutions risquent fortement de présenter les mêmes inconvénients que le Pacing dans le cas des connexions longues car elles permettent une augmentation très rapide et significative de la CWND et donc une saturation du réseau toujours plus importante et préjudiciable.

L'utilisation combinée du Pacing et d'une augmentation de l'IW semble donc capable de résoudre les défauts inhérents aux grandes IWs et devrait donc se révéler très compétitive dans le cas de la transmission de flux courts. En revanche, il est fort probable que cette solution subisse de plein fouet les défauts du Pacing dans le cas des connexions longues.

Les sections précédentes ne se contentent pas de mettre en exergue les faiblesses et les forces des différents concepts, elles dessinent également le contour d'un nouveau mécanisme. Celui-ci doit prendre en compte l'ensemble des effets des bursts pour permettre une amélioration significative des performances de TCP dans la transmission des flux courts, quel que soit l'état de congestion du réseau, sans pour autant dégrader la transmission des flux longs.

4.2 Présentation de l'Initial Spreading

L'idée de base de notre mécanisme, intitulé Initial Spreading [3], est l'étalement d'une grande IW de n segments sur le premier RTT avant de laisser TCP continuer de façon conventionnelle, soit en Slow Start, soit dans l'état de reprise sur erreur, soit encore dans son état stable.

De façon imagée, cela revient à considérer l'émission légèrement décalée de n connexions évoluant en Slow Start avec une IW unitaire jusqu'à l'occurrence de la première perte. Cette dernière affecte la totalité de la connexion en déclenchant les mécanismes de reprise sur erreur et permet une réponse efficace à la détection de l'événement de congestion. Cette image n'est pas complètement anodine, puisque des études [35] ont par exemple montré que les navigateurs Web les plus populaires tels qu'Internet Explorer et Firefox ouvrent régulièrement de nombreuses connexions en parallèle afin de pallier la lenteur des mécanismes classiques et ainsi améliorer leur durée de téléchargement. A la différence de notre proposition, cette stratégie empêche TCP d'entrer dans son état stable [54] et finalement dégrade la performance globale du réseau.

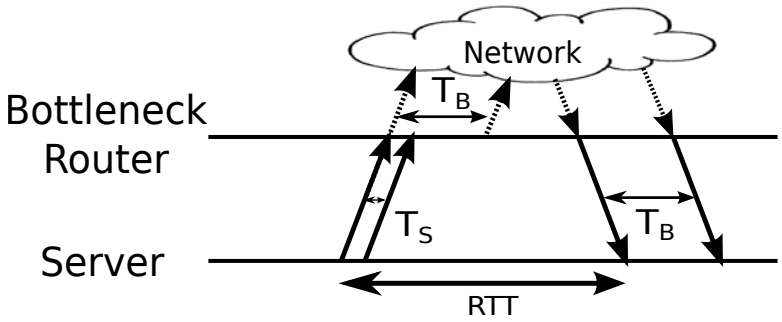

FIGURE 4.3 – Illustration des différents temps utilisés dans le manuscrit

Dans la suite du manuscrit, T_B et T_S représentent respectivement le temps d'émission d'un paquet par le routeur du goulot d'étranglement et par l'émetteur, comme illustré par la Figure 4.3. Ces deux valeurs sont donc indépendantes de la congestion du réseau.

La Figure 4.4 montre le comportement des différents mécanismes lors de la transmission de 12 segments. La taille de l'IW est fixée à 4 segments dans les trois cas de figure proposés. $T_{Spreading}$, le temps entre l'émission de deux segments dans le cas d'une utilisation de l'Initial Spreading, est pris égal à l'espacement régulier du Pacing. L'échange de SYN-SYN-ACK est utilisé par le Pacing et l'initial Spreading pour mesurer le RTT et en déduire ainsi l'espacement entre deux segments successifs, ce dernier valant alors $T_{Spreading} = \frac{RTT}{IW}$.

La Figure 4.4 illustre donc par l'exemple les trois comportements des mécanismes dans la transmission des bursts :

– Cas d'une grande IW : l'IW est transmise en un unique burst, puis chaque ACK permet l'émission d'un mini-burst de deux segments.

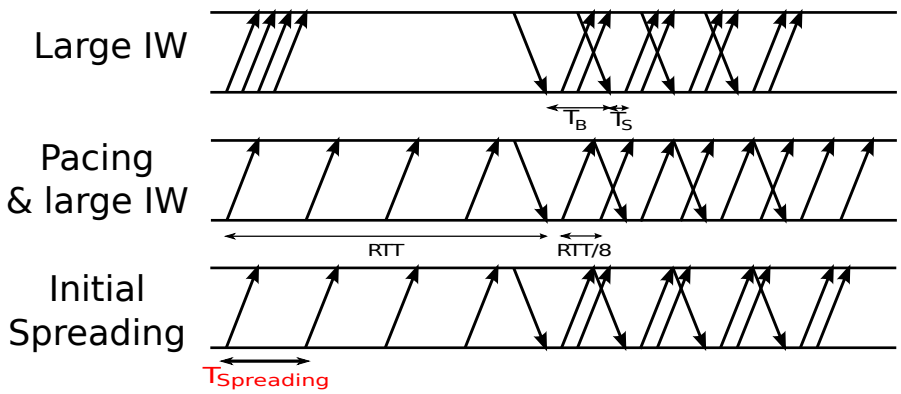

FIGURE 4.4 – Chronogramme représentant la transmission de 12 segments avec une IW de 4 segments en utilisant différents mécanismes

– Cas de l'association du Pacing et d'une grande IW : la totalité du flux est transmise sans burst.
– Cas de l'Initial Spreading : l'IW est transmise sans burst, puis la réception des accusés de réception dicte l'émission de mini-bursts de deux segments.

L'objectif de l'Initial Spreading est double :

– empêcher les bursts dans la transmission de la connexion courte afin de décorréler et minimiser les pertes des segments et donc d'optimiser les performances ;
– permettre la création de bursts dès le second RTT afin de ne pas être affecté par leur absence et de pouvoir réguler le débit de la connexion grâce aux détections des événements de congestion.

La Figure 4.5 dépeint la façon avec laquelle les différents protocoles modifient l'état de la file d'attente du buffer du goulot d'étranglement lors d'un Slow Start initié avec une IW de 10 segments. Le scénario est le même que celui de la Figure 4.2.

La différence majeure entre le Pacing et l'Initial Spreading est la création de bursts dès le second RTT qui vont peser sur le remplissage du buffer. Nous pensons que ces mini-bursts seront en mesure d'endiguer les défauts du Pacing, tout en garantissant de très bonnes performances pour les connexions courtes.

Dans la partie suivante, nous utilisons le simulateur NS2 afin de mener une première validation empirique de notre mécanisme ainsi que de nos hypothèses sur l'impact des bursts.

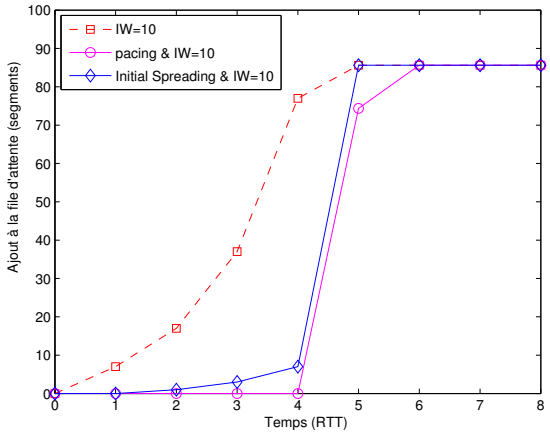

FIGURE 4.5 – Évolution des files d'attente pour une IW de 10 segments, avec et sans Pacing et avec Initial Spreading

4.3 Validation Empirique par simulation

4.3.1 Banc de tests utilisé

NS2 est le simulateur de réseau le plus largement employé, mis à jour et partagé par la "communauté réseau", la plupart des dernières versions de TCP y sont implantées. Cela l'avantage face à des simulateurs plus récents et performants tels que NS3 et OMNET.

Afin de garantir une certaine fiabilité à nos résultats, nous avons pris soin d'utiliser les "graines" de NS2 qui nous permettent de nous assurer que chaque mécanisme est testé dans les mêmes conditions, malgré le nombre important d'itérations effectuées. Un intervalle de confiance à 95% est tracé pour chacun des points. De très nombreuses valeurs de délais et de débits ont été considérées dans nos simulations afin d'observer le comportement d'Initial Spreading sous différentes conditions de trafic et de charge.

Finalement, nous avons choisi d'utiliser la version Cubic de TCP pour la plupart de nos simulations, tout en prenant garde que nos observations restent valables pour les autres versions de TCP.

La Figure 4.6 décrit la topologie employée dans la plupart de nos simulations et expériences. Cette dernière correspond à celle utilisée dans le chapitre précédent [1], ce qui nous permet de nous placer dans les conditions qui ont révélé les faiblesses du Pacing et de pouvoir ainsi confronter nos résultats.

Afin d'observer l'Initial Spreading évoluer dans un environnement congestionné mais réaliste, nous avons pris soin de créer une congestion à partir de connexions TCP illimitées établies entre S_i et R_i avec $i \in [13, 15]$. Nous attendons alors plusieurs secondes avant de lancer les flux de

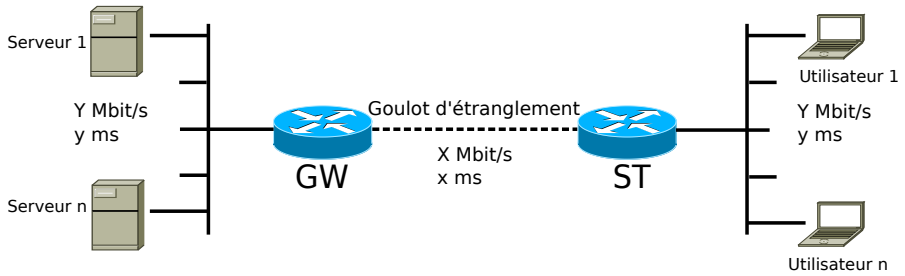

FIGURE 4.6 – La topologie Dumbbell retenue dans la plupart des simulations et expériences

tests afin de laisser le temps aux 3 flux en compétition d'entrer en régime stable et ainsi de se partager équitablement et efficacement le débit disponible. Nous avons ensuite mesuré l'effet des différents mécanismes en fonction de la taille des connexions. Pour ce faire, nous avons lancé 12 flux d'une taille comprise entre 1 et 100 segments à des temps et intervalles sélectionnés de façon aléatoire entre S_i et R_i avec $i \in [1, 12]$. Les flux courts peuvent ainsi interagir ou non entre eux, ce qui est assez proche de la réalité. Finalement, nous avons pu calculer le temps de délivrance moyen pour une quantité fixée de données à transmettre.

Par ailleurs, les simulations ont été réalisées avec et sans accusés de réception retardés (Del Ack). Nous ne présentons ici que les résultats obtenus sans l'utilisation de cette option TCP, aucune différence notable n'ayant été observée. En effet, si l'on considère les connexions longues, l'utilisation de Cubic diminue les effets du Del Ack, ce dernier prenant en compte l'utilisation de cette option pour adapter l'évolution de son CA. Cela lui permet de continuer à diminuer le nombre d'ACKs en transit tout en conservant un débit élevé. Dans le cas des connexions courtes, il n'y a toutefois pas de réel changement dans la propagation des bursts.

4.3.2 Performance pour les flux courts

Réseau non congestionné

Dans un réseau non congestionné, l'augmentation de l'IW sans Initial Spreading garantit la meilleure performance. Ainsi, l'émission en continu dès l'établissement de la connexion de la totalité du flux à transmettre est la solution la plus rapide qui soit.

En introduisant l'espacement de l'IW sur le RTT, le temps de délivrance de i segments avec une IW de n ($n \geq i$) n'est plus égal à $RTT + (i-1) * T_B$ qui est le temps de transmission minimal, mais est égal à $RTT + (i-1) * \frac{RTT}{n}$ où n est la taille de l'IW. Utiliser l'Initial Spreading dans un cas sans aucune congestion rallonge donc la durée totale de délivrance de $(i-1) * (\frac{RTT}{n} - T_B)$ secondes avec $(\frac{RTT}{n} - T_B) > 0$.

Ce temps supplémentaire peut varier de quelques millisecondes dans le cas des réseaux terrestres à faible RTT à plusieurs centaines de millisecondes dans le cas des réseaux à très long RTT tels que les satellites. En revanche, il se réduit considérablement dès que l'IW ne suffit plus à l'émission de la totalité du flux. Par exemple, les transmissions de 11 segments avec et sans

Initial Spreading et une IW de 10 segments durent exactement le même temps.

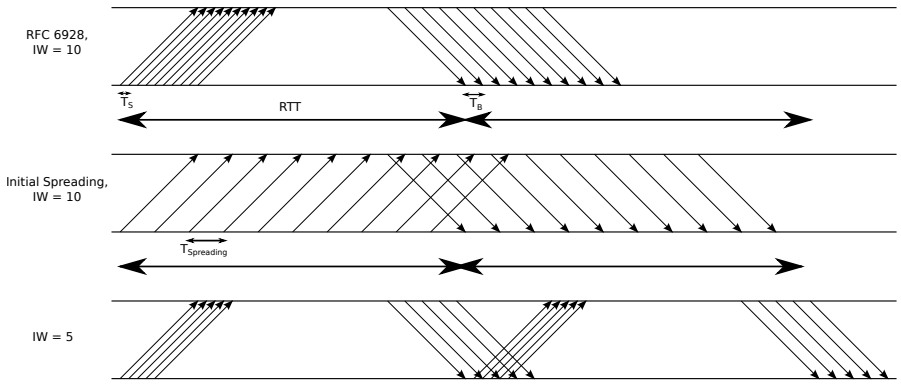

FIGURE 4.7 – Chronogramme comparant la transmission de 10 segments sans congestion pour différents mécanismes

La Figure 4.7 illustre la transmission d'un flux de 10 segments avec une IW de 10 segments avec et sans Initial Spreading. On considère ici le cas où $T_B = 2 * T_S$. L'introduction de l'espacement entre les segments cause une diminution du gain apporté par l'augmentation de l'IW et l'on se rapproche des performances des IW plus petites. Néanmoins, ce phénomène s'estompe à mesure que la taille du flux augmente.

Ainsi, bien que les performances de l'Initial Spreading dans ce cas précis soient en deçà de celles obtenues avec la RFC6928, elles restent néanmoins meilleures que les performances atteintes par les protocoles actuels (RFC3390).

Réseau congestionné

Dans un réseau congestionné, le phénomène de burst affecte de façon importante les performances. Nous avons montré précédemment que le burst induit par l'augmentation de l'IW semble augmenter la probabilité de rejet d'un segment, et réduire l'efficacité de la transmission de la connexion courte. L'utilisation de l'Initial Spreading a pour objectif de lisser le burst en étalant l'émission et ainsi assurer aux segments de l'IW une indépendance des pertes.

Les figures suivantes montrent les résultats des simulations conduites afin de vérifier nos hypothèses. Le goulot d'étranglement est fixé à 10 Mbit/s et le délai à 150ms, les autres liens ont un délai de 5ms et un débit de 40 Mbit/s. La taille du buffer est fixée à la moitié du produit délai-bande passante.

La Figure 4.8 illustre l'évolution de la durée moyenne de délivrance pour une taille de flux donnée avec et sans Initial Spreading et une IW de 10 segments. Ces deux courbes permettent de juger de l'intérêt de l'Initial Spreading.

L'interprétation des courbes peut être séparée en 3 phases distinctes :

– Phase 1 : La taille des flux varie de 1 à 10 segments. Le flux d'une taille inférieure à l'IW est donc envoyé complètement dans le premier RTT. Les deux solutions présentent alors

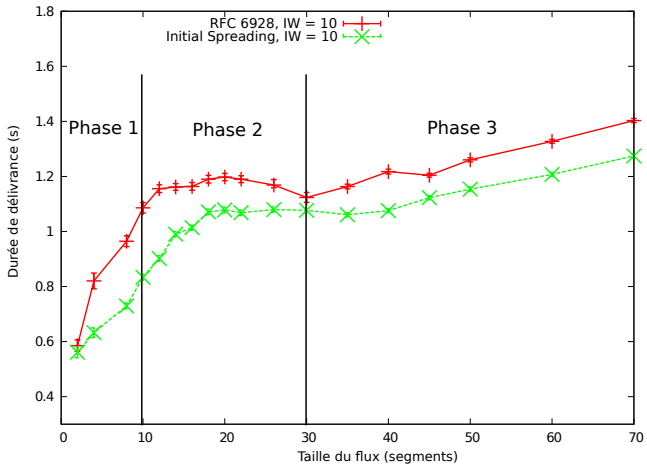

FIGURE 4.8 – Durée de délivrance moyenne pour une IW de 10 segments avec et sans Initial Spreading

des comportements très différents :

- Dans le cas sans Initial Spreading, le flux est envoyé en un unique burst dont la taille varie entre 1 et 10 segments. L'accroissement important du temps de délivrance moyen s'explique par l'augmentation de la probabilité de perte des segments due à l'augmentation de la taille du burst initial. Les segments les plus susceptibles d'être perdus étant les derniers segments du burst, ces pertes ne pourront vraisemblablement pas être récupérées par l'utilisation des mécanismes de reprise et vont nécessiter l'attente d'un RTO et le retour en Slow Start avec une IW égale à 1 segment.
- Avec l'Initial Spreading, le burst initial est lissé et un segment est émis toutes les $\frac{RTT}{10}$ secondes. Le trafic de fond établi par les différentes connexions TCPs étant considéré comme stable, ce temps apparaît suffisant pour décorréler les pertes et pouvoir considérer que la probabilité de rejet et de succès des segments demeure ainsi constante, et plus faible que celle des segments du burst. Ainsi, la perte d'un segment a des chances de pouvoir être récupérée par les mécanismes de reprise car les segments émis après le segment perdu ont une probabilité non nulle d'être correctement transmis et donner lieu à un DupACK. Les résultats sont donc nettement meilleurs avec Initial Spreading que sans.

– Phase 2 : La taille du flux varie de 11 à 30 segments, ce qui correspond au nombre maximal de segments pouvant être transmis dans les 2 premiers RTTs. Une nouvelle fois, les deux solutions conduisent à des performances très différentes :
  - Sans Initial Spreading, la probabilité de pertes des segments appartenant à un burst augmentant en fonction de la place du segment dans le burst, les accusés de réception que l'émetteur reçoit valident probablement les premiers segments envoyés. Ainsi pour chaque

nouvel ACK, l'émetteur, inconscient de potentielles pertes survenues sur les segments suivants, augmente sa CWND et émet deux nouveaux segments. Ces segments sont émis par mini-bursts de 2 segments au rythme du goulot d'étranglement et ont donc une probabilité de succès plus importante que les segments envoyés dans le grand burst Initial. Donc, en augmentant la taille du flux, on augmente le nombre de segments pouvant être émis dès le deuxième RTT et de fait le nombre de potentiel DupACKs. On améliore ainsi la probabilité de pouvoir utiliser des mécanismes de reprise sur erreur et donc de ne pas avoir à subir les conséquences de l'attente d'un RTO.

On peut ainsi constater un phénomène non-intuitif : en moyenne, la transmission d'un flux d'une taille supérieure à un autre peut se terminer plus rapidement si l'augmentation du nombre de segments transmis permet l'utilisation de mécanismes de reprise sur erreur. Ce n'est toutefois pas une solution à privilégier car elle entraîne une surcharge inutile.

- Avec l'Initial Spreading, on observe une situation inverse. En effet, chaque nouvel ACK correctement reçu entraîne également l'émission de 2 segments, mais ces derniers, au lieu d'avoir une meilleure probabilité de succès que les segments émis dans le burst initial, se retrouvent à pâtir de leur émission groupée, et ont par conséquent une probabilité de perte supérieure. Cependant, cet effet est rapidement contrebalancé par l'augmentation du nombre de DupACKs permettant la reprise sur erreur. Lorsqu'un nombre suffisant de segments a été transmis, la probabilité d'entrer en Fast Retransmit et Fast Recovery permet à la durée de délivrance moyenne d'évoluer très lentement.

- Phase 3 : La taille des flux est supérieure à 30 segments. Dans les deux cas, l'entrée probable dans la phase de CA dirige maintenant le débit moyen. Ainsi, les pentes des deux courbes sont sensiblement les mêmes et les deux mécanismes permettent un débit similaire.

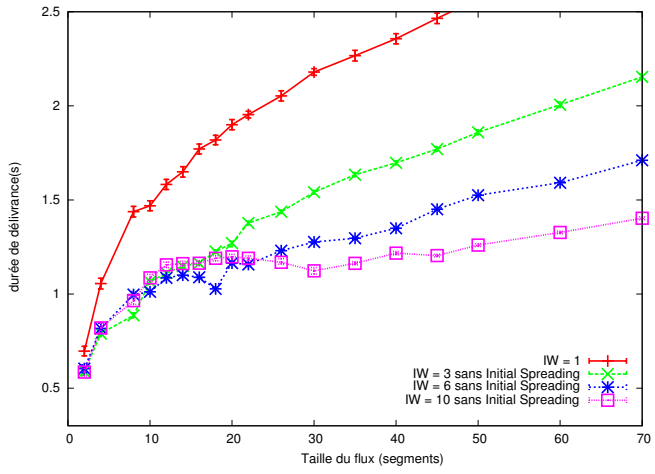

FIGURE 4.9 – Conséquences de l'augmentation de l'IW sans Initial Spreading

La Figure 4.9 et la Figure 4.10 illustrent les effets des différentes tailles d'IW pour des con-

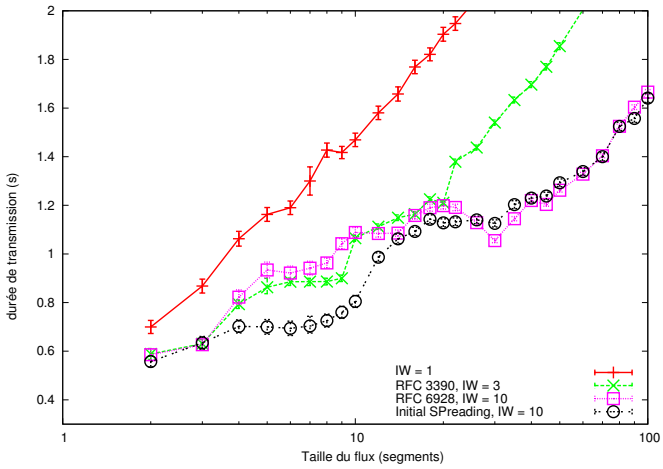

FIGURE 4.10 – Comparaison de la durée moyenne de délivrance pour différentes IW avec et sans Initial Spreading

nexions TCP avec et sans Initial Spreading.

Pour les connexions courtes (autour de 10 segments), on peut constater que l'augmentation de l'IW sans Initial Spreading dégrade les performances moyennes, ce qui valide les explications apportées à la figure précédente. Ainsi, la taille d'IW la plus efficace pour la transmission de 10 segments est égale à 3 dans notre scénario.

En revanche, l'Initial Spreading offre des performances très bonnes pour les grandes valeurs de l'IW. Ainsi on peut constater que l'utilisation de l'Initial Spreading permet une réduction de la durée de délivrance des connexions courtes de plus de 30%.

En conclusion, l'utilisation de l'Initial Spreading associée à une grande IW permet un gain significatif dès lors que le réseau est sujet à congestion.

4.3.3 Performance pour les flux longs

Dans la partie précédente, nous avons mesuré l'intérêt de l'Initial Spreading dans le cas des connexions courtes, ce qui revient à juger des gains relatifs à l'utilisation conjointe du Pacing et d'une grande IW. Nous allons maintenant examiner les effets de l'Initial Spreading sur les connexions longues afin de vérifier que notre mécanisme n'introduit pas les mêmes phénomènes de synchronisation et de surcharge du réseau que le Pacing.

Pour ce faire, nous utilisons un scénario de simulation modifié, similaire à celui de l'article [1]. Ainsi, alors que dans les tests précédents nous étudions la performance d'une connexion dans un environnement déjà congestionné, nous allons maintenant émettre de façon simultanée une quinzaine de flux de très grande taille dans un réseau vide.

La Figure 4.11 illustre les conséquences de l'émission d'un flux avec une grande IW avec et sans Initial Spreading. Les résultats sont normalisés en utilisant la même méthode que dans la partie

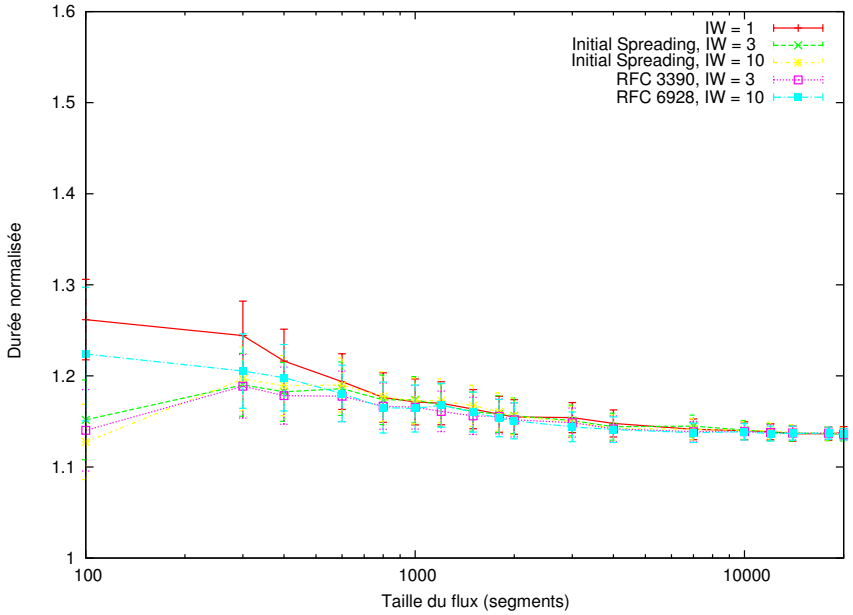

FIGURE 4.11 – Conséquences des effets de la synchronisation sur les flux longs avec et sans Initial Spreading

3.3.2. L'objectif de cette normalisation est de pouvoir comparer les effets des mécanismes sur des flux de tailles très différentes. Les résultats obtenus par simulation sont donc divisés par le temps idéal de délivrance d'un flux de même taille. Cette estimation de la durée de délivrance idéale correspond au temps nécessaire à la transmission d'une certaine quantité de donnée considérant une connexion évoluant en Slow Start avec une IW de 3 segments jusqu'à atteindre une CWND permettant un partage équitable du débit entre les flux, soit dans le cas présent, un débit stable égal à un quinzième du débit disponible. Cette CWND est ensuite conservée jusqu'à la fin de la transmission du flux. Ce temps est considéré comme idéal car on exploite de façon optimale le Slow Start et l'état stable.

Les performances obtenues sont similaires pour les flux longs avec et sans Initial Spreading. Aucun des deux mécanismes ne donne lieu à des phénomènes de synchronisation dont les flux avec Pacing sont victimes.

Alors que l'utilisation du Pacing provoque une surcharge du réseau en différant les congestions jusqu'à un seuil critique, la création de mini-bursts dès le second RTT permet à l'Initial spreading d'augmenter la probabilité de détecter des événements de congestions isolés et donc de pouvoir entrer dans la phase d'évitement de congestion. Cela permet aux différentes connexions une évaluation et un partage précis et dynamique du réseau.

En outre, il est intéressant de noter que lorsqu'un RTO est nécessaire à la détection d'une perte, l'Initial Spreading devient transparent contrairement au Pacing. En effet, après un RTO,

une connexion recommence avec une IW d'un segment, donc l'Initial Spreading n'a plus aucun effet.

4.3.4 Équité de l'Initial Spreading

Comme remarqué dans la partie 3.3.2, les performances du Pacing sont également dégradées par la compétition avec d'autres mécanismes. A partir d'une certaine taille de CWND, l'émission étalée des segments augmente la probabilité que l'un d'entre eux rencontre un burst, et soit donc rejeté.

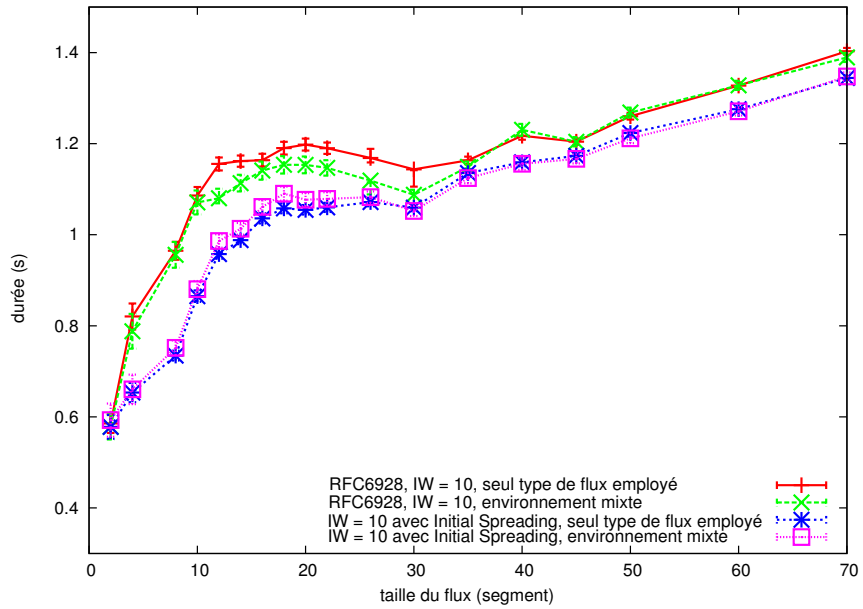

FIGURE 4.12 – constat de l'équité de l'Initial Spreading

La Figure 4.12 montre les performances des différents mécanismes selon l'environnement dans lequel ils évoluent. L'Initial Spreading ne voit pas ses propres performances dégradées par la concurrence des autres types de connexions. Au contraire, il optimise l'utilisation du réseau et permet donc une légère amélioration des performances des autres mécanismes.

4.3.5 Choix de la taille de l'IW

Nous avons jusqu'à présent présenté les bénéfices résultant de l'association de l'Initial Spreading et des IWs des RFC 3390 et RFC 6928 [7, 49]. Notre objectif étant l'optimisation de l'émission des connexions courtes (taille inférieure ou égale à 10 segments), nous avons montré les gains significatifs permis par l'utilisation de l'Initial Spreading avec une IW de 10 segments. Nous allons maintenant juger de la possibilité d'augmenter le nombre de segments initiaux envoyés.

Pour ce faire, nous avons mis en œuvre plusieurs jeux de simulation en faisant notamment varier la charge et le délai du réseau.

	IW=10	IW=12	IW=15
Gain moyen	35.2%	41.5 %	37 %
Gain minimal	30 %	38 %	30 %

TABLEAU 4.1 – Gains atteints par l'utilisation d'Initial Spreading pour l'émission de 10 segments

La Table 4.1 présente les bénéfices moyens et minimaux obtenus en utilisant l'Initial Spreading avec des IW de 10, 12 et 15 segments pour des flux de taille inférieure ou égale à 10 segments dans un environnement congestionné. Le gain de temps est relatif au meilleur résultat obtenu sans Initial Spreading, c'est-à-dire à ce qui a pu être obtenu sans Initial Spreading lorsque l'IW la mieux adaptée au niveau de congestion et à la taille du flux était sélectionnée.

Les résultats de nos tests présentés dans la Table 4.1 montrent un gain minimal significatif de plus de 30 % mais permettent également de noter un point d'inflexion à la signification intéressante. Ainsi les meilleurs résultats sont obtenus avec une IW de 12 segments tandis que les IW de 10 et 15 segments permettent des résultats semblables. Une étude plus approfondie fondée sur l'influence réelle des métriques du scénario testé nous permet d'expliquer cette inflexion en différenciant notamment deux cas de figures :

- Dans le cas des longs délais, l'espace entre 2 segments est suffisamment important pour que l'émission de 15 segments n'ait pas plus de conséquence que l'émission de 10. Les segments conservent une semblable "indépendance", et donc une même probabilité de perte.
  Ainsi, en utilisant une IW de 15 segments au lieu de 10, on ne fait que réduire le temps minimal de transmission dans le cas sans perte en espaçant les segments de $\frac{RTT}{15}$ au lieu de $\frac{RTT}{10}$. Ce comportement a été notamment vérifié dans nos tests sur les liens satellites comme nous le reverrons dans le chapitre 6.
- Dans le cas des délais plus faibles, l'augmentation du nombre de segments émis, induisant la réduction de l'espacement entre deux segments, est responsable de la corrélation des pertes des segments et donc de l'augmentation du taux de perte.
  Par exemple, dans le cas des tests avec un délai de quelques millisecondes, une IW de 12 segments a montré une meilleure efficacité qu'une IW de 15 segments.

Considérant les flux de taille inférieure ou égale à 10 segments, le choix de la taille de l'IW s'apparente donc à un compromis à trouver entre le gain de temps obtenu par la réduction de l'étalement dans le cas d'une transmission sans perte, et la potentielle dégradation due à un espacement trop faible. L'Initial Spreading ciblant principalement les connexions courtes, il nous semble préférable de privilégier l'option conservative et donc de retenir l'IW validée par la RFC 6928.

Toutefois, cette marge de manœuvre dans le choix de l'IW est un atout de plus au crédit de l'Initial Spreading. En effet, si la tendance de l'augmentation de la taille moyenne des objets WEBs se confirme, l'Initial Spreading, contrairement aux autres mécanismes, pourra alors être

utilisé avec une IW encore supérieure et conserver une efficacité remarquable.

4.4 Conclusion de ces premières simulations

Les simulations réalisées ont donc permis d'évaluer l'intérêt de l'Initial Spreading sur les performances de TCP. Sa mise en oeuvre permet d'augmenter la taille de l'IW quel que soit l'état du réseau. Nos simulations ont notamment montré que l'utilisation de l'Initial Spreading permet de conserver un niveau de performances supérieur à celui de la RFC 3390 dans les réseaux non congestionnés, tout en offrant des gains substantiels dans les cas congestionnés.

Plus particulièrement, la confrontation de l'Initial Spreading avec les mécanismes classiques associant Slow Start et une taille quelconque d'IW permet de constater dans le cas des réseaux congestionnés :

- un gain de temps supérieur à 30 % pour la transmission de connexions courtes ;
- un débit au moins aussi bon pour les connexions longues ;
- l'absence avec l'Initial Spreading des défauts du Pacing (surcharge du réseau et synchronisation)

En conclusion, bien que dans le cas des transmissions sans congestion, nous restions en deçà des performances obtenues par la RFC 6928, l'utilisation de l'Initial Spreading semble en mesure d'améliorer les connexions courtes et donc les performances de TCP. Sa simplicité en fait par ailleurs un mécanisme aisément intégrable dans les piles modernes et laisse espérer que l'on puisse à terme se passer des TCP-PEPs.

Pour autant, même si l'utilisation de la simulation est un passage obligatoire dans le cycle de développement d'un protocole, dans le cas de TCP, l'interaction entre les flux est tellement complexe qu'il serait insuffisant de s'en contenter. De plus, lors de nos simulations, nous n'avons pas tenu compte des interactions nombreuses avec les autres couches de la pile protocolaire. Ces dernières, sensiblement différentes selon la technologie utilisée, peuvent avoir un impact important sur notre mécanisme et nécessitent de plus amples analyses, notamment au travers d'expériences réelles.

Dans la partie suivante, nous proposons un modèle mathématique qui nous permet de valider nos hypothèses sur la compréhension des bursts et donc de valider l'intérêt de notre mécanisme dans les réseaux congestionnés.

5 Validation analytique de l'Initial Spreading

5.1 Introduction

Le chapitre précédent nous a permis de constater empiriquement l'intérêt de l'Initial Spreading. Les résultats des nombreuses simulations réalisées sur un large panel de scénarios montrent en effet un gain de performance en milieu congestionné d'au moins 30% dans la transmission des connexions courtes, c'est-à-dire des connexions de taille inférieure ou égale à 10 segments. Ces résultats ont ainsi corroboré nos hypothèses sur les conséquences des bursts de segments sur les performances individuelles des connexions dans un réseau congestionné.

Pour autant, lorsqu'il s'agit d'étudier les apports d'un mécanisme sur la performance d'un ensemble de protocoles aussi complexe que TCP, les résultats des simulations ne peuvent et ne doivent être considérés qu'à titre indicatif afin de valider la ligne directrice. C'est pourquoi de nombreux modèles analytiques ont été proposés ces dernières années [55, 56] afin de caractériser les performances de TCP dans son état stable mais également en Slow Start. Peu de ces modèles ont toutefois pour objectif de décrire précisément la transmission des flux courts [57, 58] et à notre connaissance, aucun ne permet une analyse fine de la propagation des bursts [59, 60] pour évaluer les performances de TCP dans ce dernier cas de figure.

Dans la suite de ce chapitre, nous proposons un modèle analytique axé sur la transmission de flux TCP courts dans un environnement congestionné. Celui-ci s'appuie sur l'étude de la propagation des bursts et de leurs conséquences pour décrire et tenir compte de façon exhaustive de l'ensemble des scénarios possibles pouvant amener à la transmission d'une faible quantité de données. Notre objectif est ainsi de modéliser la durée moyenne de délivrance d'une connexion courte en ayant connaissance uniquement du RTT, du taux de perte moyen par segment dans le réseau et du mécanisme de démarrage rapide de TCP employé pour la transmission du flux.

5.2 Étude des bursts

Nous avons pu voir dans les chapitres précédents qu'il est communément admis que les bursts ont un impact significatif sur la performance d'une connexion. Pour autant, bien qu'elles soient

souvent redoutées, les conséquences réelles du burst initial, engendré par l'IW, sur la transmission d'un flux court demeurent relativement méconnues.

5.2.1 Définition des bursts

Dans le paragraphe 3.2 et le chapitre précédent, nous avons utilisé la notion de burst et montré les conséquences théoriques de deux différentes sortes de burst sur la file d'attente du routeur sans pour autant en donner une définition précise. Afin de modéliser finement leurs impacts sur la transmission d'un flux court, nous ne pouvons toutefois plus nous contenter de définir un burst comme une succession de segments.

Dans la littérature, la définition de burst est généralement reliée à celle de cycle (ou "round"). Un cycle commence avec la transmission du premier segment d'une CWND et se termine par la réception de son ACK. Les segments envoyés durant un même cycle forment alors un burst [59, 61].

Cette définition est parfaitement adaptée à la modélisation de TCP dans son état stable où l'échelle temporelle plus importante se prête à une caractérisation cycle par cycle de l'état de TCP. Néanmoins, elle ne sied pas à une modélisation précise de la transmission d'un flux court, et notamment à la prise en compte des différents mécanismes de démarrage rapide tels que l'Initial Spreading ou l'augmentation de l'IW. En effet, cette définition ne permet pas en l'état de distinguer les deux mécanismes du point de vue de la transmission des bursts et donc de justifier par le modèle de la différence de performance constatée.

Pour la suite de ce manuscrit, nous allons donc considérer que l'appartenance de deux segments à un même burst est associée à l'impact que leur émission successive peut avoir sur le réseau et non plus à des considérations de cycle. Nous faisons ainsi l'hypothèse que deux segments ne sont pas considérés comme appartenant à un même burst si l'état du buffer du routeur du goulot d'étranglement a eu la possibilité d'évoluer entre leurs réceptions successives, et ce, quel que soit leur cycle d'émission. Cela revient à considérer que deux segments d'une même connexion arrivant dans des états de buffer "indépendants" auront leurs pertes considérées comme non "corrélées". Par analogie, dès lors que la probabilité de perte d'un segment est modifiée par le précédent segment du flux, les deux segments sont considérés comme appartenant à un même burst.

Cette hypothèse étant difficile à vérifier mathématiquement mais aussi expérimentalement, nous adopterons dans la suite du travail la définition suivante : deux segments consécutifs d'une même connexion TCP seront considérés comme faisant partie du même burst si l'écart entre leur temps d'émission au niveau TCP est inférieur au temps d'émission des paquets IP correspondant au débit du lien le plus faible qu'ils traversent. C'est donc une définition théorique puisque de nombreux phénomènes vont se produire alors de type multiplexage, attentes aléatoires ... jusqu'à ce qu'ils parviennent dans le buffer du routeur considéré.

Cette définition permet notamment de différencier l'émission de l'IW avec et sans Initial Spreading et de prendre en compte les deux types de bursts dont nous avons parlé dans les

chapitres précédents :

- le burst initial dû à l'émission de l'IW ;
- les mini-bursts induits par l'augmentation de la CWND lorsqu'elle peut se produire.

5.2.2 Étude empirique des conséquences des bursts sur un réseau congestionné

Afin d'évaluer les conséquences de l'émission d'un burst initial dans un réseau congestionné, nous avons réalisé une série d'expériences sur des machines Linux en utilisant à nouveau une topologie Dumbbell semblable à la Figure 4.1. Le débit du goulot d'étranglement est fixé à 10 Mbit/s tandis que celui des autres liens est égal à 100 Mbit/s. Netem, un outil de contrôle du trafic permettant d'émuler certaines caractéristiques d'un lien, est utilisé pour introduire un délai de 50ms dans le lien du goulot d'étranglement. 8 connexions de durée infinie sont établies en parallèle pour engendrer la congestion et créer un taux de perte moyen par segment de 5% et un taux d'occupation moyen du buffer de 85%. Finalement, nous avons transmis des bursts de taille variable et observé l'influence que leur taille peut avoir sur leur performance, notamment en terme de durée moyenne de délivrance et de probabilité de perte. 100 000 itérations ont été faites pour chaque taille de flux afin de garantir la pertinence des résultats et des intervalles de confiance suffisamment faibles d'une valeur relative inférieure à 5% pour un niveau de confiance de 95%.

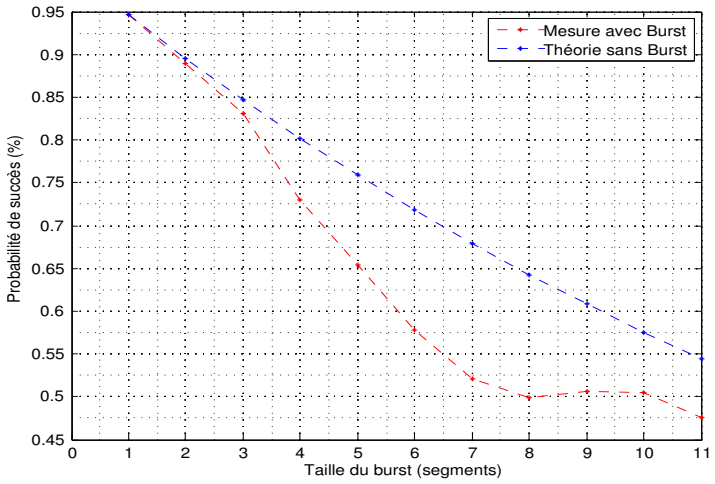

FIGURE 5.1 – Probabilité de transmettre l'intégralité de l'IW en fonction de sa taille avec et sans burst

La Figure 5.1 trace la probabilité de transmettre sans aucune perte la totalité des segments émis en un unique burst en fonction de sa taille. Ce résultat expérimental est alors comparé à un modèle théorique pour lequel les pertes seraient vraiment indépendantes avec un taux de perte

de 5%. Cette courbe théorique vaut donc simplement $1-(1-p)^b$ où b est la taille du burst et p le taux de perte.

Cette figure nous permet de constater que l'émission des segments en bursts a un impact significatif sur la performance moyenne, ce dernier croissant d'ailleurs avec la taille du burst :

- Pour les flux de moins de 4 segments, les deux courbes présentent des comportements similaires. L'émission en burst n'ajoute donc que peu de corrélation entre les pertes.
- Pour les flux plus longs, la différence entre les deux courbes est très significative, l'émission en burst réduisant la probabilité de transmettre correctement l'ensemble des segments d'un même burst. Ainsi, l'hypothèse d'indépendance des pertes n'est plus vérifiée et la probabilité qu'un segment soit perdu sachant que les y segments précédents du burst auquel il appartient ont été correctement traités par le routeur du goulot d'étranglement augmente sensiblement avec y.

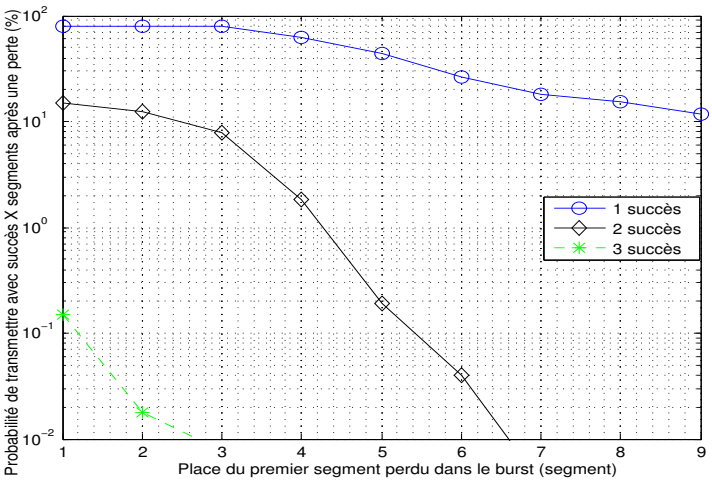

FIGURE 5.2 – Probabilité d'avoir X des segments du burst qui font suite au premier segment perdu correctement transmis ($X \in 1, 2, 3$) en fonction de la place de la perte

Par ailleurs, la Figure 5.2 nous permet d'approfondir notre compréhension des conséquences des bursts dans un milieu congestionné en illustrant la corrélation entre les pertes des segments appartenant à un même burst. Pour ce faire, nous considérons uniquement les bursts de 10 segments émis à travers le réseau ayant souffert d'au moins une perte, et nous intéressons plus spécifiquement aux segments du burst qui font suite au premier segment perdu. Par exemple, si le troisième segment d'un burst est perdu, nous analysons le comportement des segments 4 à 10 (comme illustré par la Figure 5.3).

La Figure 5.2 représente alors la probabilité mesurée d'avoir 1, 2, ou 3 de ces segments "restants" correctement transmis en fonction de la place du premier segment perdu dans le burst.

Deux résultats principaux méritent d'être soulignés :

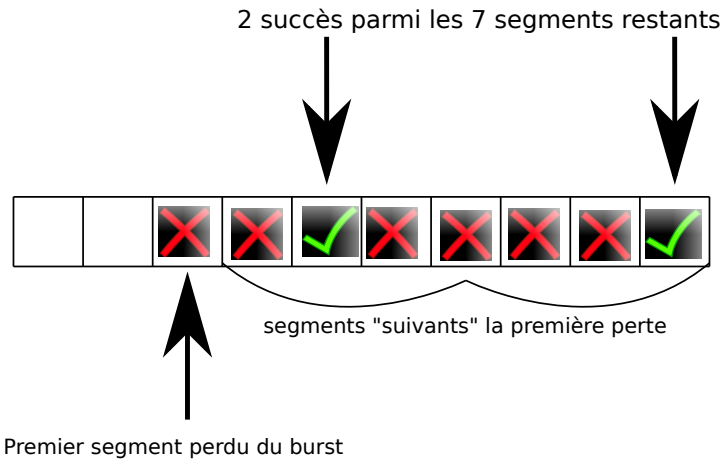

FIGURE 5.3 – Cas où 2 des 7 segments “suivants” sont correctement transmis

- la perte d’un des segments du burst ne signifie pas forcement la perte de tous les segments “restants” ;
- quelle que soit la position du premier segment perdu dans un burst de 10 segments, la probabilité d’avoir 3 ou plus des segments “restants” correctement reçus est quasiment nulle.

Ce dernier point est essentiel car il montre l’inefficacité des mécanismes de reprise classiques lorsqu’une grande IW est utilisée pour la transmission d’un flux court. En effet, en augmentant l’IW de sorte à pouvoir émettre la totalité du flux court en un burst unique, on augmente non seulement le risque de subir une perte d’un des segments, mais on réduit également de façon drastique la possibilité de récupérer cette perte via les mécanismes de reprise classiques qui nécessitent la réception de 3 DupACKs.

En conclusion, l’utilisation de bursts dans la transmission de flux courts dégrade fortement les performances en corrélant les pertes de segments, augmentant la probabilité de perte et en réduisant l’efficacité des mécanismes de reprise.

5.2.3 Modélisation des bursts

Certaines modélisations de TCP attentives aux conséquences des bursts sur la performance moyenne des connexions adoptent un modèle de burst [59, 61] appelé “bursty loss model”. Ce modèle est régulièrement utilisé pour modéliser un burst dans un réseau dont les routeurs ont une politique DropTail de file d’attente. Ce modèle fait l’hypothèse que tous les segments appartenant à un même cycle ont la même probabilité d’être perdus tant qu’aucun des segments du cycle n’est perdu, et ce, indépendemment de toute autre considération, telle que la taille de la CWND, la position du segment dans le cycle, ou encore l’influence des autres cycles. En revanche, dès lors

qu'un des segments du cycle est perdu, tous les segments suivants du cycle le sont également.

Le paragraphe 5.2.2 nous a permis de constater les faiblesses d'une telle hypothèse. Ainsi, la Figure 5.1 a montré que la perte d'un segment du burst n'implique pas la perte de tous les segments suivants, tandis que la Figure 5.2 illustre quant à elle la forte corrélation entre la place du segment dans le burst et sa probabilité de perte.

La définition du burst que nous donnons dans le paragraphe 5.2.1, qui nous permet de considé|rer l'"indépendance" de plusieurs bursts appartenant à un même cycle en différenciant les deux notions, réduit l'imprécision d'une telle modélisation en garantissant la corrélation des pertes des segments appartenant à un même burst.

Dans la suite du manuscrit, nous utilisons malgré tout la modélisation précédente des bursts à la nuance près que celle-ci n'est pas appliquée à l'ensemble du cycle mais à un burst tel que nous l'avons défini dans le paragraphe 5.2.1.

En conservant ce modèle de burst, nous sommes conscients que l'imprécision du modèle TCP que nous proposons va augmenter avec la taille des bursts. Notre objectif principal étant la validation des bénéfices de l'Initial Spreading sur les flux courts, il est essentiel que notre modèle soit précis dans la modélisation de l'émission des segments indépendants et des mini-bursts (burst de 2 segments). Nos expériences ayant montré que la perte du premier segment d'un mini-burst était généralement suivie de celle du second segment, nous faisons l'hypothèse que ce modèle de burst simple répond à nos attentes en nous permettant d'une part une évaluation analytique fine de l'Initial Spreading, et d'autre part une modélisation suffisamment précise des bursts de taille plus importante pour que l'on puisse comparer les performances des différents mécanismes de démarrage rapide de TCP.

5.3 Modèle analytique pour les connexions courtes TCP

L'objectif de notre modèle est d'estimer la durée de délivrance moyenne de i segments émis avec une IW de n segments [4]. Nous nous intéressons tout particulièrement aux conséquences des bursts sur la transmission de flux courts en réseau congestionné afin de pouvoir notamment valider les résultats expérimentaux obtenus au chapitre précédent.

5.3.1 Hypothèses

Cette étude cible la modélisation des connexions courtes. Cela a de nombreuses répercussions sur nos hypothèses.

Nous faisons tout d'abord l'hypothèse que tous les segments d'un même flux traversent le réseau en parcourant le même chemin. Nous pouvons ainsi considérer que le taux de perte des segments dits "indépendants" est le même, c'est-à-dire que deux segments n'appartenant à aucun burst ont la même probabilité d'être perdus. De même, le RTT moyen pour la transmission d'un segment seul est supposé fixe et connu.

Considérant la faible taille de la connexion, nous faisons également l'hypothèse que le débit est régulé seulement par l'émetteur via la gestion de la CWND et non par le récepteur au travers de l'AWND.

D'autre part, dès lors qu'ils peuvent être utilisés, nous considérons que le Fast Retransmit et le Fast Recovery associés au mécanisme SACK permettent de récupérer un faible nombre de segments perdus. Ainsi, la réception de 3 DupACKs entraîne-t-elle la ré-émission du premier segment perdu et chacun des nouveaux accusés de réception reçus permet ensuite la ré-émission d'un des segments perdus. En revanche, si le nombre de DupACKs ne permet pas d'utiliser les mécanismes de reprise, la connexion entre à nouveau en Slow Start avec une IW d'1 segment après expiration du RTO.

Cette hypothèse nous permet de nous affranchir de la modélisation des différents algorithmes de congestion de TCP, cet autre problème ayant été traité dans de nombreux travaux [55, 56].

La RFC 6298 [40] est utilisée pour définir le RTO utilisé. Ainsi la durée du RTO est initialement fixée à 1 seconde, puis est égale au maximum entre 1 seconde et une valeur calculée à partir des mesures du RTT. Nous faisons l'hypothèse que la durée du RTO demeure égale à 1 seconde dans les scénarios que nous considérons. De plus, nous respectons l'implantation du RTO préconisée par la RFC 6298 [40] et utilisons un seul timer de retransmission au niveau de l'émetteur qui est remis à zéro à chaque fois qu'un ACK valide un nouveau segment, c'est-à-dire, pour chaque accusé de réception qui n'est pas un DupACK.

Par ailleurs, la différence de débit entre le goulot d'étranglement et le reste du réseau a des conséquences importantes sur l'analyse de performance. Ainsi, comme nous l'avons vu précédemment, nous considérons qu'i segments appartenant au même cycle ne peuvent être émis que de 3 façons différentes :

- en un burst de i segments dans le cas de l'émission d'une IW supérieure ou égale à i sans Initial Spreading ;
- de façon indépendante (c'est-à-dire qu'aucun des i segments n'appartient à un burst) si l'IW supérieure ou égale à i est émise avec Initial Spreading ;
- par mini-bursts de deux segments dans les cas de figure où la réception d'un ACK dans la phase de Slow Start engendre l'émission de deux nouveaux segments.

La Figure 5.4 montre les trois différents types de bursts considérés au travers de l'émission de 10 segments avec et sans Initial Spreading et une IW de 5 segments.

Dans la suite, nous tenons compte des bursts en utilisant le modèle "bursty loss model" modifié tel que nous l'avons présenté en 5.2.3.

Finalement, nous faisons l'hypothèse que les différents temps d'émission sont négligeables par rapport aux temps de propagation. Ainsi, dans la suite du modèle, T_S et T_B (voir Figure 4.3) sont considérés comme négligeables, et nous ne tenons compte que du RTT et de $T_{Spreading}$ qui est égal à $\frac{RTT}{n}$ où n est la taille de l'IW.

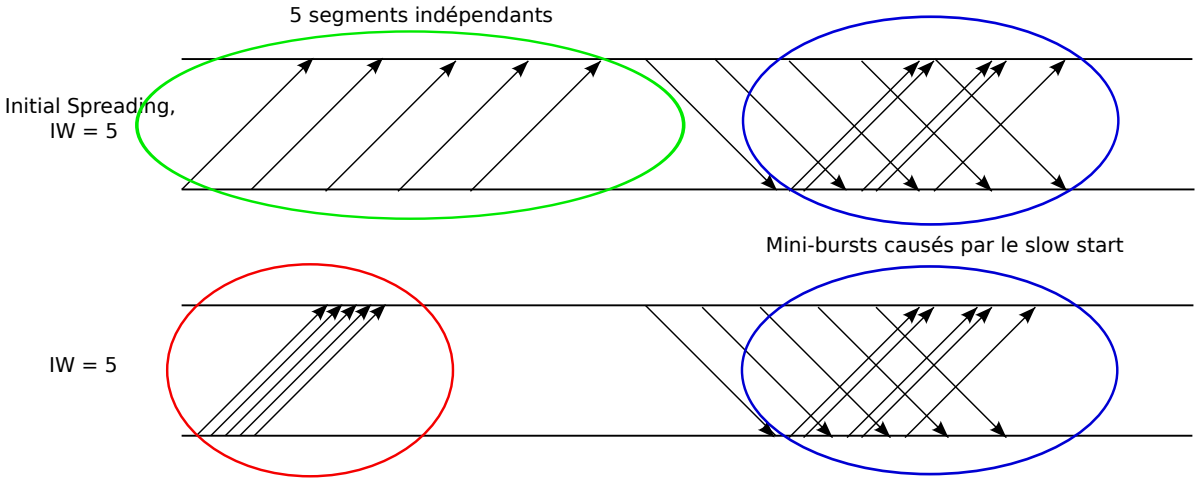

FIGURE 5.4 – Illustration des différents bursts considérés par le modèle

5.3.2 Description du modèle

Afin d'évaluer les performances de la connexion TCP dans les transmissions de flux courts, on peut en modéliser l'évolution par un automate à états finis temporisé stochastique dont l'objectif est de décrire la transmission d'une connexion à travers l'évolution de la fenêtre de l'émetteur (CWND ou IW), le nombre de segments restant à transmettre et la façon avec laquelle ils seront émis (indépendamment, par mini-bursts ou en un seul large burst initial).

Chaque état correspond ainsi à la délivrance d'un certain nombre de segments avec une fenêtre donnée (que ce soit l'IW ou la CWND) et un type de gestion des bursts particulier. Seules deux transitions peuvent permettre un changement d'état :

- la réception de l'ensemble des accusés de réception de la fenêtre de segments émise ;
- la détection d'une erreur par l'émetteur.

Suite à chaque transition, un nouvel état est atteint qui correspond à la délivrance des segments qui doivent encore être transmis avec une CWND et un type de gestion des bursts actualisé. Les transitions seront aléatoires et l'on déterminera les probabilités de ces transitions en fonction des événements auxquels elles correspondent. Le temps de franchissement de la transition sera en revanche déterministe étant données les hypothèses proposées dans le paragraphe précédent.

Cet automate permet une reproduction fidèle du comportement d'une entité TCP qui fait effectivement évoluer sa capacité d'émission au gré des accusés de réception et des détections d'événements de congestion. Nous allons ainsi être en mesure de suivre pas à pas l'émission d'une connexion courte, de l'émission de l'IW avec et sans Initial Spreading jusqu'à la réception du dernier ACK validant la transmission réussie de l'ensemble de la connexion.

La Figure 5.5 montre la correspondance entre notre modèle et les premières étapes d'un scénario de transmission réelle des segments représentant la transmission d'une connexion de 10

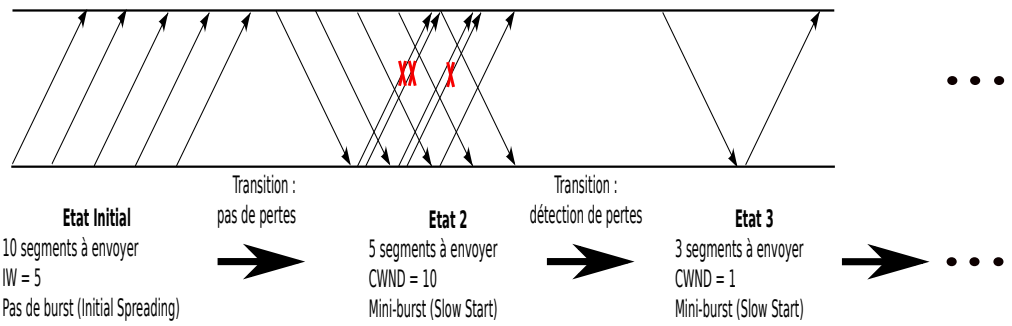

FIGURE 5.5 – Correspondance entre le modèle et un scénario de transmission réelle des segments

segments avec une IW de 10 segments et l'utilisation de l'Initial Spreading. Ce dernier point influe sur la façon dont les segments initiaux sont émis.

Nous avons donc défini deux états initiaux qui nous permettent de différencier les mécanismes de démarrage rapide :

- D_i^n : i segments sont délivrés avec une IW de taille n. L'Initial Spreading n'est pas utilisé : l'IW est donc transmise en un unique burst.
- S_i^n : i segments sont délivrés avec une IW de taille n. L'Initial Spreading est utilisé : l'IW est donc transmise sans aucun burst.

Dès lors que la connexion sort de l'état initial, la différence de débit entre le goulot d'étranglement et les autres liens va entraîner un changement dans la façon dont les segments sont envoyés, et ce, quel que soit le mécanisme de démarrage rapide de TCP employé (voir Figure 5.4). Dans les parties précédentes, nous avons fait l'hypothèse que ces mini-bursts ont un impact important sur la performance des connexions courtes ; ils nécessitent donc d'être modélisés précisément.

Ainsi, les états intermédiaires seront modélisés de la façon suivante :

- B_i^n : i segments sont délivrés avec une CWND de taille n. La CWND est transmise en $\lfloor \frac{n+1}{2} \rfloor$ bursts de 2 segments.

L'état final est atteint lorsque l'ensemble des segments a été délivré.

Un couple de valeur est associé à chaque transition : la probabilité associée au passage d'un état à un autre et le temps requis.

Finalement, nous calculons $\bar{T}(D_i^n)$, $\bar{T}(S_i^n)$ et $\bar{T}(B_i^n)$, les durées moyennes de délivrance des flux de i segments avec une taille de fenêtre de n segments et une gestion particulière des bursts. Ce temps correspond à la durée moyenne pour passer de l'état courant à l'état final.

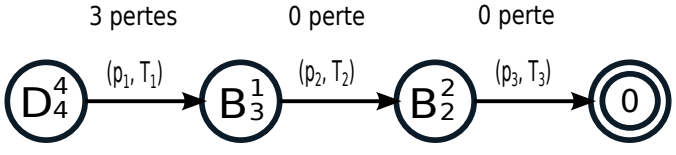

FIGURE 5.6 – Un des scénarios possibles pour transmettre D_4^4

Par exemple, la Figure 5.6 montre une sous-partie de l'automate correspondant à un des scénarios possibles pour transmettre D_4^4, c'est-à-dire, pour transmettre 4 segments sans Initial Spreading et avec une IW de 4 segments. La première transition est l'expiration du RTO, due à la perte de 3 des 4 segments. Les 3 segments restants ont donc à être transmis en Slow Start avec une IW d'un segment, d'où l'entrée dans l'état B_3^1. La seconde transition est la réception de l'accusé du segment émis. Ainsi, le nouvel état correspond à la délivrance de 2 segments avec une CWND actualisée égale à 2 segments (B_2^2). Comme aucun des deux segments transmis dans le cycle n'est perdu, la réception des deux ACKs permet de basculer dans l'état final. Pour ce scénario, la durée de délivrance est donc égale à $T_1 + T_2 + T_3$. Considérant nos hypothèses, les quantités T_i seront considérées comme constantes.

La Table 5.1 récapitule les différents états et les variables utilisés pour décrire le modèle :

Variables ou états	Définitions
i	Nombre de segments à transmettre
n	Taille de la fenêtre (IW ou CWND)
D_i^n	Etat définissant la transmission de i segments avec une IW de n segments sans Initial Spreading
$\bar{T}(D_i^n)$	Durée moyenne de délivrance de D_i^n
S_i^n	Etat définissant la transmission de i segments avec une IW de n segments avec Initial Spreading
$\bar{T}(S_i^n)$	Durée moyenne de délivrance de S_i^n
B_i^n	Etat définissant la transmission de i segments avec une CWND de n segments et des mini-bursts
$\bar{T}(B_i^n)$	Durée moyenne de délivrance de B_i^n
R	le RTT moyen supposé constant
T_0	le Timer de Retransmission
p	la probabilité de perte pour un segment n'appartenant pas à un burst
$q = 1 - p$	la probabilité de succès pour un segment n'appartenant pas à un burst

TABLEAU 5.1 – Tableau récapitulatif des états et paramètres utilisés dans notre modèle

5.3.3 Initialisation

Par définition, nous pouvons écrire que :

$$\forall i, \begin{cases} D_i^1 = S_i^1 \\ \bar{T}(D_i^1) = \bar{T}(S_i^1) \end{cases}$$

car l'Initial Spreading est transparent lorsqu'il est utilisé avec une IW unitaire.

Par ailleurs, le Slow Start est utilisé avec ou sans Initial Spreading, donc à chaque accusé de réception autre qu'un DupACK, un burst de 2 segments est envoyé. Dans le cas d'une IW

unitaire, la réception du premier ACK entraîne l'émission de deux nouveaux segments. Cela se traduit par :

$$\forall i, \quad \bar{T}\left(D_i^1\right) = \bar{T}\left(D_1^1\right) + \bar{T}\left(B_{i-1}^2\right)$$

Considérant que T_0 est doublé quand une perte n'a pas été récupérée à l'expiration du T_0 enclenché à la ré-émission précédente, la durée moyenne pour transmettre un segment en fonction de p, R et T_0 est égale à :

$$\bar{T}\left(D_1^1\right) = R + q\sum_{i=1}^{\infty} p^i \sum_{j=1}^{i} 2^{j-1} T_0 = R + T_0 \frac{p}{1-2p}$$

Nous connaissons désormais la durée moyenne nécessaire à la transmission d'une connexion ne contenant qu'un segment. Nous allons maintenant détailler les différents scénarios qui peuvent permettre l'émission de 2 segments avec une IW égale à 2. La ré-émission d'un segment étant nécessaire dans les scénarios incluant une ou plusieurs pertes, nous utiliserons alors notre calcul précédent de $\bar{T}\left(D_i^1\right)$.

Émission de D_2^2

La Figure 5.7 montre les cas possibles lorsque l'Initial Spreading n'est pas utilisé et les différences que cela engendre au niveau des temps de retransmission.

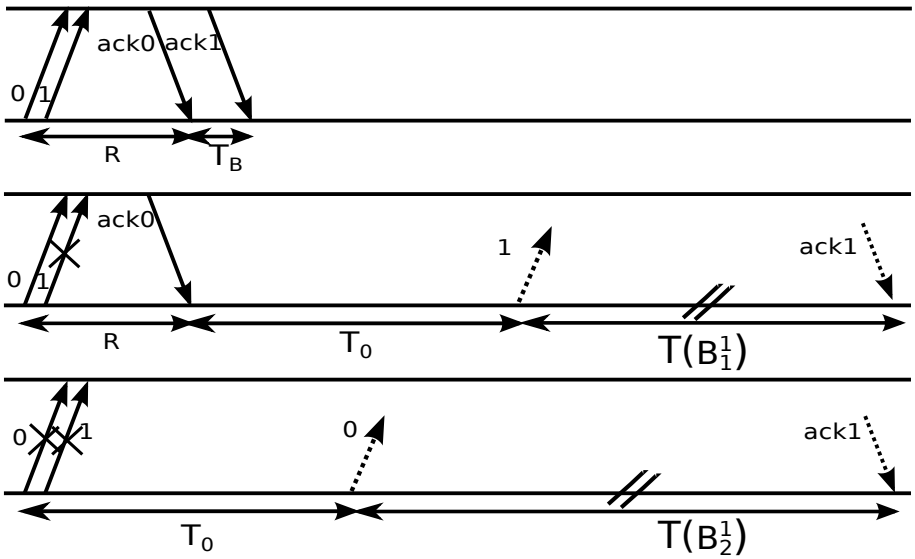

FIGURE 5.7 – $\bar{T}\left(D_2^2\right)$, les différents scénarios possibles

Ainsi, lors de la perte du second segment, la retransmission du segment perdu ne peut se

faire qu'à $R+T_0$, alors que la perte du premier segment, qui implique la perte du burst complet, nécessite une retransmission des deux segments à T_0. Dans le premier cas, T_0 a été remis à zéro par la réception du premier ACK.

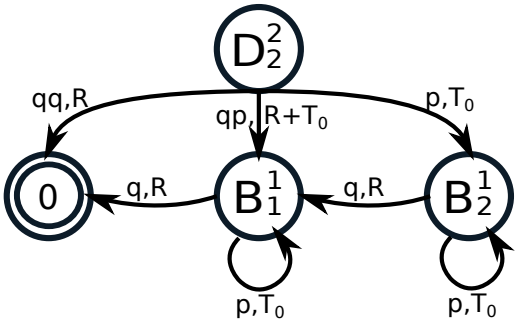

FIGURE 5.8 – Diagramme d'états pour D_2^2

La Figure 5.8 donne le diagramme d'états pour D_2^2.

D'après le diagramme d'états et la représentation des cas possibles, nous pouvons déduire $\bar{T}\left(D_2^2\right)$ et $\bar{T}\left(B_2^2\right)$, les durées moyennes de délivrance de 2 segments avec un IW de 2 segments sans Initial Spreading et dans les états intermédiaires :

$$\bar{T}\left(D_2^2\right) = q^2R + qp(R + T_0 + \bar{T}\left(B_1^1\right)) + p(T_0 + \bar{T}\left(B_2^1\right))$$
$$\bar{T}\left(B_2^2\right) = \bar{T}\left(D_2^2\right)$$

Émission de S_2^2

En utilisant Initial Spreading, on espace les segments et on va considérer des pertes indépendantes. Ainsi, le second segment du flux peut être transmis avec succès alors que le premier segment a été perdu. La Figure 5.9 montre les 4 différents scénarios possibles pour S_2^2.

On peut ainsi en déduire $\bar{T}\left(S_2^2\right)$:

$$\bar{T}\left(S_2^2\right) = q^2(R + \frac{R}{2}) + qp(R + T_0 + \bar{T}\left(B_1^1\right)) + pq(T_0 + \bar{T}\left(B_1^1\right)) + p^2(T_0 + \bar{T}\left(B_2^1\right))$$

Nous allons maintenant généraliser notre modèle pour $n > 2$ et résoudre séparément le cas des différents mécanismes de démarrage rapide de TCP.

5.3.4 Généralisation : $\forall n \geq i \geq 2$

L'intérêt principal de l'augmentation de l'IW avec ou sans Initial Spreading réside dans la transmission des flux courts, et notamment des flux dont la taille est inférieure ou égale à l'IW. Dans cette partie, nous allons décrire notre modèle pour $n \geq i$.

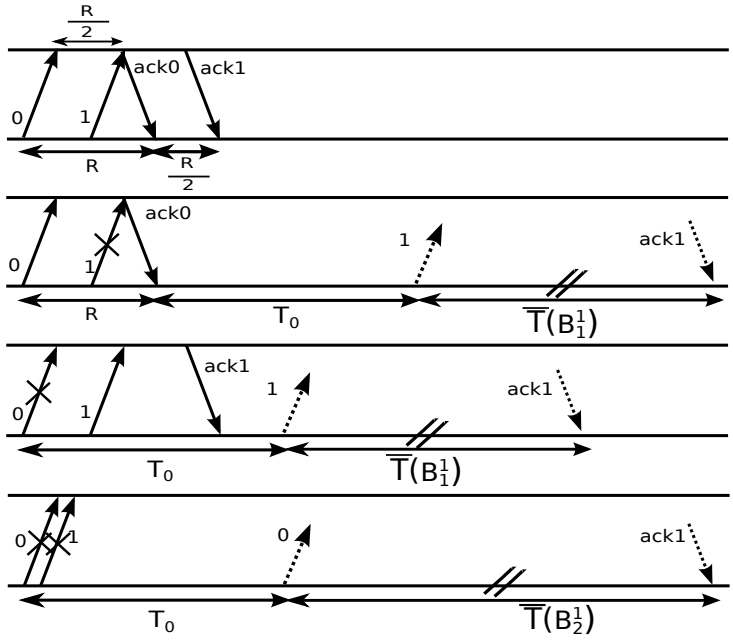

FIGURE 5.9 – $\bar{T}\left(S_2^2\right)$, les différents scénarios possibles

Cas des grandes IW sans Initial Spreading

Nous considérons maintenant le cas D_i^n ; les i segments sont envoyés en un unique burst que nous modélisons par le "bursty loss model".

Nous pouvons tout d'abord noter que :

$$\forall n > i, \quad \bar{T}\left(D_i^n\right) = \bar{T}\left(D_i^i\right)$$

Suite à la première transition, l'état initial peut être décliné en $i+1$ états différents représentants les i différents segments pouvant être affectés par la première perte et le cas sans perte.

Ainsi, en utilisant un raisonnement similaire à celui employé pour D_2^2, on peut écrire :

$$\bar{T}\left(D_n^n\right) = q^n R + \sum_{i=1}^{n-1} q^{n-i} p(R + T_0 + \bar{T}\left(B_i^1\right)) + p(T_0 + \bar{T}\left(B_n^1\right))$$

Cas des grandes IW avec Initial Spreading : cas de S_n^n

L'indépendance des pertes que nous supposons avec l'utilisation de l'Initial Spreading rend la résolution du cas de S_n^n plus compliquée et augmente notablement le nombre de scénarios à considérer. S_n^n peut ainsi être décliné en 2^n différents états selon le nombre de pertes et leurs positions dans le burst.

Comme nous avons pu le voir dans le cas de D_2^2, la position de la première perte d'un segment

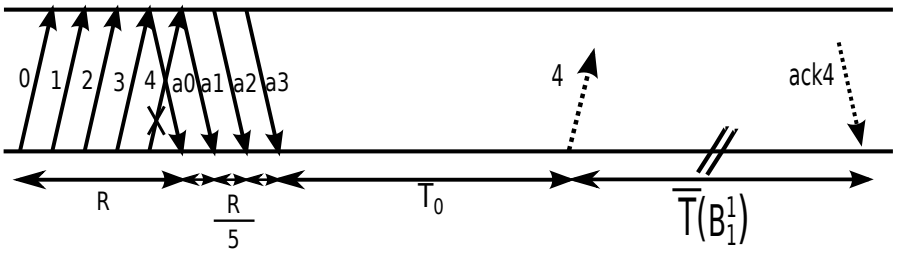

FIGURE 5.10 – $\bar{T}\left(S_5^5\right)$, perte du cinquième segment

de l'IW conditionne le temps de retransmission. La Figure 5.10 illustre le cas de la perte du dernier segment d'une IW égale à 5 avec Initial Spreading. T_0 est finalement déclenché à $R + \frac{3R}{5}$.

Ainsi, à l'exception de la perte du premier segment, chaque perte ajoute un délai supplémentaire $\in \{R, R+\frac{R}{n}, ..., R+(n-1)\frac{R}{n}\}$ au RTO.

$R_{i,n,j}$ est introduit pour sommer ces délais additionnels lorsque j pertes surviennent parmi les i segments transmis avec une IW de taille n, k indiquant la position de la première perte.

$$R_{i,n,j} = \sum_{k=2}^{i+1-j} \binom{i-k}{j-1} \left\{ R + (k-2)\frac{R}{n} \right\}$$

$\bar{T}\left(S_n^n\right)$ est donc égal à la somme des termes qui décrivent les scénarios pouvant amener un changement d'état :

- le scénario idéal où le flux est transmis sans perte ;
- le scénario où toute l'IW est perdue ;
- les scénarios avec j pertes $\in \{1, .., n-1\}$.

$$\bar{T}\left(S_n^n\right) = q^n(R + (n-1)\frac{R}{n}) + p^n\bar{T}\left(B_n^1\right) + \sum_{j=1}^{n-1} q^{n-j}p^j \left\{ \binom{n}{j} \left(T_0 + \bar{T}\left(B_j^1\right)\right) + R_{i,n,j} \right\} \quad (5.1)$$

S_n^n, avec Fast Retransmit et Fast Recovery

Selon nos différentes hypothèses, B_j^1 est envoyé lorsque 3 DupACKs ont permis d'authentifier le segment perdu.

La Figure 5.11 représente la perte des 2 premiers segments pour S_5^5. Le Fast Retransmit est déclenché à $R + 4\frac{R}{5}$. Il permet donc l'économie de $T_0 - 4\frac{R}{5}$. Soit FR le gain de temps moyen engendré par l'utilisation du Fast Retransmit, si l'on note j le nombre de pertes et x le dernier

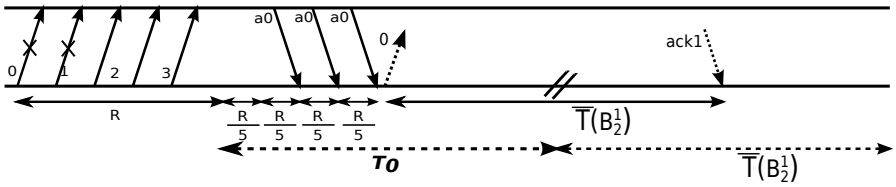

FIGURE 5.11 – $\bar{T}\left(S_5^5\right)$, perte de 2 segments : déclenchement du Fast Retransmit après réception de 3 DupACKs

segment perdu, nous obtenons alors :

$$FR = \sum_{x=j}^{n-3} \binom{x-1}{j-1} \left\{ T_0 - (x+2)\frac{R}{n} \right\} \tag{5.2}$$

et

$$\bar{T}\left(S_n^n\right) = q^n(R + (n-1)\frac{R}{n}) + p^n \bar{T}\left(B_n^1\right) + \sum_{j=1}^{n-1} q^{n-j}p^j \left\{ \binom{n}{j} \left(T_0 + \bar{T}\left(B_j^1\right)\right) + R_{i,n,j}) \right\} - FR$$

Cas de S_i^n, avec $i < n$

Contrairement au cas de D_i^n, S_i^n n'est pas égal à S_i^i. En effet, l'espacement entre deux segments est fixé par le quotient du RTT par la taille maximale autorisée de l'IW. Ainsi, l'espacement entre deux segments est plus important pour S_i^i que pour S_i^n lorsque $i < n$.

Dans le cas de S_i^n, le RTT est séparé en n intervalles et seuls les i premiers sont utilisés.

En utilisant l'équation (5.1), on peut définir S_i^n comme :

$$\bar{T}\left(S_i^n\right) = q^i(R + (n-1)\frac{R}{n}) + p^i \bar{T}\left(B_i^1\right) + \sum_{j=1}^{i-1} q^{i-j}p^j \left\{ \binom{n}{j} (T_0 + \bar{T}\left(B_j^1\right)) + R_{i,n,j}) \right\} - FR$$

Cas des états intermédiaires : B_i^n

Nous étudions maintenant le cas où n segments sont envoyés sous la forme de $\lfloor \frac{n+1}{2} \rfloor$ "mini-bursts" de 2 segments.

Dans cet état intermédiaire marqué par la différence de débit entre le goulot d'étranglement et le débit d'émission, chaque burst est considéré comme indépendant et les pertes entre les bursts également. En revanche, la perte du premier segment du burst entraîne irrémédiablement celle du second.

Soit $P_{j,n}$ la probabilité d'avoir j pertes parmi n segments sachant que les segments sont envoyés par bursts de 2, alors :

$$P_{j,n} = \begin{cases} = \sum_{t=max(0,j-u)}^{\lfloor \frac{j}{2} \rfloor} \left\{ \binom{u}{t,j-2t,u-(j-t)} \times p^t (qp)^{j-2t} q^{2(u-(j-t))} \right\} & \text{si } n = 2u \\ = \quad q \times P_{j,2u} + p \times P_{j-1,2u} & \text{si } n = 2u+1 \end{cases}$$

où t, $j-2t$ et $u-(j-t)$ représentent respectivement le nombre de bursts avec 2 pertes, 1 perte et 0 perte.

Nous pouvons maintenant décrire $P_{j,n}$ par 3 probabilités conditionnelles que nous définissons en fonction du résultat de la transmission du premier des mini-bursts :

$$P_{j,n} = \begin{cases} P_{Z_{j,n}} & \text{si le premier burst subit 0 perte} \\ P_{X_{j,n}} & \text{si le premier burst subit 1 perte} \\ P_{Y_{j,n}} & \text{si le premier burst subit 2 pertes} \end{cases} \tag{5.3}$$

En utilisant (5.3) et le même raisonnement que pour S_n^n, B_n^n peut maintenant être défini.

Dans un premier temps, nous pouvons noter que :

$$\forall n > i, \ B_i^n = B_i^i$$

Nous allons maintenant différencier le cas où le premier segment n'est pas perdu des autres cas. $P_{X_{j,n}} + P_{Y_{j,n}}$ est la probabilité d'avoir j pertes parmi les n segments envoyés sachant que le premier segment n'a pas été perdu. De la même façon que pour la Figure 5.7, cela implique que le prochain état est atteint à $R + T_0$.

En revanche, avec une probabilité $P_{Z_{j,n}}$, le prochain état est atteint à T_0.

$\bar{T}(B_n^n)$ est donc égal à la somme des différentes durées de délivrance qui dépendent directement du nombre de pertes et de leurs positions.

$$\bar{T}(B_n^n) = q^n R + \sum_{j=1}^{n} \left\{ (P_{X_{j,n}} + P_{Y_{j,n}}) \times (R + T_0 + \bar{T}(B_j^1)) + P_{Z_{j,n}} \times (T_0 + \bar{T}(B_j^1)) \right\} \tag{5.4}$$

5.3.5 Généralisation : $\forall i \geq n$

Notre modèle cible principalement la transmission d'un nombre de segments inférieur ou égal à l'IW. En restreignant ainsi la portée de notre modèle, nous prenons garde à ce que ni les mécanismes de reprise, ni les algorithmes de CA ne réduisent sa précision et donc son intérêt. Ainsi, vu de la complexité supplémentaire nécessaire à la modélisation de D_i^n et S_i^n pour $i \geq n$ et conformément à l'hypothèse précédente, nous avons décidé de ne modéliser que B_i^n pour $i \geq n$. La modélisation de B_i^n est quant à elle essentielle car ce dernier est utilisé par D_i^n et S_i^n avec

$\forall i \leq n$ dès lors que des pertes nécessitent la retransmission de segments de l'IW.

$B_i^n \ \forall i \geq n$

Nous considérons que les n premiers segments ont été envoyés. Dans le cas sans perte, les $i-n$ segments restant sont envoyés en Slow-Start ; dans les autres cas, les segments perdus le sont à l'expiration de T_0 avec une CWND égale à 1.

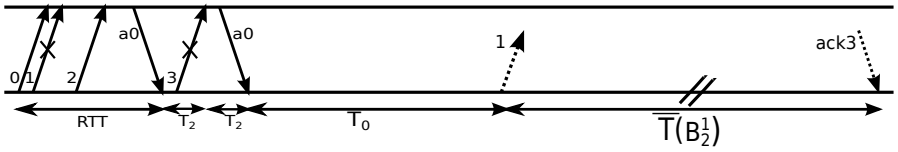

FIGURE 5.12 – $\bar{T}\left(B_4^3\right)$, avec perte des segments 2 et 4

La Figure 5.12 dépeint un des scénarios possibles de B_4^3 dans lequel le deuxième et le quatrième segments sont perdus. A l'arrivée de l'ACK du premier segment, T_0 est mis à jour et le quatrième segment est envoyé. $2 * T_B$ secondes après, soit le temps correspondant à deux fois le temps d'émission du routeur du goulot d'étranglement, l'ACK du troisième segment arrive. Comme l'ACK du premier segment n'arrive pas avant l'expiration du timer de retransmission, on rentre dans l'état B_2^1 à $R + T_0$, avec les segments numéros 1 et 3 à retransmettre.

Lorsqu'il n'y a pas de pertes dans le premier RTT, $CWND = 2 * IW$, tandis que dans le cas avec perte, la position de la première perte détermine le nombre de mini-bursts de 2 segments qui peuvent être envoyés avant de devoir attendre l'ACK du segment perdu, c'est-à-dire avant que l'émetteur ne puisse avoir connaissance d'une perte.

Soit j le nombre de pertes dans l'IW et u le nombre de succès initiaux, $M = min\{i-n, 2\times u\}$ est alors le nombre de segments en attente d'émission qui peuvent être envoyés par mini-bursts de 2 segments dans le second RTT.

Dans la suite, nous définissons $\bar{T}\left(\beta_{i,j,M}^n\right)$ comme la durée moyenne nécessaire à l'émission de i segments avec une IW de taille n sachant que j segments ont été perdus et que le segment $u+1$ a été le premier segment perdu. Cela signifie que les u premiers segments ont bien été validés dans le premier RTT, et que l'émetteur envoie ensuite M segments dans le second RTT par mini-bursts de 2 segments avant d'attendre pour le $(u+1)^{ème}$ ACK. Lorsque le timer de retransmission expire, les $n-j$ segments perdus dans l'IW, les t segments potentiellement perdus durant les M retransmissions et les $i-n$ segments qui n'ont pas encore été émis sont envoyés avec une IW égale à 1.

$$\bar{T}\left(\beta_{i,j,M}^n\right) = \sum_{t=0}^{M} \left\{ P_{t,M} \times \left[R + T_0 + \bar{T}\left(B_{i-(n-j)-(M-t)}^1\right) \right] \right\} \tag{5.5}$$

Soit $P_{j,n,u}$, la probabilité d'avoir j segments perdus parmi n segments sachant que les u premiers segments envoyés ont été validés :

$$P_{j,n,u} = \begin{cases} P_{X_{j-1,n-(u+1)}} & \text{si } u \text{ pair} \\ P_{Y_{j,n-u}} & \text{si } u \text{ impair} \end{cases}$$

En utilisant (5.5), nous pouvons définir $\bar{T}\left(B_i^n\right)$:

$$\begin{aligned} \bar{T}\left(B_i^n\right) &= q^n(R + \bar{T}\left(B_{in}^{2n}\right)) + p^{\lfloor \frac{n}{2} \rfloor + 1}(T_0 + \bar{T}\left(B_i^1\right)) \\ &+ \sum_{j=1}^{n-1} \sum_{u=1}^{n-j} \left\{ P_{j,n,u} \times \bar{T}\left(\beta_{i,j,M}^n\right) + P_{Y_{j,n}}(T_0 + \bar{T}\left(B_{i-(n-j)}^1\right)) \right\} \end{aligned}$$

En utilisant D_i^n, B_i^n et S_i^n, nous pouvons maintenant résoudre les équations précédentes et donc calculer la durée de délivrance moyenne nécessaire à la transmission d'une IW avec et sans Initial Spreading.

5.4 Validation

L'objectif que nous recherchons dans ce chapitre est double :

- proposer et valider un modèle de TCP pour la transmission des flux courts qui tienne notamment compte des différentes sortes de bursts ;
- valider grâce à ce modèle l'intérêt de l'Initial Spreading et confirmer ainsi nos résultats obtenus par simulation.

Afin d'atteindre nos objectifs, nous avons confronté nos résultats de simulation aux estimations obtenus par le modèle. Nous estimons que le recours à NS2 est pertinent pour la validation de nos hypothèses car :

- La petite taille des flux qui nous intéressent permet d'accorder du crédit aux simulations NS2 et de garantir la pertinence de leurs résultats. En effet, dans la majorité des transmissions d'un flux court, l'algorithme de CA n'est pas utilisé, et seuls l'IW, le Slow Start et les mécanismes de reprises affectent le résultat. Or, au contraire de Cubic, l'implantation de ces mécanismes dans NS2 ne souffre aucune contestation.
- En utilisant NS2, nous avons pu contrôler les protocoles utilisés par la pile TCP/IP et nous assurer ainsi que le modèle décrit les mêmes protocoles que ceux employés par les dernières versions de NS2.

Dans la suite, nous utilisons le même banc de test que dans les chapitres précédents (Figure 4.1). Les milliers d'itérations de nos simulations nous ont permis d'obtenir un bon intervalle de confiance ; nous l'avons représenté sur nos courbes pour un niveau de confiance de 95 %.

5.4.1 Évaluation du modèle

Le modèle a pour objectif d'estimer le temps moyen de délivrance d'une connexion courte en ayant connaissance du RTT moyen et du taux de perte par segment. Nous utilisons en entrée de notre modèle le RTT moyen mesuré lors des simulations, ainsi que le taux de perte par segment mesuré sur l'émission d'un grand nombre de segments "indépendants" dans le réseau congestionné.

Bursts de faible taille

Notre modèle est fondé sur une appréhension précise des bursts et en particulier des différents impacts qu'ils peuvent avoir sur la performance d'une connexion en milieu congestionné. Nous avons ainsi pris soin d'isoler les deux principaux types de burst : les bursts initiaux et les mini-bursts. Finalement, nous avons fait l'hypothèse que seuls les mini-bursts affectent la performance d'un flux transmis avec Initial Spreading.

Nous allons donc évaluer en premier lieu la précision de notre modélisation des mini-bursts. Pour cela, nous comparons les résultats de notre modèle à ceux obtenus par simulation pour la transmission de flux courts avec une IW d'1 segment.

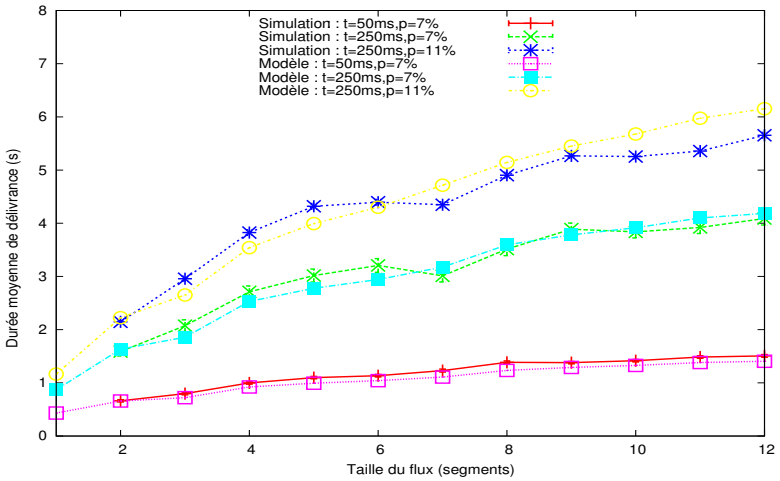

FIGURE 5.13 – Comparaison entre le modèle et les simulations pour la transmission de flux courts avec IW = 1 dans différents environnements

La Figure 5.13 présente une comparaison entre le temps de délivrance moyen prévu par le modèle et le résultat des simulations pour la transmission de flux courts avec une IW égale à 1 pour divers RTTs et taux d'erreur par segment. Le modèle permet une prédiction fine des résultats obtenus par simulation et semble valider nos hypothèses sur les bursts autres que le burst initial et notamment sur l'évolution du trafic TCP par mini-bursts.

Nous pouvons néanmoins remarquer que les bénéfices des mécanismes de reprise sont plus marqués sur les courbes obtenues par simulation que par le modèle. Ainsi, comme nous avons pu

le voir dans le chapitre 4, l'utilisation de mécanismes de reprise en environnement congestionné a pour conséquence une progression non-monotone de la durée moyenne de délivrance. Par exemple, un flux de 7 segments qui a 4 segments transmis dans le troisième RTT a une probabilité plus importante de récupérer une perte sans avoir à attendre un RTO qu'un flux de 5 segments n'ayant que 2 segments émis dans le troisième RTT. La durée de délivrance moyenne pour un flux de 7 segments est donc légèrement inférieur ou sensiblement égal à celui d'un flux de 5 segments. Or cette progression non monotone n'est pas visible dans nos prédictions.

Bursts de taille plus importante

Dans le cas de la prédiction des bursts initiaux, le modèle "bursty loss model" que nous utilisons est moins précis. Dans la partie 5.2.3, nous avons souligné ses faiblesses. Celles-ci se retrouvent dans la précision de notre modèle analytique et sont mis en évidence par la Figure 5.14 qui permet de comparer les résultats prédits par le modèle à ceux mesurés par simulation dans le cas d'une fenêtre initiale de 10 segments sans Initial Spreading.

Ainsi, on peut constater que pour les tailles de bursts inférieures à 7 segments, le fait que le modèle de burst ne tienne pas compte de l'augmentation du taux de perte relative à la taille du burst rend notre modélisation du flux court plus optimiste que les simulations. En revanche, pour les tailles de bursts plus importantes, notre prédiction devient pessimiste. En effet, avec le modèle de burst choisi, une perte entraîne forcément la perte de l'ensemble des segments suivants du burst. Or nous avons pu constater que cette hypothèse n'était pas complètement vérifiée. Il est donc probable que certains des segments du burst puissent être correctement transmis malgré la perte d'un des segments, ce qui, compte tenu de la faible taille du flux a une importance forte.

En conclusion, les faiblesses du modèle de burst pris en considération dans notre modélisation TCP affectent sa capacité à prédire précisément les comportements de la transmission des grandes tailles de bursts. Néanmoins, notre modèle permet une très bonne précision dans le cas de la transmission de bursts de faible taille, ce qui doit nous permettre d'estimer avec précision le comportement de l'Initial Spreading.

5.4.2 Validation de l'Initial Spreading

Nous pouvons maintenant utiliser notre modèle pour valider les résultats obtenus avec l'Initial Spreading.

La Figure 5.14 compare les durées moyennes de délivrance estimées par notre modèle mathématique à celles obtenues par simulation dans le cas d'une transmission de flux courts avec et sans Initial Spreading et des fenêtres initiales de 1 et 10 segments. Dans ce test, le taux moyen d'erreur par segment a été pris égal à 6.5% et le RTT moyen était de 480ms. A nouveau, les résultats obtenus par le modèle et par simulation sont très proches, à l'exception du cas sans Initial Spreading.

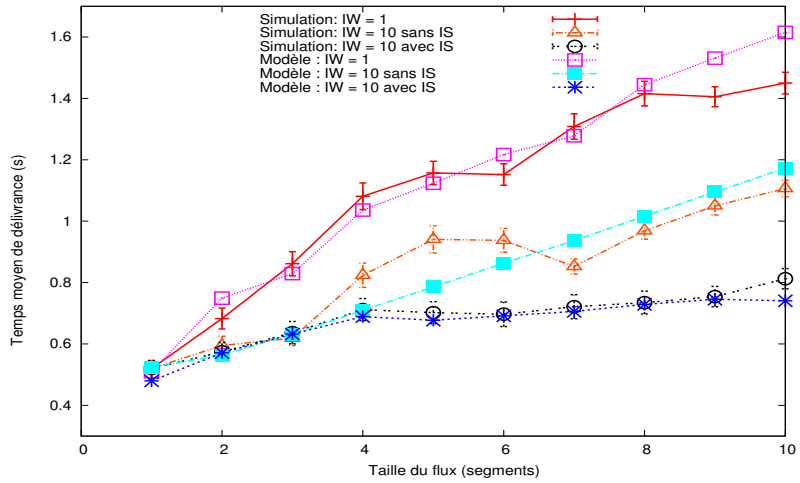

FIGURE 5.14 – Comparaison entre le modèle et les simulations pour la transmission de flux courts avec IW = 1 et IW = 10 avec et sans Initial Spreading

Le modèle confirme ainsi notre compréhension des bursts et les résultats obtenus en utilisant l'Initial Spreading pour la transmission de flux courts avec une grande IW.

Par ailleurs, bien que la précision de notre modèle soit moins bonne dans le cas des grands bursts, il permet néanmoins de constater l'écart de performance entre la transmission d'un flux court avec et sans Initial Spreading dans un environnement congestionné.

Finalement, le modèle confirme que l'utilisation de l'Initial Spreading permet une meilleure prédictibilité des performances, tout comme semblait le montrer les différences entre les intervalles de confiance des simulations avec et sans Initial Spreading.

5.5 Conclusion

Dans ce chapitre, nous avons présenté un modèle analytique pour les connexions TCP visant à estimer le temps de délivrance moyen de flux courts. Ce modèle est adaptatif et peut ainsi être aisément modifié pour modéliser les futures évolutions de TCP qui ne manqueront pas d'être standardisées (par exemple Tail Loss Probe 3.1.2 et Early Retransmit 3.1.2).

Ce modèle est focalisé sur une compréhension fine des bursts et de leurs conséquences sur la transmission d'un flux court dans un environnement congestionné. Il prend en compte de façon itérative les différents scénarios pouvant survenir, afin de montrer dans un premier temps l'impact de l'augmentation de la taille du burst initial, mais aussi les bénéfices apportés par l'utilisation de l'Initial Spreading.

Le modèle confirme ainsi que l'Initial Spreading permet l'augmentation de la taille de la fenêtre initiale de 3 à 10 segments sans pénaliser la performance moyenne, et ce, même en milieu congestionné.

6 Implantation de l'Initial Spreading

Dans les chapitres précédents, nous avons proposé et évalué par simulation une solution qui répond au besoin d'amélioration de la transmission des flux courts dans un réseau congestionné. Notre modélisation analytique des flux TCP courts a ensuite corroboré les bons résultats obtenus et montré que l'appréhension et le traitement fin des bursts permettent à l'Initial Spreading d'offrir un gain de performance significatif par rapport aux autres solutions envisagées.

Néanmoins, si la simulation et la résolution analytique sont nécessaires pour dessiner les grandes lignes d'une évolution protocolaire de TCP, il est indispensable de leur associer des expérimentations réelles afin de leur garantir crédibilité et légitimité. En effet, de nombreux paramètres rendent l'évaluation d'un protocole tel que TCP complexe et sujette à caution, a fortiori dans un environnement congestionné.

Nous retiendrons notamment :

- A l'échelle du réseau :
  - les nombreuses interactions entre les différentes connexions qui influent sur leurs comportements de façon peu prédictible ;
  - les interactions avec les autres éléments du réseau tels que les routeurs dont le remplissage des buffers affecte la performance d'une connexion ;
  - l'environnement instable dans lequel évolue la connexion.
- A l'échelle du protocole :
  - les interactions avec les différents protocoles de la pile ;
  - le caractère évolutif du protocole TCP lui-même. En effet, comme on a pu le voir dans les chapitres 2 et 3, TCP est issu d'un travail "collaboratif" et son comportement évolue au gré d'ajouts et de modifications de parcelles protocolaires.

Par ailleurs, l'implantation du mécanisme elle-même peut révéler de nombreuses surprises.

Dans le cas de l'Initial Spreading, la phase d'implantation, bien que délicate, s'est de plus avérée essentielle car elle a permis de faire progresser le mécanisme initial vers sa version finale. Plusieurs implantations ont ainsi été nécessaires. Les premières nous ont permis de relever certaines limites de notre mécanisme et de le modifier avant de pouvoir le soumettre à la communauté "transport" de l'IETF. L'accueil favorable reçu par l'Initial Spreading ainsi que les nombreux commentaires et conseils nous ont alors permis d'optimiser notre implantation et d'affiner notre algorithme afin d'aboutir à la version actuelle de l'Initial Spreading.

Alors que le chapitre suivant validera le comportement de l'Initial Spreading, ce chapitre permet de retracer l'évolution du mécanisme, de son implantation et des tests que nous avons menés.

6.1 Implantation de l'Initial Spreading

L'objectif de cette implantation est d'évaluer le comportement de l'Initial Spreading en environnement réel et de confronter les résultats obtenus à ceux de nos simulations et analyses. Nous avons néanmoins profité de cette phase de développement pour tester également le comportement et les performances de variations protocolaires autour de l'algorithme originel. Ainsi, nous avons notamment introduit la possibilité de décider manuellement de la durée d'espacement entre deux émissions de segments appartenant à l'IW.

Afin de réaliser notre première implantation de l'Initial Spreading, nous avons modifié le code de la couche TCP du noyau Linux, ajoutant un "patch" de quelques centaines de lignes qui nous a permis de tester l'Initial Spreading tel qu'il a été présenté dans les chapitres précédents mais également d'évaluer les conséquences d'un éventuel changement d'espacement.

Dans la suite de ce chapitre, nous présentons cette implantation. Cette partie est essentielle car elle permet non seulement de reproduire l'Initial Spreading mais souligne également l'écart entre la validation d'une solution par simulation et modélisation analytique et la validation en conditions réelles.

6.1.1 Implantation fondée sur l'utilisation de jiffies

Le jiffy : une contrainte d'implantation

Un jiffy est une unité atomique de temps fixant la granularité des temporisations dans le noyau et en particulier de TCP. Le micro-processeur se réveille ainsi à chaque jiffy pour exécuter les instructions de la couche TCP en attente. Ce paramètre est très important dans le cas de notre implantation car il conditionne les valeurs que l'on peut attribuer à l'espacement.

Principe général de l'implantation

Afin d'implanter le mécanisme de l'Initial Spreading, nous avons besoin de :

- calculer le RTT afin de déterminer une durée d'espacement ;
- réguler l'émission des paquets de la fenêtre initiale.

Le calcul du RTT est déjà implanté dans le noyau, il nous suffit donc de récupérer le résultat dès qu'il a été obtenu, c'est-à-dire après réception du segment SYN/ACK.

La régulation de l'émission des segments est plus complexe et nécessite deux étapes :

- l'allocation en mémoire des paquets ;
- la transmission des segments dans la pile protocolaire, de la couche TCP à la couche IP.

Variables d'implantation

Nous utilisons 4 variables distinctes :

- *count* : nombre de segments de la première fenêtre restant à allouer en mémoire. Initialisée à la valeur de l'IW en utilisant la variable *initcwnd* présente dans le noyau, cette variable est décrémentée à chaque nouvelle allocation.
- *burst* : nombre de segments de la première fenêtre restant à émettre. Initialisée également à *initcwnd*, cette variable est décrémentée à chaque émission de paquet.
- $T_{Spreading}$: valeur de l'espacement :
  - Une valeur nulle correspond à l'utilisation d'un espacement proportionnel au RTT, soit $T_{Spreading} = \frac{RTT}{initcwnd}$). Cela nous permet de tester l'Initial Spreading dans les mêmes conditions que la simulation.
  - Toute autre valeur correspond à la valeur en jiffies de l'espacement entre deux segments. Nous y reviendrons dans la section suivante.
- *lock* : variable booléenne dont le rôle est de bloquer l'émission de segments durant les périodes d'espacement.

Lorsque les variables *count* et *burst* sont nulles, l'Initial Spreading est inactif et la pile reprend son comportement normal.

Nous avons également introduit d'autres variables afin de pouvoir tester différents comportements d'Initial Spreading. Ces variables sont accessibles à l'utilisateur via la commande *sysctl* ou le système de fichiers virtuels */proc/sys/net/ipv4* et nous permettent entre autres de modifier les paramètres d'étalement :

- *tcp_initial_spreading* : booléen permettant d'activer ou de désactiver l'Initial Spreading ;
- *tcp_initial_spreading_fixed_delta* : initialise $T_{Spreading}$ à la valeur désirée;
- *tcp_initial_spreading_window_delta* : permet de modifier le nombre de paquets devant être espacés.

Finalement, ces variables sont principalement utilisées à travers 3 fonctions qui nous permettent d'utiliser l'Initial Spreading :

- tcp_spreading_recalc : cette fonction permet de calculer le nombre de segments à espacer ainsi que la durée d'espacement. Elle est appelée après réception du SYN/ACK une fois le RTT déterminé.
- tcp_spreading_reset_timer : cette fonction relance le timer pour une durée égale à $T_{Spreading}$. Elle est appelée après l'émission d'un segment tant que l'Initial Spreading est actif.
- tcp_spreading_timer : cette fonction est appelée à l'expiration du timer et déclenche l'émission d'un nouveau segment.

Détails de l'implantation

La Figure 6.1 représente le diagramme de fonctionnement de la pile TCP. La relation que nous avons fait figurer entre *tcp_send_skb* et *tcp_write_xmit* est essentielle car elle permet de

contrôler l'état de la connexion (fenêtre, TSP, Nagle...) avant d'appeler *tcp_transmit_skb* pour transmettre les segments à la couche IP.

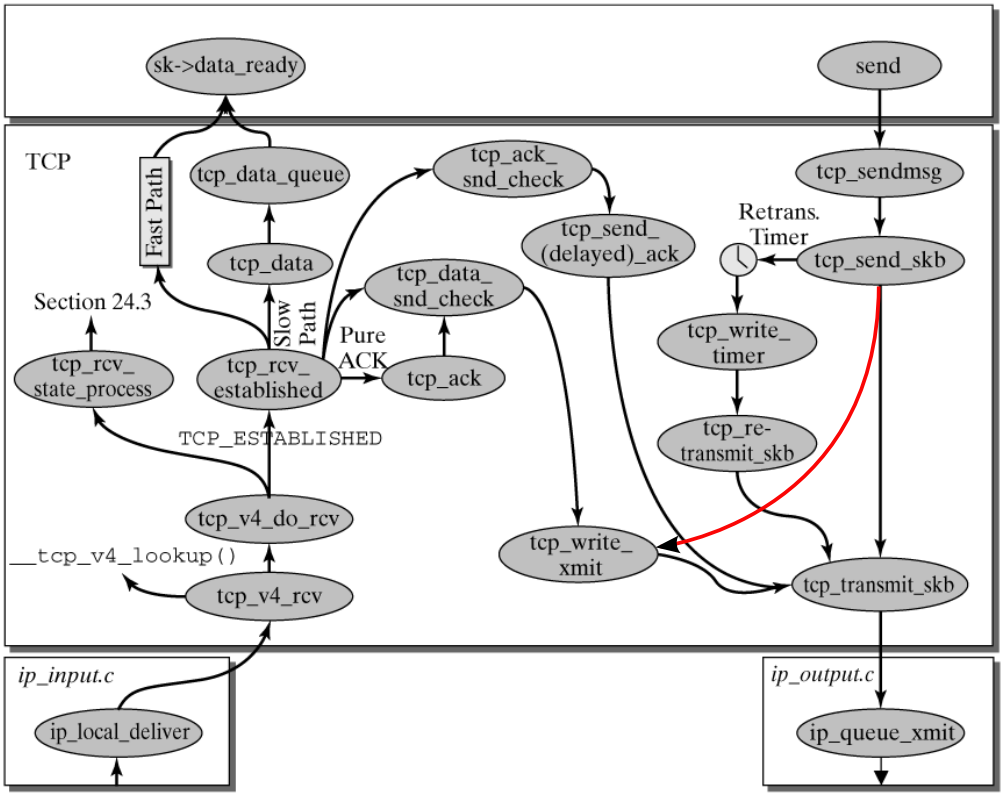

FIGURE 6.1 – Fonctionnement de la pile TCP dans le noyau Linux

Après avoir initialisé les différentes variables et notamment le temps d'étalement, l'essentiel de notre mécanisme consiste à vérifier l'activité de l'Initial Spreading avant l'émission d'un segment et d'exécuter si nécessaire les actions résultantes. Nous ajoutons donc un nouveau test à la fonction *tcp_write_xmit* afin de vérifier la valeur de *lock*, et d'appeler *tcp_transmit_skb*.

Dans le cas où il reste des segments à espacer, c'est-à-dire *burst*$\neq$ 0, l'émission est alors verrouillée et le timer d'étalement est réarmé. L'expiration du timer appelle la fonction tcp_spreading_timer qui déverrouille l'émission et pousse l'envoi immédiat du segment.

6.1.2 Limite de cette implantation

Incompatibilité des mécanismes TSO/GSO, Nagle et Initial Spreading

TSO (TCP Segmentation Offload) et GSO (Generic Segmentation Offload) visent à réduire la charge du processeur en réduisant le nombre de segments traversant la couche TCP du système. Au lieu de faire la fragmentation dans le noyau, un segment unique de taille supérieure ou égale à la Maximum Segment Size (MSS) est envoyé à la carte réseau – Network Interface Card (NIC)

– ce qui réduit considérablement le nombre d'opérations effectuées par le processeur et rend le processus de transmission plus efficace. La segmentation est ensuite effectuée par la carte réseau. Cette dernière doit alors ajouter le bon en-tête à chacun des nouveaux segments créés.

L'utilisation de TSO/GSO annihile complètement le mécanisme de l'Initial Spreading tel que nous l'avons implanté. L'IW est envoyée sous la forme d'un super-segment contenant l'intégralité des données jusqu'à la carte réseau, cette dernière fragmentant le super-segment en n segments qu'elle envoie en rafale sans tenir compte de notre volonté d'espacement.

TSO et GSO doivent donc être désactivés pour que cette implantation de l'Initial Spreading fonctionne, l'idéal étant de limiter leur phase d'inactivité à l'émission de l'IW. En effet, durant l'émission de l'IW, la faible quantité de données envoyées ne justifie pas leur utilisation. En revanche, il est intéressant de les conserver lorsque TCP est dans son état stable.

L'algorithme de Nagle [62] a pour objectif d'agréger les données en retardant l'émission des segments afin de ne pas surcharger le réseau d'en-têtes inutiles. Il est donc également incompatible avec l'Initial Spreading. En revanche, le rendre inactif a moins de conséquences car son intérêt a été fortement amoindri par l'évolution des standards de communication. A son introduction, il ciblait notamment les segments d'1 octet de données envoyés sur le réseau.

Finalement, en implantant notre mécanisme, nous avons pu nous rendre compte de l'impact des contraintes réelles "à l'échelle du protocole", et notamment des nombreuses interactions avec les autres optimisations qui peuvent diminuer voir annuler l'efficacité de l'Initial Spreading. Par ailleurs, nous avons maintenant la possibilité de tester l'Initial Spreading dans un environnement réel afin d'observer les conséquences des contraintes réelles "à l'échelle du réseau". Nous étudierons notamment le comportement de l'Initial Spreading dans un environnement congestionné.

6.2 Premiers enseignements des expérimentations en environnement réel

L'objectif de cette section n'est pas d'évaluer l'efficacité de l'Initial Spreading face aux solutions concurrentes, mais d'étudier l'impact de l'environnement réel sur son comportement.

6.2.1 Comportement du mécanisme dans le cas avec congestion

La Figure 6.2 présente les résultats obtenus lors de tests réalisés en environnement réel congestionné. Nous expliciterons dans le chapitre suivant le mode opératoire utilisé et analyserons plus en détail les résultats obtenus. Nous pouvons toutefois d'ores et déjà utiliser cette figure pour cerner les limites de notre algorithme initial.

Ainsi, alors que nous avons montré dans les chapitres précédents que le principal attrait de l'Initial Spreading est son efficacité à transmettre des flux courts dans les réseaux congestionnés, les premières expériences réelles que nous avons menées nous ont permis d'identifier une faiblesse

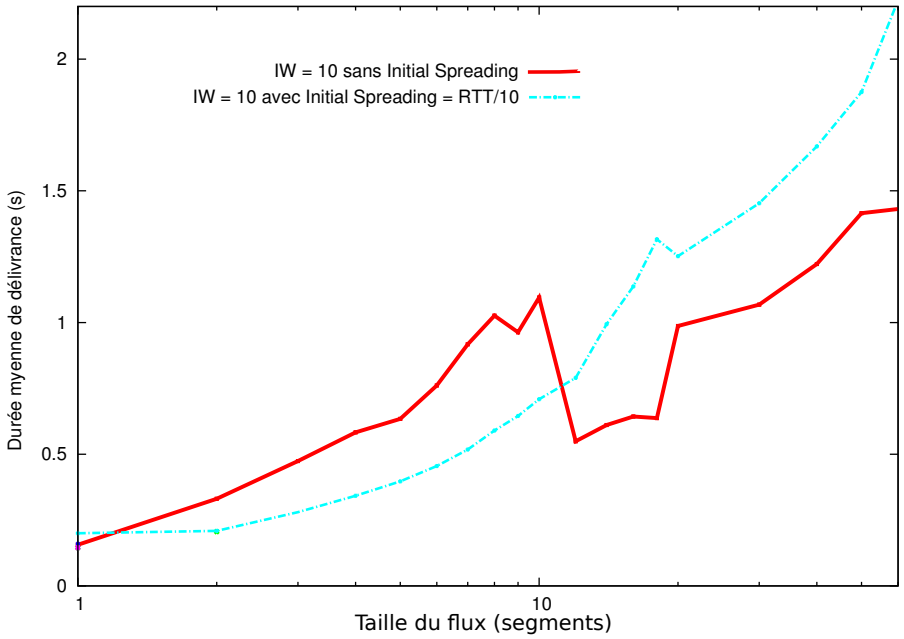

FIGURE 6.2 – Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 42ms

potentielle de l'Initial Spreading jusqu'alors indétectable par nos simulations et analyses.

En effet, le mécanisme de l'Initial Spreading se fonde sur la mesure instantanée du RTT lors de l'échange de SYN-SYN/ACK pour fixer l'étalement ($T_{Spreading}$). Or, dans le cas d'une perte du segment SYN, le RTT mesuré ne reflète pas le RTT réel mais le RTT additionné du RTO.

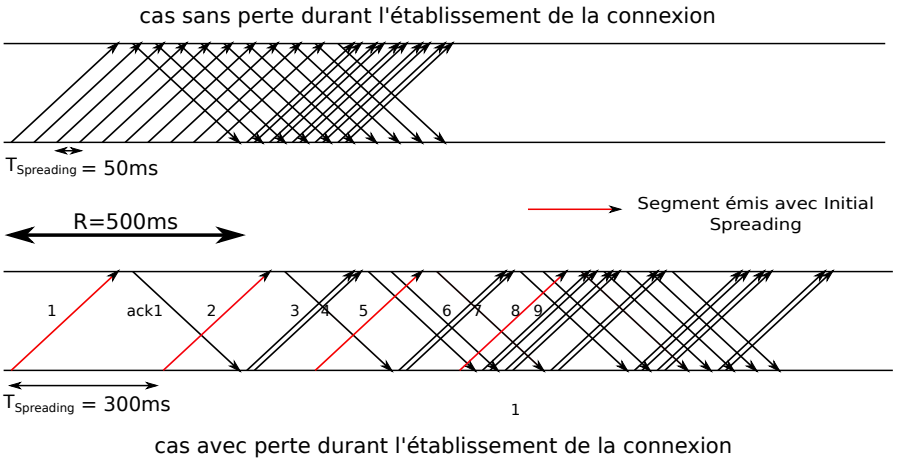

FIGURE 6.3 – Conséquences d'une mesure erronée du RTT durant l'établissement de la connexion sur la transmission de 20 segments

L'utilisation de cette mesure erronée du RTT conduit l'Initial Spreading à utiliser un espacement trop important entre les segments, ce qui aura pour conséquence d'annuler l'intérêt de l'augmentation de la fenêtre initiale. En effet, comme nous pouvons le voir avec la Figure 6.3, l'utilisation d'un $T_{Spreading}$ trop important réduit d'une part le nombre de segments émis durant le premier RTT, mais également le nombre de segments qui seront émis grâce à l'Initial Spreading, ce dernier n'étalant que les segments appartenant à l'IW. Les segments en rouge sur la figure réprésentent les segments émis à l'expiration des timers dédiés à l'Initial Spreading.

On peut ainsi constater que la courbe représentant le temps de délivrance moyen pour une connexion courte lorsque l'on utilise un $T_{Spreading}$ classique, c'est-à-dire $T_{Spreading} = \frac{RTT}{IW}$, montre des performances moins bonnes que celle représentant la RFC 6928. La qualité de la performance moyenne de l'Initial Spreading est détériorée par les quelques cas où une perte a lieu pendant l'établissement de la connexion. Ces cas sont pourtant peu nombreux, les segments échangés pendant l'établissement de connexions étant beaucoup plus petits que les segments de données et donc moins susceptibles de souffrir de la congestion.

Nos simulations ne nous ont pas permis d'identifier ce problème car l'échange SYN-SYN/ACK n'étant pas simulé, nous utilisions en paramètre d'entrée la valeur du délai aller-retour introduit (T_{AR}), et non la valeur réelle du RTT. Finalement, nous testions un espacement fixe de grande taille et occultions les potentielles pertes qui peuvent survenir lors de l'établissement de la connexion.

6.2.2 Comportement du mécanisme proposé dans le cas sans congestion

Les expériences précédentes ont donc montré que l'utilisation d'un $T_{Spreading}$ variable proportionnel au RTT pouvait altérer l'efficacité de l'Initial Spreading dans les environnements congestionnés et finalement détériorer la performance moyenne. Or, le comportement de l'Initial Spreading en congestion était jusqu'alors son atout principal, et nous permettait d'accepter une efficacité moindre dans les réseaux sans congestion.

Dans la section 4.3.2, nous avons ainsi mis en avant les conséquences liées à l'utilisation de l'Initial Spreading dans les cas sans congestion. L'ajout d'espacements correspondant à la division du RTT par l'IW peut ainsi augmenter la durée de transmission d'un flux court comparativement à la solution sans Initial Spreading. En effet, dès lors que le flux peut être transmis en intégralité dans le premier RTT, c'est-à-dire que le nombre i de segments à transmettre est inférieur à n la taille de l'IW, alors l'utilisation de l'Initial Spreading ajoute une durée supplémentaire égale à $(i-1) * (\frac{RTT}{i} - T_B)$ secondes.

Ce temps supplémentaire est directement dépendant de la durée du RTT et peut dépasser plusieurs centaines de millisecondes dans les pires cas. En revanche, ce délai se réduit considérablement dès que la transmission du flux nécessite plus d'un RTT. Par exemple, les temps minimaux de délivrance de 11 segments avec et sans Initial Spreading et une IW de 10 segments sont égaux. De fait, l'utilisation de l'Initial Spreading dans les réseaux sans congestion avec une

grande IW permet malgré tout d'obtenir des performances sensiblement meilleures que celles obtenues avec la RFC3390 ($IW = 3$) dès que le flux a une taille supérieure à 3 segments. Vu les performances obtenues en environnement congestionné, nous avions considéré cette faiblesse comme un moindre mal.

Finalement, ces premières expérimentations réelles montrent qu'en l'état, l'implantation de l'Initial Spreading n'atteint pas les performances escomptées pour la transmission des flux courts, y compris en comparaison de la RFC 6928, que le réseau soit congestionné ou non. Retrouver l'efficacité et l'intérêt de l'Initial Spreading nécessite un changement dans le calcul de l'espacement, et donc une étude approfondie des conséquences que cela peut occasionner.

6.2.3 Discussion sur $T_{Spreading}$

Les paragraphes précédents ont montré la nécessité de faire évoluer l'espacement afin de rendre notre algorithme plus performant. Une solution intuitive est d'utiliser un $T_{Spreading}$ fixe de durée minimale. Pour autant, il est primordial de ne pas perdre de vue l'objectif principal de l'espacement qui est d'éviter l'envoi de bursts de segments qui seraient massivement perdus en cas de congestion.

Ainsi, dans les chapitres précédents, nous avons souligné les effets indésirables des bursts et notamment du burst initial sur la performance moyenne d'une connexion courte, et proposé une solution à même de les contrer fondée sur un étalement suffisant de l'émission des segments pour permettre l'éclatement du burst considéré.

Le recoupement des résultats obtenus par simulation et modélisation analytique prouve que l'espacement engendré par l'Initial Spreading permet effectivement de ne plus subir les conséquences des bursts. En effet, la concordance des résultats de simulations et des résultats analytiques valide le fait que le temps d'étalement $T_{Spreading}$ utilisé dans les simulations est suffisant pour garantir aux segments de l'IW une relative "décorrélation". Ainsi, séparer l'émission de deux segments d'un $T_{Spreading}$ égal à $\frac{T_{AR}}{n}$, où n est la taille de l'IW et T_{AR} le temps d'aller retour minimal, leur permet de ne pas appartenir à un même burst.

Nous sommes désormais en droit de nous demander si ce temps est indispensable ou s'il est possible d'envisager une durée d'espacement plus restreinte.

D'après la définition du burst que nous avons donnée en 5.2.1, deux segments n'appartiennent pas à un même burst si l'état du buffer du routeur du goulot d'étranglement a eu la possibilité d'évoluer entre leurs réceptions successives. Ainsi, une condition minimale à la "décorrélation" des segments émis consécutivement est d'avoir un espacement suffisant pour permettre au buffer d'évoluer entre leurs réceptions successives, c'est-à-dire qu'il ait eu le temps de vider au moins un segment de sa file d'attente entre les 2 émissions. Finalement, cela revient à considérer comme nous l'avons posé dans le chapitre 5 que deux segments appartiennent à un même burst s'ils sont émis à un rythme supérieur au débit du goulot d'étranglement. Par analogie à 4.1, nous supposons ainsi que les deux segments n'appartiennent pas à un même burst car leur réception

est suffisamment espacée pour ne pas augmenter de façon directe la taille de la file d'attente.

Ceci revient à considérer qu'il existe une limite inférieure au temps d'étalement $T_{Spreading}$ égale à T_B, où T_B est le temps nécessaire au routeur du goulot d'étranglement pour émettre un segment. Le franchissement de cette limite résulte en l'émission de bursts.

Dans l'exemple du scénario simulé dans 4.3.2 où le débit du goulot d'étranglement est pris à 10 Mb/s et considérant une taille de segments de 1500 octets, le temps minimal d'espacement serait donc égal à 1.2 ms alors que si on ne prend en compte que le RTT moyen qui vaut 320ms, l'espacement utilisé jusqu'à présent pour une IW de 10 segments est égal à 32ms.

Faire tendre $T_{Spreading}$ vers sa limite basse peut donc permettre d'économiser de précieuses millisecondes dans les réseaux sans congestion tout en préservant l'intérêt de l'Initial Spreading dans les environnements congestionnés.

Bien évidemment, nous n'oublions pas les phénomènes de multiplexage statistique qui vont faire qu'entre l'émission de ces paquets par la source et le passage par le goulot d'étranglement, l'écart entre ces paquets aura pu évoluer.

6.2.4 Considérations autour de l'Initial Spreading

L'Initial Spreading a pour vocation d'améliorer les performances moyennes des connexions TCP en proposant une modification minimale de la pile TCP. Ce dernier point est d'ailleurs crucial car la simplicité de notre mécanisme est l'une des clés de son intérêt.

Afin de tenir compte des dernières considérations qui peuvent nuancer l'intérêt de l'Initial Spreading, ce dernier doit être capable de tirer le meilleur parti des contraintes suivantes :

- $T_{Spreading}$ doit être suffisamment grand pour assurer la "non-corrélation" des segments et de leurs pertes.
- $T_{Spreading}$ doit être le plus petit possible afin de ne pas ajouter de latence non nécessaire, et notamment dans les cas sans congestion.
- $T_{Spreading}$ ne doit pas pâtir d'une mesure erronée du RTT durant l'établissement de la connexion.
- L'implantation de l'Initial Spreading doit être la plus simple et légère possible afin d'être aisément diffusable et reproductible.

Pour répondre à ces contraintes, nous introduisons le paramètre T_{max} qui représente la limite supérieure de $T_{Spreading}$. T_{max} a deux objectifs :

- permettre à l'Initial Spreading de ne pas être dépendant du RTT et ainsi ne plus subir les contrecoups de longs RTT ou d'une mesure erronée du RTT ;
- assurer un espacement minimal et réduire ainsi la latence moyenne.

Le choix de T_{max} résulte d'un compromis. En effet, un T_{max} grand permet de "décorréler" les pertes pour des débits de goulot d'étranglement faibles. En revanche, un T_{max} petit réduit

l'impact de l'Initial Spreading sur les réseaux non-congestionnés ayant un débit de goulot d'étranglement plus élevé.

Débit minimal considéré	T_{max} adapté au débit minimal considéré	Latence ajoutée pour l'émission de 10 segments par rapport à la RFC6928 si le débit réel du lien congestionné est égal à 6 Mb/s
1 Mb/s	12ms	90ms
3 Mb/s	4ms	18ms
6 Mb/s	2ms	0ms
10 Mb/s	1.2ms	création d'un burst maximal de 4 segments
100 Mb/s	0.12ms	création d'un burst maximal de 9 segments

TABLEAU 6.1 – Conséquences du choix de T_{max} pour un goulot d'étranglement avec un débit réel de 6 Mb/s

La Table 6.1 illustre les conséquences que peuvent avoir différents T_{max} sur un réseau avec un goulot d'étranglement ayant un débit réel de 6 Mb/s. Par exemple, choisir un T_{max} de 12ms suffisant pour que l'Initial Spreading ne subisse pas les bursts lorsque le débit le plus faible est supérieur ou égal à 1 Mb/s rajoute 90ms à l'émission de 10 segments avec une IW de 10 segments lorsque le réseau traversé à un débit réel limitant de 6 Mb/s. En revanche, considérant le même goulot d'étranglement, un T_{max} de 1.2ms qui est idéal pour un débit de goulot d'étranglement de 10 Mb/s, crée ici un burst de 4 segments.

Prise en compte des jiffies dans le choix de T_{max}

Le choix de T_{max} n'est pas uniquement conditionné par des considérations théoriques, il est également nécessaire de prendre en compte les limites logicielles de notre implantation et notamment la dépendance vis-à-vis du jiffy (voir 6.2) qui a des implications fortes sur l'émission des segments.

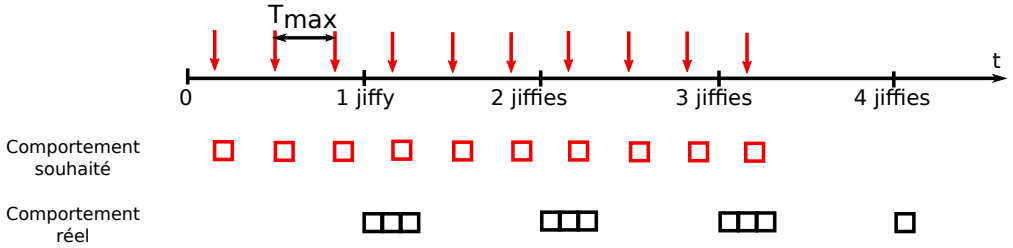

FIGURE 6.4 – Conséquences de l'utilisation de jiffy dans le choix de T_{max}

La Figure 6.4 illustre les conséquences relatives au choix d'un T_{max} trois fois plus faible que le jiffy. Cela se traduit par l'émission de bursts de 3 segments à chaque jiffy. En effet, depuis la dernière interruption, l'Initial Spreading a autorisé l'émission de 3 segments qui partent en un unique burst à l'interruption suivante. T_{max} doit donc être supérieur à la valeur d'un jiffy pour ne pas créer de bursts.

La valeur d'un jiffy est inversement proportionnelle à la variable *HZ*. *HZ* est définie à la compilation du noyau et peut prendre 4 différentes valeurs : 100, 250, 300 et 1000. La valeur définie par défaut dans la plupart des distributions courantes étant 300Hz, le jiffy permet donc une précision des timers d'environ 3.3ms.

Finalement, la considération des jiffies fixe une nouvelle limite à la durée de T_{max} et donc de $T_{Spreading}$. La Table 6.2 nous montre que pour la majorité des utilisateurs, un $T_{Spreading}$ d'un jiffy correspondant à un espacement de 3.3ms permet à l'Initial Spreading de disperser les bursts tant que le débit du goulot d'étranglement des réseaux parcourus est supérieur à 3,6 Mb/s.

Jiffy	T_{max}	Débit minimal
1	$\sim 3.3ms$	3,6 Mb/s
2	$\sim 6.6ms$	1,8 Mb/s
3	$\sim 10ms$	1,2 Mb/s

TABLEAU 6.2 – Conséquences des jiffies sur le T_{max} avec $HZ = 300Hz$

6.2.5 Synchronisation entre connexions due aux jiffies

Outre influencer le choix de l'espacement, l'utilisation de jiffy peut avoir un véritable impact sur l'efficacité de notre mécanisme.

L'Initial Spreading a ainsi pour objectif de limiter l'impact du burst initial en étalant son émission. L'implantation que nous avons proposée modifie le comportement du protocole TCP pour espacer l'émission de l'IW en émettant un segment tous les x jiffies. Les résultats obtenus par cette implantation sont très satisfaisants dès lors que l'émetteur n'émet qu'une connexion à la fois.

En revanche, dans les cas où l'émetteur souhaiterait émettre plusieurs connexions en parallèle, l'utilisation des jiffies va réduire l'efficacité de l'Initial Spreading en recréant des bursts de segments. En effet, à chaque jiffy, le processeur va exécuter les tâches en attente et regrouper en un burst unique l'ensemble des segments des différentes connexions dont la date d'émission est arrivée à échéance entre le temps présent et les x jiffies précédents.

La Figure 6.5 illustre les conséquences des jiffies pour l'émission de trois connexions en parallèle utilisant l'Initial Spreading avec $T_{Spreading}$ =1 jiffy. Bien que les dates d'émission des segments soient suffisamment espacées pour ne pas créer de bursts dans le cas d'une émission instantanée, la synchronisation par jiffy entraîne un regroupement des segments. On peut alors constater que les connexions parallèles rendent impossible l'émission "indépendante" des segments par un même émetteur.

Ainsi, le comportement obtenu par simulation avec NS2, ne tenant pas compte des cycles de processeurs, est différent de ce que l'on obtient avec notre implantation, et n'est reproductible que dans le cas où l'émetteur se borne à émettre une connexion à la fois.

L'implantation fondée sur l'utilisation de jiffies a donc non seulement des conséquences sur le choix de l'espacement, mais peut également entraver le bon fonctionnement de l'Initial Spreading.

Toutefois, ce problème n'est significatif que lorsque l'utilisateur se sert de ce procédé pour améliorer son débit en établissant plusieurs connexions parallèles simultanées avec la même entité finale. Dans les autres cas de figure, l'importance est moindre car les adresses finales étant différentes, la fonction de routage va séparer les segments et seuls les premiers routeurs traversés seront susceptibles d'être affectés.

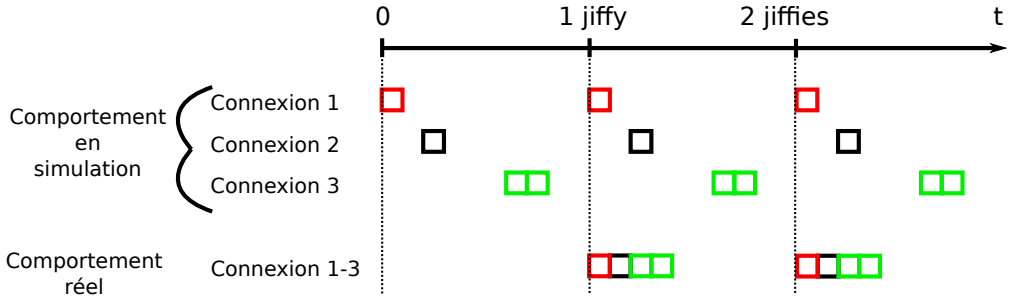

FIGURE 6.5 – Différence entre simulation et implantation réelle dans le cas de connexions en parallèle

En conclusion, notre première implantation nous a permis d'affiner notre compréhension de l'impact réel de l'Initial Spreading et de proposer des améliorations à notre algorithme initial. Nous avons présenté l'Initial Spreading et les conséquences des différents choix d'espacement à l'IETF et pu juger ainsi de l'intérêt qu'il suscitait auprès de ses différents membres. Les nombreuses discussions qui ont suivi nous ont par ailleurs aiguillé vers une solution d'implantation capable de résoudre les problèmes dus à l'utilisation de jiffy. Cela nous a permis de raffiner notre mécanisme.

6.3 Évolution de l'implantation et nouvel algorithme de l'Initial Spreading

6.3.1 Implantation dans le noyau fondée sur FQ/Pacing

Les performances de notre implantation de l'Initial Spreading sont sévèrement amoindries par les réalités logicielles telles que l'utilisation des TSO et GSO, ou encore la synchronisation jiffy. Par exemple, l'Initial Spreading tel que nous l'avons implanté n'arrive pas à gérer les cas où l'émetteur transmet des connexions en parallèle vers une même entité TCP de destination.

Cependant, l'Initial Spreading n'est pas le seul mécanisme d'optimisation à être affecté par ces limites logicielles. Ces derniers temps, de très nombreux chercheurs luttant contre le phénomène de "Bufferbloat" [63], dans l'objectif de réduire la latence ajoutée par le temps que passe chaque segment à attendre dans les buffers des routeurs empruntés, ont été également confrontés à ces problèmes. Ils ont alors proposé quelques solutions permettant de passer outre.

La dernière en date (en cours de développement, mais déjà en test dans les noyaux postérieurs à 3.11) est l'association d'un trio de mécanismes :

- le dimensionnement du TSO ;
- le Pacing ;
- l'utilisation d'un ordonnanceur au niveau de la couche réseau.

Cette solution permet d'appréhender les conséquences du TSO et de garantir une bonne efficacité au traitement des connexions en parallèle.

Gestion du TSO

Cette solution a permis une forte évolution de la gestion du TSO. Jusqu'à présent, la taille du super-paquet transmis via l'utilisation de TSO n'était limitée que par une quantité maximale de données (64 KB dans la majorité des cas, soit environ 42 segments). Une IW de 10 segments ($\sim 15KB$) était donc irrémédiablement transmise en un unique super-segment, avec les conséquences que nous avons précédemment évoquées.

La nouvelle implantation du TSO propose un dimensionnement dynamique de la taille du super-segment fondé sur le débit d'émission et le Pacing. Ainsi, le nouveau TSO permet de regrouper en un super-segment l'équivalent de la transmission d'approximativement 1ms de données au débit de l'émetteur tout en tenant compte d'un potentiel espacement.

L'intérêt de cette évolution est de limiter la taille des bursts engendrés dans l'état stable de TCP, et ce, quel que soit le débit considéré :

- dans le cas de débits faibles : la limitation temporelle est suffisante pour limiter la quantité de donnée envoyée ;
- dans le cas de débits plus importants : la limitation temporelle n'étant plus suffisante pour réduire la taille du burst, le Pacing a été réintroduit dans l'optique de permettre malgré tout une régulation du burst. En effet, l'algorithme tient compte de l'espacement souhaité entre segments ou groupements de segments pour limiter la quantité de données à émettre.

Gestion des connexions multiples en parallèle

FQ (Fair Queuing) est un ordonnanceur chargé d'organiser le flux de segments transmis par la couche réseau. Il permet de différencier et de garder en mémoire chacune des connexions émises afin de leur attribuer un traitement spécifique. Chaque connexion se voit attribuer sa propre file d'attente.

Il est alors possible de jouer sur la priorité des flux mais aussi sur l'espacement entre segments ou super-segments d'un même flux.

Par ailleurs, des timers à haute résolution sont utilisés à la place des jiffies permettant le traitement des données toutes les microsecondes.

L'association de ces trois mécanismes ouvre de réelles perspectives dans le traitement des données, et permet notamment une implantation simple et efficace de l'Initial Spreading.

Implantation finale de l'Initial Spreading

Le trio de mécanismes décrit précédemment a été conçu pour améliorer l'efficacité des connexions à fort débit lorsque les connexions TCP sont dans leur état stable. Pour autant, il est aisé d'en tirer profit et de le modifier pour implanter l'Initial Spreading.

Ainsi, nous pouvons facilement ajouter une variable $T_{Spreading}$ qui nous permet d'étaler les segments de l'IW en prenant toutefois garde de la mettre à zéro dès la réception du premier ACK afin de ne pas espacer les transmissions de segments suivants. Grâce à cet espacement, nous limitons à 1 le nombre de segments de l'IW transmis par super-segment via l'utilisation de TSO, mais n'influençons pas le comportement de TSO pour le reste de la connexion. Finalement, les segments des connexions parallèles ayant des traitements différenciés pourront être émis au travers d'un minimum de bursts grâce à l'utilisation des timers à haute résolution.

6.3.2 Algorithme final de l'Initial Spreading

En se fondant sur les possibilités offertes par la nouvelle implantation, nous pouvons proposer une dernière version de l'Initial Spreading [5] qui tient compte de l'ensemble de nos constats précédents :

1. Le RTT est mesuré pendant l'échange de SYN-SYN/ACK.
2. Si n est la taille de l'IW, jusqu'à n segments peuvent être envoyés tous les $T_{Spreading}$. L'algorithme 1 est utilisé pour calculer $T_{Spreading}$ par rapport à la valeur du RTT et celle de T_{max} qui est la valeur maximale de l'espacement.

Algorithme 1 Calcul du $T_{Spreading}$

Si $\frac{RTT}{n} \leq T_{max}$ **Alors** % si le RTT est faible

$T_{Spreading} = \frac{RTT}{n}$ % on utilise un espacement inférieur à T_{max}

Sinon % cas des grands délais ou des pertes de SYN-SYN/ACK

$T_{Spreading} = T_{max}$, % on utilise T_{max}

Fin Si

3. Après l'émission de l'IW, TCP continue de façon régulière.

Cette nouvelle version du mécanisme résout la plupart des problèmes soulevés par nos implantations et tests successifs. Ainsi, le nouveau calcul de $T_{Spreading}$ permet d'optimiser la performance dans les réseaux avec et sans congestion, pour la plupart des débits des goulots d'étranglements et des RTTs :

1. En effet, dans le cas d'un RTT faible, inférieur à $IW \times T_{max}$, $T_{Spreading}$ est choisi égal à $\frac{RTT}{IW}$ afin que la somme des temps d'espacement ne soit pas supérieure au RTT. Cela a des implications différentes selon le débit du goulot d'étranglement :

- Le débit du goulot d'étranglement est élevé et l'espacement demeure supérieur à $\frac{MSS}{débit}$: Initial Spreading continue à être très efficace en environnement congestionné malgré l'espacement faible, et est également très efficace dans un milieu sans congestion ; le délai additionnel est minimal.
- Le débit du goulot d'étranglement est faible et l'espacement est inférieur à $\frac{MSS}{débit}$: l'intérêt de l'Initial Spreading est limité par l'apparition de bursts. Toutefois, l'Initial Spreading reste au moins aussi efficace que la RFC 6928 puisque le burst produit par l'Initial Spreading a une taille inférieure ou égale à celui produit par la RFC 6928.
  Néanmoins, considérant le phénomène de "Bufferbloat" évoqué précédemment, il est peu probable que ce cas de figure se produise, le RTT mesuré ne pouvant être inférieur au temps d'émission de 10 segments par le routeur du goulot d'étranglement.

2. Dans le cas d'un RTT supérieur à $IW \times T_{max}$, $T_{Spreading}$ est choisi égal à T_{max}. Cela limite la latence ajoutée par l'Initial Spreading, améliore son efficacité dans les cas sans congestion et empêche les problèmes dus à une mesure de RTT erronée dans les cas congestionnés. L'Initial Spreading fait ainsi preuve d'une efficacité similaire à celle constatée lors nos simulations et expérimentations.

En conclusion, notre nouvelle proposition d'Initial Spreading semble résoudre les problèmes relevés tout au long de ce manuscrit. Dans le chapitre suivant, nous allons nous intéresser aux performances de l'Initial Spreading en le testant dans différents environnements réels.

Nous analyserons ainsi son comportement dans les réseaux filaires mais également dans les réseaux satellitaires, et tâcherons de vérifier les différentes hypothèses que nous avons faites dans ce dernier chapitre.

7 Évaluation de l'Initial Spreading en environnement réel

Dans le chapitre précédent, nous avons modifié l'algorithme de l'Initial Spreading et proposé une implantation qui exploite les dernières améliorations logicielles disponibles afin de tenir compte des contraintes réelles dites "à l'échelle du protocole" et "à l'échelle du réseau" (voir 6).

Nous allons maintenant pouvoir évaluer l'Initial Spreading dans un contexte réel et finalement valider les nombreux avantages que nos modèles analytiques et nos simulations ont laissé augurer.

7.1 Validation sur réseau filaire

Une des forces de l'Initial Spreading est de ne pas être une solution spécifique à un réseau donné mais bien une solution bout-en-bout efficace dans un contexte global. L'amélioration des communications pour tout type de lien, y compris le segment satellite, répond non seulement à notre problème de départ mais offre à notre solution une visibilité et une attractivité bien supérieures. S'il n'améliore pas les communications terrestres, un mécanisme bout-en-bout n'a d'ailleurs que peu de chances d'être déployé, et ce, quel que soit les bénéfices qu'il apporte aux communications par satellite.

Ainsi, avant même de nous soucier des performances de l'Initial Spreading dans un contexte satellitaire, il est essentiel de s'assurer que notre mécanisme est également performant dans les autres contextes, et notamment les réseaux filaires.

7.1.1 Résultats expérimentaux

La Figure 7.1 présente le banc de test que nous avons mis en œuvre pour nos expérimentations. Le trafic provoquant la congestion est composé de 8 connexions longues. Celles-ci induisent un taux d'occupation du buffer du goulot d'étranglement supérieur à 80% et un taux de perte de l'ordre de 7%.

La Figure 7.2 et la Figure 7.3 comparent les performances obtenues en utilisant la RFC 6928, l'Initial Spreading avec un $T_{Spreading}$ dynamique égal à $\frac{RTT}{10}$ et différentes valeurs de $T_{Spreading}$ statiques. Les délais utilisés pour le lien congestionné sont respectivement de 40ms et de 250ms.

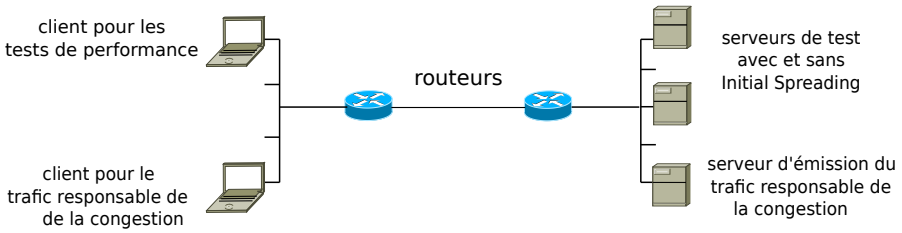

FIGURE 7.1 – Topologie du réseau retenue pour nos expérimentations

Chaque routeur a un débit d'émission de 10 Mb/s, alors que les serveurs et les clients ont un débit d'émission de 100 Mb/s.

Les deux figures présentent des résultats conformes à nos attentes et valident non seulement l'amélioration des performances obtenue grâce à l'utilisation de l'Initial Spreading, mais soulignent également les progrès réalisés par la dernière version de notre mécanisme.

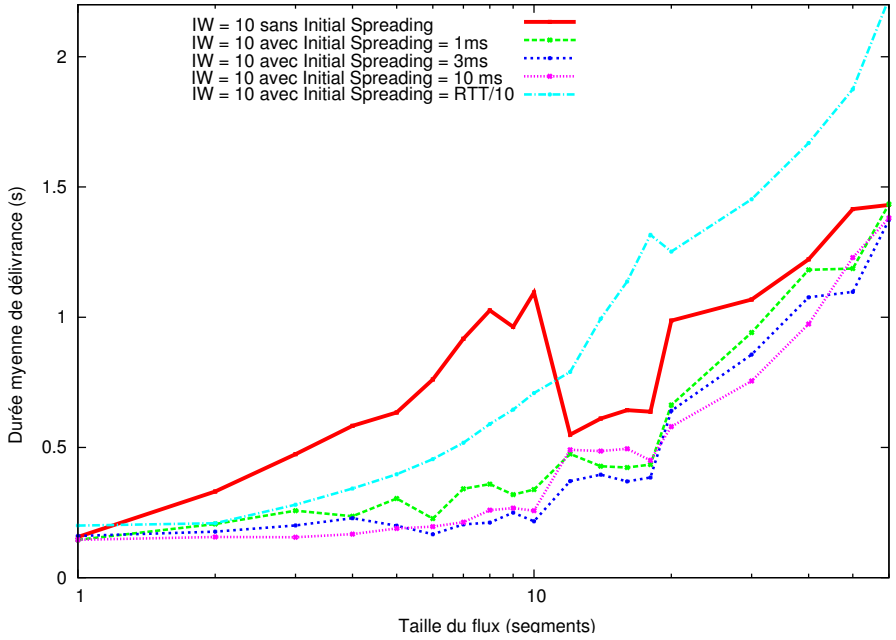

FIGURE 7.2 – Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 42ms

Validation des modifications apportées à l'Initial Spreading Le principal changement apporté au mécanisme de l'initial Spreading dans le chapitre précédent réside dans l'utilisation d'un espacement fixe. Ce dernier a été introduit afin de limiter la latence ajoutée par l'Initial Spreading dans les réseaux sans congestion mais également pour empêcher qu'une mesure erronée

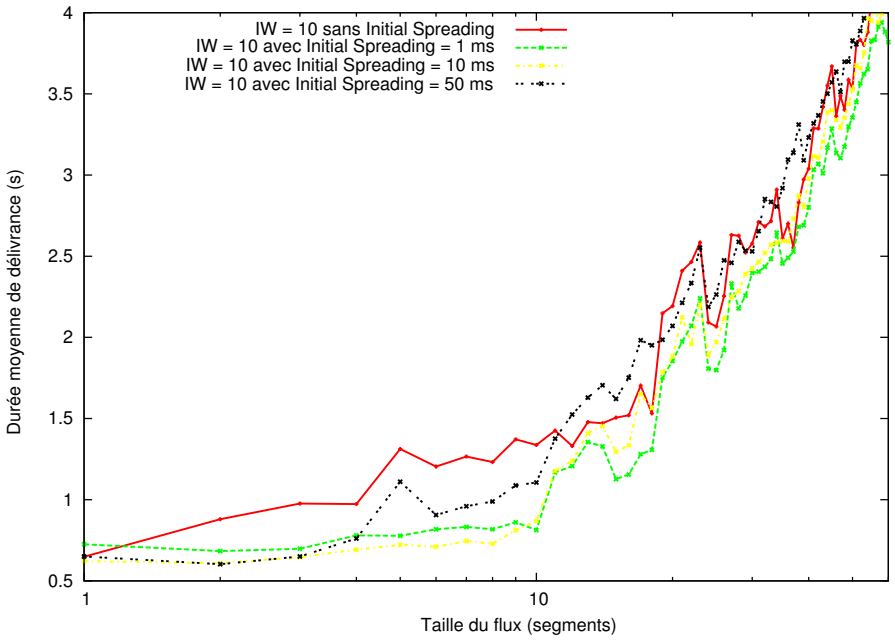

FIGURE 7.3 – Comparaison des performances de différentes versions d'Initial Spreading avec la RFC 6928 pour un délai de bout-en-bout de 252ms

du RTT puisse détériorer les performances.

La Figure 7.2 permet de vérifier l'efficacité de l'espacement fixe. Ainsi, alors que la version fondée sur un étalement proportionnel au RTT voit sa performance moyenne se dégrader avec la taille du flux, et ce même lors de la transmission de connexions courtes, l'espacement fixe permet à l'Initial Spreading de ne pas pâtir outre mesure de la congestion, et de conserver une durée de délivrance faible et quasiment constante pour les flux de taille inférieure à l'IW.

Par ailleurs, nous avons fait l'hypothèse en 6.2.3 que la durée minimale d'espacement pour conserver l'"indépendance" des segments est égale à T_B. Les deux courbes précédentes semblent confirmer notre hypothèse. En effet, le débit du goulot d'étranglement étant fixé à 10Mb/s, la valeur de T_B qui est égale au temps nécessaire à l'émission d'un segment par le goulot d'étranglement est donc de 1.2 ms (soit $\frac{MSS}{Débit}$). Or, on peut constater que les performances de l'Initial Spreading avec un espacement fixe sont moins bonnes lorsque $T_{Spreading}$, équivalent dans ce scénario à T_{max}, est inférieur à T_B.

On peut d'ailleurs noter que l'utilisation d'un espacement trop important (par exemple, $T_{max} = 10ms$ par rapport à $T_{max} = 4ms$) nuit moins aux performances que l'utilisation d'un espacement inférieur à T_B.

Validation de l'Initial Spreading Les deux courbes précédentes nous permettent également de constater des bénéfices importants engendrés par l'Initial Spreading. Ainsi, alors que la trans-

mission d'un burst de plus en plus grand sans Initial Spreading augmente fortement la durée de délivrance moyenne, confortant nos résultats obtenus par simulation (voir 4.3.2) et solution mathématique (voir 5.4.1), l'utilisation de l'Initial Spreading permet un gain de performance remarquable.

Par ailleurs, il est intéressant de noter que l'apport des nouveaux mécanismes cités dans le chapitre 3 ne devrait pas changer ce constat. En effet, si l'utilisation de *Tail Loss Probe* 3.1.2 et *Early Retransmit* 3.1.2, attendue dans un futur proche, devrait améliorer les performances des mécanismes sans Initial Spreading en optimisant la reprise sur erreur, celles-ci continueront à pâtir des bursts. Finalement, les performances avec Initial Spreading qui profiteront elles-aussi de ces mécanismes resteront donc nettement supérieures.

7.1.2 Initial Spreading : un outil de lutte contre le bufferbloat

L'augmentation de la taille du buffer a longtemps été considérée comme le procédé le plus efficace pour lutter contre la congestion et les taux de pertes élevés. Toutefois, cette solution qui diminue effectivement le nombre de pertes en permettant aux routeurs de conserver plus de données a pour contrepartie de rallonger le temps d'attente moyen passé par les segments dans les files d'attente rencontrées sur leur chemin. De nombreuses études sont actuellement menées pour permettre de lutter contre ce phénomène intitulé "bufferbloat" [63] (voir 6.3.1).

Les nouvelles versions de TCP dont nous avons parlé en 2.1.1.7 ont permis de nuancer l'impact négatif d'une perte en améliorant la capacité de réaction du protocole. La "communauté réseau" commence donc à cesser de craindre les pertes, intégrant le fait que TCP a besoin de celles-ci pour se comporter efficacement.

Pour autant, comme nous avons pu le voir dans les chapitres précédents, les pertes continuent à dégrader de façon significative les performances des connexions courtes, et la solution de facilité consistant à opter pour un surdimensionnement des buffers est régulièrement adoptée.

La Figure 7.4 permet de constater que l'Initial Spreading peut être utilisé pour réduire la capacité du buffer. Ainsi, si la taille du buffer continue de peser sur la latence moyenne, elle ne semble pas avoir de conséquences sur le comportement de l'Initial Spreading. Réduire la taille du buffer de 200 à 50 segments ne perturbe pas le bon fonctionnement de l'Initial Spreading, ce dernier semblant toujours en mesure de permettre une transmission quasi-optimale de la fenêtre initiale et donc de la connexion courte.

En conclusion, la nouvelle version de l'Initial Spreading est capable de s'accommoder des contraintes réelles. Le gain constaté semble encore meilleur que celui espéré suite à nos simulations et analyses mathématiques. L'Initial Spreading est donc en mesure de prendre le meilleur parti de l'augmentation de l'IW et de ne pas souffrir des bursts qui contrarient la performance de la RFC 6928.

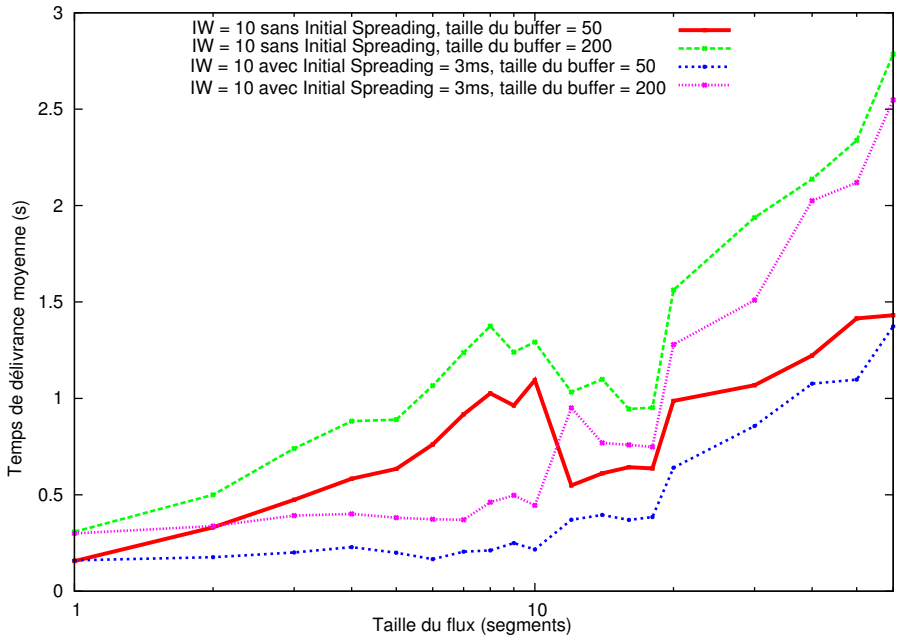

FIGURE 7.4 – Conséquences de la taille du buffer sur les performances avec et sans Initial Spreading

7.2 Validation sur réseau satellite

En offrant une amélioration significative des performances sur les réseaux filaires, l'Initial Spreading remplit une première partie de ses objectifs. En effet, la validation de l'Initial Spreading sur les réseaux filaires était une condition *sine qua non* pour que l'Initial Spreading ne soit pas juste perçu comme un mécanisme spécifique aux communications par satellite le vouant irrémédiablement à l'anonymat.

Pour autant, la validation de notre mécanisme sur les liens filaires n'est pas un gage de succès pour les communications par satellite. En effet, l'utilisation du lien satellite induit l'utilisation de protocoles différents au niveau du points d'accès des satellites, et notamment de leur couche MAC, qui peuvent avoir d'importantes répercussions sur la performance de l'Initial Spreading.

Jusqu'à présent, que ce soit dans nos simulations ou notre modèle analytique, nous n'avons jamais tenu compte des particularités de la couche MAC des liens traversés. Les expérimentations réelles sont donc indispensables pour statuer de l'intérêt de l'Initial Spreading dans ce type de réseau.

La Figure 7.3 qui illustre le comportement de l'Initial Spreading lorsque le RTT est long ($\sim 500ms$, soit un RTT proche de celui des liens satellite), ne tient en effet pas compte des autres spécificités du segment satellite, et ne peut donc pas être considérée comme un indicateur définitif de performance.

Ainsi, afin de valider l'Initial Spreading sur des réseaux satellites et comparer ses performances aux TCP-PEPs, nous allons utiliser une plateforme d'émulation des communications par satellite appelée Opensand [19] permettant une représentation fidèle du comportement du lien satellite [64], et des TCP-PEP distribués commerciallisés. Ces derniers nous garantissent un meilleur réalisme et une plus grande fiabilité que les émulateurs de TCP-PEPs tel que PEPSAL [65] communément utilisés par la communauté scientifique.

La Figure 7.5 décrit le banc de test utilisé pour analyser le comportement de l'Initial Spreading sur des réseaux hybrides satellites et terrestres.

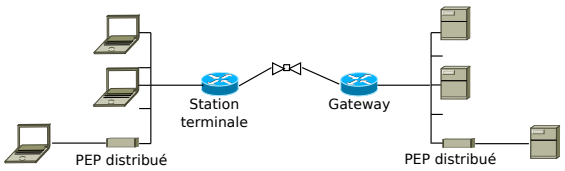

FIGURE 7.5 – Topologie du réseau employée pour nos expérimentations avec satellite

7.2.1 Impact de la couche MAC sur l'Initial Spreading

Les satellites de télécommunications européens fonctionnent essentiellement dans le cadre des standards DVB-S (Digital Video Broadcasting - Satellite). Dans le cadre d'un raccordement Internet, la couche MAC ressemble à celles utilisées dans les réseaux d'accès mobiles du type WIMAX ou LTE par exemple. On aura plusieurs niveaux de qualité de service. Sur la voie aller, on utilisera les standards DVB-S ou DVB-S2 et dans ce cadre, on aura essentiellement des problèmes d'ordonnancement. Sur la voie retour, on pourra utiliser le standard DVB-RCS et des mécanismes d'allocation de ressources, éventuellement dynamiques à l'aide de requêtes [66]. Dans Opensand, c'est le couple DVB-S2/RCS qui est implanté, seule la couche physique est émulée.

Les premières expériences effectuées avec Opensand ont confirmé l'impact que peut avoir la couche accès DVB sur les performances de l'Initial Spreading. Contrairement aux résultats obtenus lorsque le segment satellite était représenté par un lien filaire classique à délai élevé, l'utilisation de l'Initial Spreading n'a ainsi pas montré d'amélioration significative des performances lors de nos premiers tests avec Opensand.

La couche MAC est en charge de l'ordonnancement et de l'allocation des ressources. Sa situation entre les couches Transport et physique lui permet d'influencer les performances en agissant sur l'émission des segments. Ainsi, selon le type de réseau considéré, les besoins du système ou la qualité de service demandée, l'interaction de la couche MAC du point d'accès au goulot d'étranglement (routeur ou Gateway) avec l'Initial Spreading semble affecter différemment son efficacité.

Cas des réseaux filaires

Dans la plupart des cas, l'ordonnancement de niveau MAC n'est pas significatif et la couche liaison a des conséquences faibles sur le comportement de l'Initial Spreading. Les trames Ethernet sont émises en continu à un débit proche de leur débit d'arrivée et ne remplissent pas de façon significative le buffer de la couche liaison. Finalement, l'appréhension des bursts au niveau Réseau est suffisante pour "décorréler" les pertes des segments et permettre un maximum d'efficacité à l'Initial Spreading.

Les cas de la voie aller et de la voie retour sont à différencier lorsque l'on étudie les communications par satellite. L'ordonnancement et la méthode d'accès différent ainsi fortement au niveau de la couche MAC pour tenir compte des besoins et capacités de chacun.

Cas de la voie aller du lien satellite

Dans nos études nous avons considéré le standard DVB-S2 (Digital Video Broadcasting - Satellite) avec une couche liaison GSE (Generic Stream Encapsulation) qui permet l'utilisation de tous types d'architectures IP [67][66].

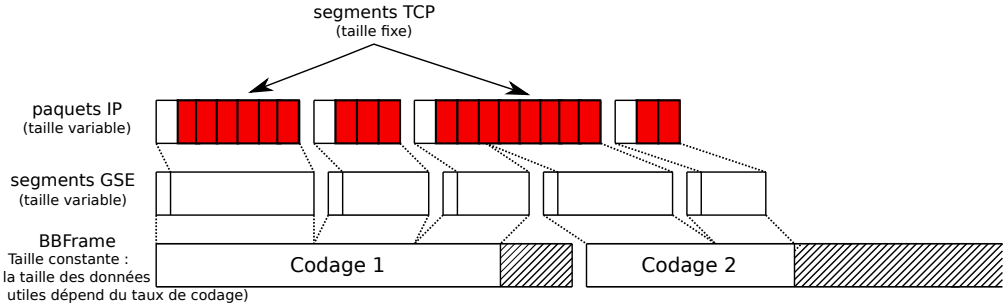

FIGURE 7.6 – Encapsulations successives des paquets IP

La Figure 7.6 illustre l'encapsulation de paquets IP dans des segments GSE puis dans des trames appelées Base Band Frames (BBFrame). Dans un premier temps, les paquets IP de taille variable sont encapsulés dans des segments GSE également de taille variable. L'encapsulation GSE fragmente également les paquets IP afin de s'adapter aux BBFrames sous jacentes. Une étape de concaténation et de multiplexage est réalisée puisqu'une même BBFrame pourra contenir des segments GSE émis vers différents terminaux sous réserve que l'on utilise les mêmes couples Modulation/Codage pour leur envoyer des informations (une procédure de déclassement est possible qui consistera à compléter des BBFrames par des segments GSE destinés à des terminaux ayant de meilleures conditions de réception et qui pourront alors décoder les BBFrames). Les techniques modernes de codage et la stabilité des canaux rendent le système Quasi Error Free (QEF).

Le standard DVB-S2 définit 2 tailles de BBFrames, les trames longues (64800 bits codés) qui

sont très majoritairement utilisées dans les systèmes de diffusion ou d'accès, et les trames courtes (16200 bits codés). Le nombre de bits utiles varie en fonction du taux de codage utilisé.

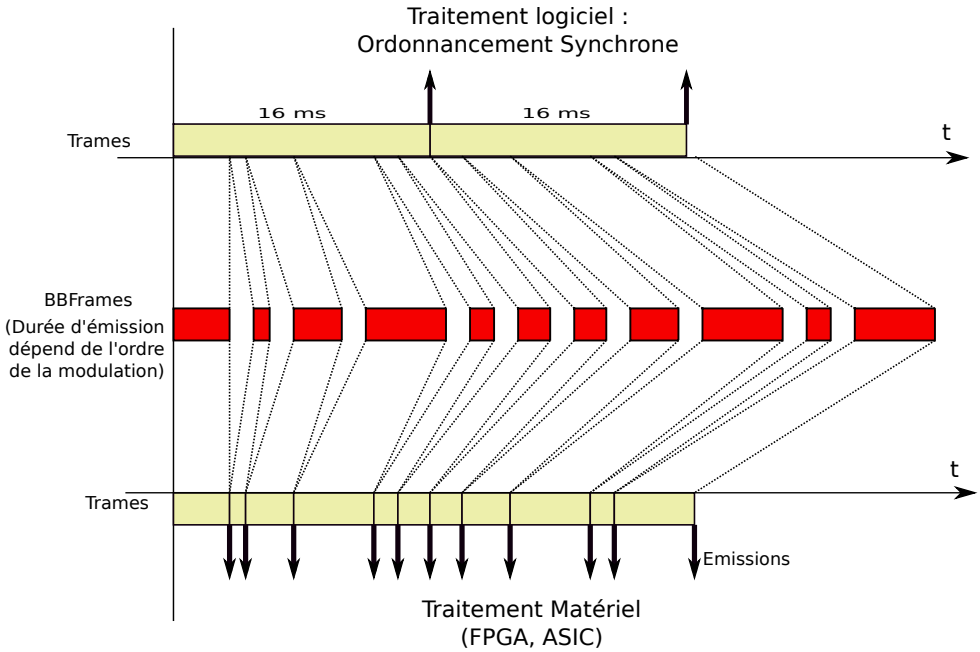

FIGURE 7.7 – Différents ordonnancements niveau MAC

L'ordonnancement de niveau MAC dépend fortement de la technologie utilisée. Celles-ci influeront donc réellement sur les performances de l'Initial Spreading. Nous allons détailler les deux différentes traitements de l'ordonnancement, illustrés par la Figure 7.7 et leurs potentiels impacts sur notre mécanisme :

1. Traitement logiciel : il s'agit du traitement le plus ancien, son utilisation est d'ailleurs vouée à disparaître dans les années à venir. A chaque période, il sélectionne différentes trames (modulation et codage) en fonction des conditions de propagation des utilisateurs, de leur besoin en trafic et de la qualité de service requise ([67]).

   Ce traitement résulte en un ordonnancement synchrone : le buffer de niveau liaison est vidé d'un coup à chaque période d'ordonnancement. Une quantité importante de segments TCP dépendant du codage de chaque BBframe mais aussi de la modulation utilisée (BPSK, QPSK, ...) et du débit symbole de la porteuse est donc retirée du buffer MAC à chaque période d'ordonnancement, ce qui se répercute sur le buffer de niveau IP. Par exemple, l'utilisation d'une valeur typique d'une période d'ordonnancement (16ms) dans nos émulations conduisait à vider d'un coup le buffer d'environ 13 segments TCP.

2. Traitement matériel (FPGA, ASIC) : ce traitement permet l'émission à la volée de chaque BBframe.

Bien que ces dernières puissent contenir plusieurs segments TCPs, le comportement des buffers est beaucoup plus proche de celui des segments terrestres. Les grands débits généralement disponibles (quelques dizaines de Mb/s) au niveau des gateways réduisent voire annulent les conséquences d'un tel ordonnancement,la période moyenne d'émission des BBframes étant en effet inférieure à T_{max}.

La Figure 7.7 retrace les deux types de fonctionnement. Les flèches noires identifient les instants de décision et de vidage des buffers. Les BBFrames ont une taille constante, mais les durées d'émission figurées sur le schéma vont dépendre de l'ordre de la modulation correspondante.

Dans la suite, nous analyserons le cas de l'ordonnancement synchrone qui reste le plus préoccupant.

La Figure 7.8 montre l'évolution des files d'attente de la couche liaison selon le type de réseau utilisé.

1. Une fois complet, le buffer du lien satellite va rejeter tous les segments arrivant, y compris les segments isolés, jusqu'à la prochaine période d'ordonnancement lors de laquelle l'envoi des données en attente d'émission a lieu.
2. Ensuite, le buffer accepte l'ensemble des segments arrivant, y compris les grands bursts, tant que la capacité de la file d'attente n'est pas atteinte de nouveau.

L'influence du buffer modifie donc la façon dont les pertes dues aux bursts sont "corrélées". Empêcher l'émission par l'émetteur de bursts au niveau TCP n'est donc plus suffisant pour garantir aux segments une meilleure probabilité de succès.

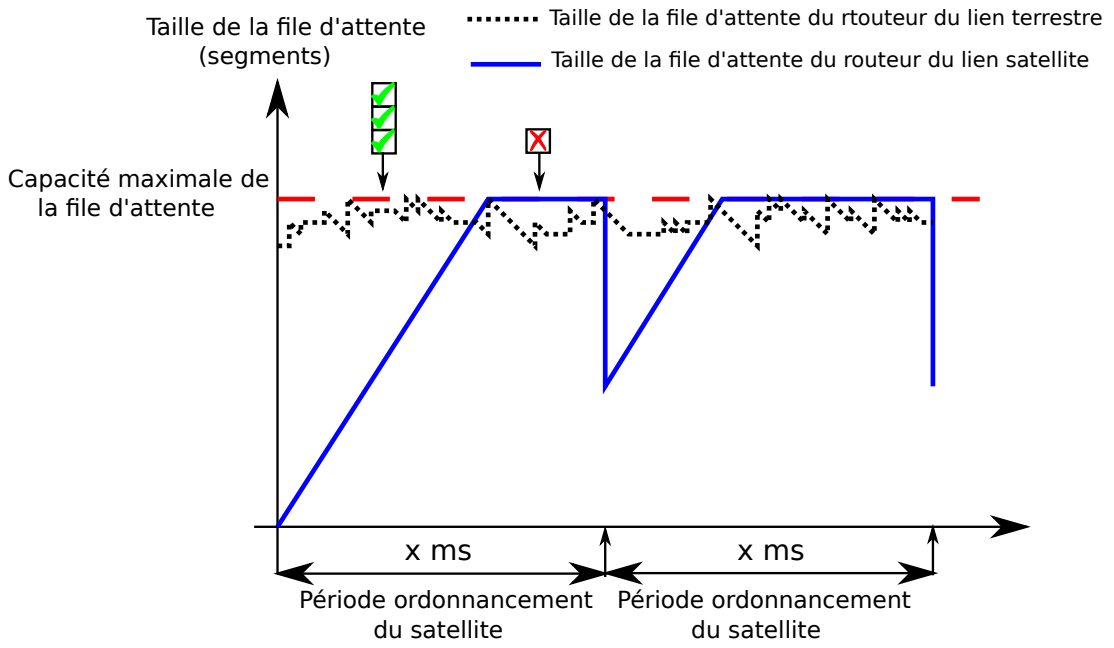

FIGURE 7.8 – Évolution de la file d'attente du routeur dans les cas d'un réseau filaire et satellite

Cas de la voie retour

Le cas de la voie retour est encore plus pénalisant pour les performances de l'Initial Spreading. En effet, les buffers niveaux MAC sont vidés à chaque trame logique, soit approximativement toutes les 25ms, afin de répondre au mieux aux différentes contraintes d'allocation et de gestion des ressources (principalement dans le cas d'un accès partagé MF-TDMA).

Toutefois, les bursts de segments sont majoritairement émis par les serveurs et donc sur la voie aller. La conséquence du tramage voie retour est donc moins importante que pour la voie aller. Ainsi, nous avons considéré préférable de privilégier l'appréhension de la voie aller à celle de la voie retour.

Dans la suite, les expériences sont donc réalisées sur la voie aller du segment satellite avec un ordonnancement périodique de 16ms. Nous considérons ce cas comme le plus compliqué à gérer par notre mécanisme pour des transmissions voie aller en raison de la taille importante de la période d'ordonnancement.

7.2.2 Configuration de l'Initial Spreading

Le comportement de l'Initial Spreading dans un contexte satellitaire n'est donc pas uniquement affecté par les buffers de niveau réseau du routeur du goulot d'étranglement, mais également par le comportement de ceux de sa couche liaison. Le paramètre le plus contraignant pour déterminer $T_{Spreading}$ cesse d'être le débit d'émission du goulot d'étranglement mais devient la période d'ordonnancement, cette dernière étant généralement d'une durée supérieure à T_B, qui représente le temps minimal nécessaire à l'Initial Spreading pour qu'il conserve son efficacité en fonction du débit du goulot d'étranglement.

De façon empirique, nous avons pu constater que le nombre de segments de l'IW émis par période d'ordonnancement influe sur l'efficacité de l'Initial Spreading. Ainsi, nos tests nous ont permis de privilégier l'émission d'un nombre maximal de segments émis par période d'ordonnancement inférieur ou égal à deux.

La Figure 7.9 compare les solutions avec et sans Initial Spreading avec différents $T_{Spreading}$ dans le cas d'une voie aller avec un $T_{Ordonnancement}$ égal à 16 ms. Dans cet environnement congestionné, on peut constater les conséquences fortes des variations de $T_{Spreading}$ sur l'Initial Spreading :

1. Les courbes avec $T_{Spreading} = 12ms$ et $T_{Spreading} = 50ms$ ont une allure semblable bien que le temps de délivrance moyen soit supérieur lorsque $T_{Spreading}$ est égal à $50ms$. Cela tend à valider la correspondance entre le nombre de segments émis par période d'ordonnancement et l'efficacité de l'Initial Spreading. Ainsi, avec une période d'ordonnancement de 16ms, un $T_{Spreading}$ de 12ms ou de 50ms a sensiblement la même conséquence sur le nombre de segments émis par période d'ordonnancement, le limitant à 1 ou 2 segments. Dans les deux cas, les segments émis ont donc une probabilité équivalente de succès et seule la latence

ajoutée par le fait que l'on n'émet pas de segment à chaque période d'ordonnancement dans le cas de $T_{Spreading} = 50ms$ semble différencier les performances.

2. La courbe avec $T_{Spreading} = 4ms$ montre les conséquences d'un étalement trop faible. Ainsi, la transmission des premiers segments pâtit-elle de l'ordonnancement et les résultats souffrent de la comparaison avec ceux sans Initial Spreading.

Finalement, nous utiliserons dans les tests suivants un $T_{Spreading} = 12ms$, sachant que la période d'ordonnancement est conservée égale à 16ms.

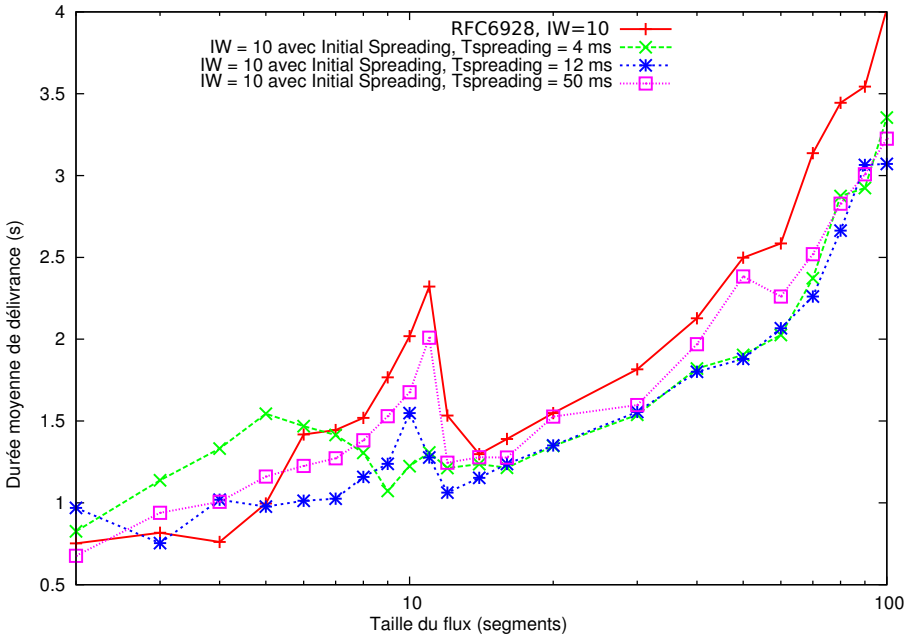

FIGURE 7.9 – Comparaison des temps moyens de délivrance pour des connexions courtes avec et sans Initial Spreading dans un réseau satellite ayant un délai de 280ms

7.2.3 Conclusion

S'assurer de l'efficacité de l'Initial Spreading dans un contexte satellitaire n'est pas aussi évident que dans un contexte filaire. En effet, la couche MAC et notamment l'interaction des buffers de niveau liaison sur l'émission des segments ajoutent de la complexité.

A l'aide de nos expériences, nous avons pu constater l'importance de la limitation du nombre de segments émis par période d'ordonnancement. Ainsi avec un $T_{Spreading}$ bien choisi, l'Initial Spreading permet des résultats significativement meilleurs que la RFC 6928. L'Initial Spreading est donc, même dans le contexte satellitaire, un mécanisme très efficace d'optimisation bout-en-bout de TCP.

Dans la suite, nous allons comparer les résultats obtenus avec TCP-PEP à ceux obtenus avec

Initial Spreading afin de déterminer si l'Initial Spreading peut s'affirmer comme une solution de bout-en-bout à même de remplacer ce type de solutions.

7.3 L'Initial Spreading face aux TCP-PEPs

7.3.1 Comparaison dans des environnements non-congestionnés

Dans le premier chapitre, nous avons mis en avant les progrès réalisés par les nouveaux TCPs ainsi que leurs limites dans le cadre des communications par satellite. Les nouveaux algorithmes de contrôle de congestion permettent désormais de rivaliser avec les TCP-PEPs dans le cas des connexions longues mais sont inefficaces dans les autres cas de figure.

La Figure 7.10 fait le point sur les améliorations permises par les mécanismes de démarrage rapide de TCP et compare la durée moyenne d'une connexion courte dans les cas d'utilisation suivants : TCP-PEP, la RFC 3390, la RFC 6928 et finalement l'Initial Spreading dans un environnement non congestionné.

Les résultats permettent de différencier 3 phases :

1. Cas des connexions courtes (1 à 10 segments) : la transmission d'une IW en un burst unique continue à être la solution la plus efficace. Comparativement à la longueur du RTT, le faible $T_{Spreading}$ utilisé permet à l'Initial Spreading d'obtenir des performances bien évidement meilleures que celles de la RFC3390 mais surtout proches de celles des deux autres mécanismes et notamment des TCP-PEPs.
2. Cas des tailles de connexions intermédiaires (10 à 1000 segments) : après la transmission de l'IW, les solutions TCP bout-en-bout, y compris l'Initial Spreading, doivent attendre les accusés de réception avant de pouvoir émettre de nouveaux segments, ce qui induit un temps supplémentaire d'un RTT. En revanche, en leurrant l'émetteur sur le destinataire final, les TCP-PEP peuvent continuer à émettre sans attendre un RTT, et ce tant que la connexion n'est pas sujette aux pertes.
3. Cas des connexions longues ($>$ 1000 segments) : les nouveaux algorithmes de congestion confirment leur efficacité. Ainsi, quel que soit le mécanisme de démarrage rapide utilisé, les nouvelles versions de TCP rivalisent avec les TCP-PEP.

Finalement, dans un environnement non congestionné, l'Initial Spreading et la RFC6928 ont des performances quasi-similaires à celles des TCP-PEP, sauf lors de la transmission de connexions de tailles intermédiaires, quand le grand RTT des connexions par satellite continue à dégrader les performances moyennes.

Toutefois, ce dernier cas de figure en faveur de l'utilisation de TCP-PEP demeure minoritaire et non critique. En effet, en dehors des connexions courtes qui représentent 90% des connexions totales, le reste du trafic correspond majoritairement à des échanges de données de très grande taille. De plus, l'état non congestionné n'est pas prépondérant dans les réseaux fondés sur le multiplexage statistique tels que le satellite.

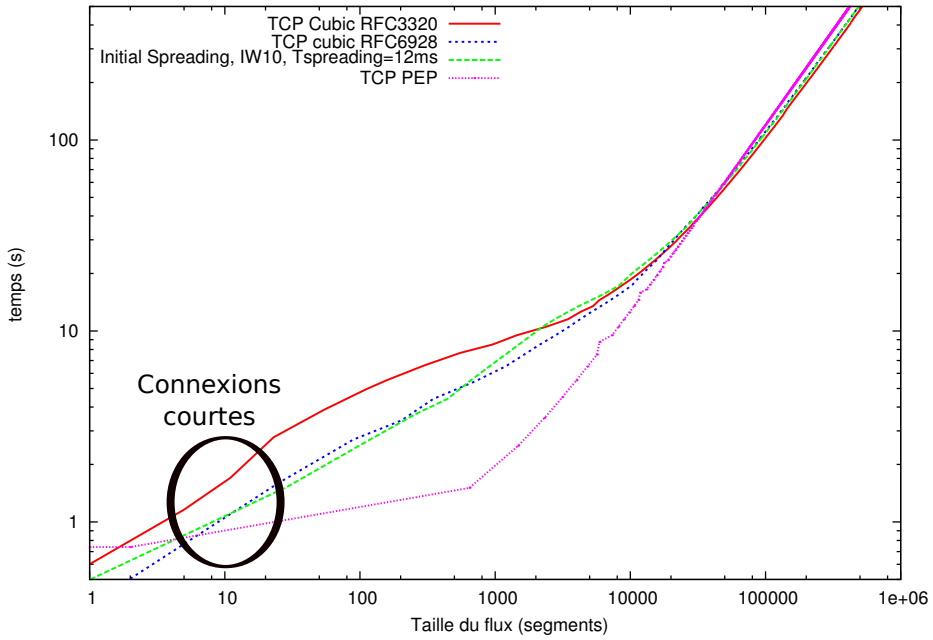

FIGURE 7.10 – Comparaison des 3 solutions existantes (TCP-PEP, RFC 3390, RFC 6928) avec l'Initial Spreading dans un réseau non congestionné

Dans la section suivante, nous présentons le comportement de ces mécanismes dans des environnements congestionnés afin d'évaluer notamment la capacité des différents mécanismes à gérer les retransmissions de segments.

7.3.2 Comparaison dans des environnements congestionnés

La Figure 7.11 compare les solutions bout-en-bout aux TCP-PEP dans les environnements congestionnés. La congestion est une fois de plus induite par l'utilisation d'un trafic concurrent de 6 flux de taille infinie partageant le même goulot d'étranglement.

Comme nous avons pu le voir dans les chapitres et sections précédents, l'utilisation d'une grande fenêtre initiale dans un environnement congestionné dégrade la performance moyenne de TCP. Les résultats montrent une nouvelle fois que dans un tel environnement, l'utilisation d'une IW de 3 ou 10 segments offre sensiblement la même efficacité pour les connexions courtes. L'Initial Spreading étant moins affecté par la congestion que les autres solutions bout-en-bout, il permet des résultats autrement meilleurs et conserve un très bon niveau de performance.

Ainsi, même si nos tests ont montré que les TCP-PEP gèrent de façon très efficace la congestion, l'Initial Spreading offre des résultats quasiment similaires. Les deux solutions surpassent largement les autres solutions présentées dans le cas des connexions courtes.

Par ailleurs, en ce qui concerne les connexions plus longues, l'Initial Spreading et les TCP-PEP ont la même faculté à gérer la congestion.

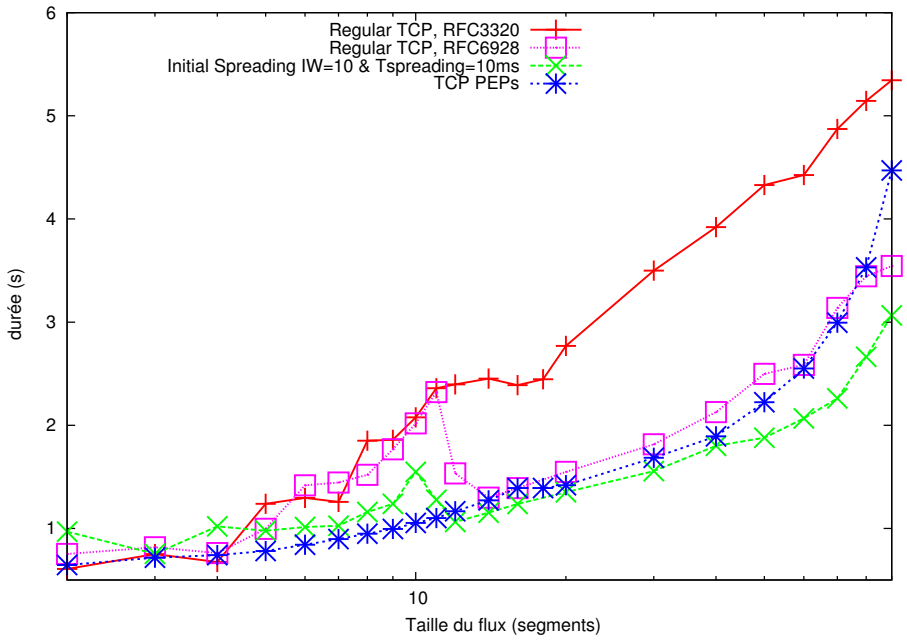

FIGURE 7.11 – Comparaison des 3 solutions existantes (TCP-PEP, RFC 3390, RFC 6928) avec l'Initial Spreading dans un réseau congestionné

En conclusion, l'Initial Spreading permet d'atteindre des performances proches de celles des TCP-PEP tout en conservant la sémantique bout-en-bout de TCP. Les légères améliorations de performances permises par les TCP-PEP, notamment vérifiables dans le cas sans congestion pour les connexions de taille intermédiaire, ne semblent plus suffisantes pour justifier leur utilisation systématique.

7.4 Quel $T_{Spreading}$ envisager?

Nous avons démontré la capacité de l'Initial Spreading à améliorer de façon significative la performance des flux courts dans deux contextes distincts : les réseaux filaires et les réseaux par satellite. L'étude comparative des comportements de l'Initial Spreading dans ces deux scénarios est désormais nécessaire pour configurer les paramètres de notre algorithme afin qu'il soit performant quel que soit le réseau traversé, et ce sans nécessiter une adaptation préalable.

En ce qui concerne les réseaux filaires, nos expérimentations ont confirmé les hypothèses concernant l'espacement minimal nécessaire pour garantir l'efficacité de l'Initial Spreading dans un environnement congestionné (T_B). La Table 6.1 nous donne quelques valeurs pour cet espacement en fonction du débit minimal considéré. Par exemple, un T_{max} de 4ms permet à l'Initial Spreading d'être efficace pour des débits supérieurs ou égaux à 6Mb/s et n'ajoute que peu de

latence lorsque le réseau n'est pas congestionné.

Notre étude des communications par satellite a quant à elle souligné que l'efficacité de l'Initial Spreading pour ce type de communication ne dépend pas seulement du débit du goulot d'étranglement mais également des interactions avec la couche MAC du lien satellite. Par exemple, pour des débits supérieurs ou égaux à 6Mb/s, la Figure 7.9 a montré qu'un T_{max} de 4ms n'est plus suffisant pour garantir une efficacité maximale à l'Initial Spreading. Ainsi, l'Initial Spreading nécessite un T_{max} plus élevé, de l'ordre de 10ms, pour contrer les effets de la politique d'ordonnancement du point d'accès satellite.

Notre solution a pour vocation d'améliorer les communications filaires **et** les communications par satellite. Elle se distingue des autres solutions en garantissant des performances très élevées dans la transmission des flux courts quel que soit l'environnement considéré, y compris en environnement congestionné. Conserver cette aptitude est primordiale, nous préconisons donc l'utilisation du plus faible T_{max} qui continue à permettre à l'Initial Spreading d'offrir de bonnes performances quel que soit le réseau rencontré et son encombrement, c'est-à-dire un T_{max} égal à 10ms.

Les conséquences du choix d'un tel T_{max} différent sensiblement selon la durée du RTT et l'encombrement du réseau :

1. Cas des RTTs longs. Considérant les topologies des réseaux modernes, un RTT long se justifie principalement par deux scénarios. L'impact de la longueur de T_{max} est différent pour chacun d'entre eux :
   - Le débit du goulot d'étranglement est faible, auquel cas la valeur minimale de l'espacement définit précédemment se rapproche des 10ms.
   - Le réseau est fortement congestionné, auquel cas la durée de l'espacement, qu'elle soit ou non supérieure à l'espacement minimal, n'affecte pas l'efficacité de l'Initial Spreading.
2. Cas des RTTs courts. Sauf perte d'un segment lors de l'établissement de la connexion, $T_{Spreading}$ n'est plus égal à T_{max} mais à $\frac{RTT}{IW}$. Ainsi, l'espacement est minimal et s'adapte à l'environnement. Si une perte survient durant la "poignée de main" initiale, $T_{Spreading}$ est égal à T_{max}, mais les conséquences d'un espacement important sont à relativiser par la congestion susceptible d'avoir causé la perte du segment.

En conclusion, le choix d'un T_{max} égal à 10ms permet de garantir l'efficacité de l'Initial Spreading quel que soit le type et l'encombrement du réseau. Ce T_{max} relativement conservatif pourra sans doute être revu à la baisse dans les années à venir en tenant compte des progrès technologiques réalisés sur les liens satellites. Pour autant, nous sommes conscients que cette gestion des contraintes relatives aux couches basses n'est pas parfaite. Notre préoccupation première portait sur le segment satellite et nous avons montré comment dans un tel contexte ajuster les paramètres de l'Initial Spreading. Des travaux complémentaires pourraient être menés, en particulier pour une meilleure interaction avec les couches basses, nous y reviendrons dans nos perspectives.

8 Conclusion et Perspectives

8.1 Conclusion

Tout au long de ce travail de recherche, nous avons pu constater l'influence des protocoles de transport sur la qualité d'une communication. Ainsi, que ce soit sur des liens satellites ou terrestres, le protocole de transport utilisé peut faire varier du simple au double la qualité moyenne de la performance.

Nous avons tout d'abord effectué ce constat dans un contexte satellitaire où les solutions de transport classiques ne semblent pas en mesure de faire face aux contraintes intrinséques et sont ainsi incapables de garantir des résultats satisfaisants. Ces dernières sont donc régulièrement remplacées ou complétées par des accélérateurs de performance de type TCP-PEP qui offrent une amélioration remarquable en coupant l'architecture bout-en-bout, au prix néanmoins d'une marginalisation du segment satellite. Cet isolement est coûteux car il risque d'empêcher la communauté satellite de prendre part aux efforts d'intégration des différents réseaux qui visent à utiliser de façon transparente un ensemble de technologies compatibles.

Une première série de tests et d'expériences nous a permis de constater que le manque d'efficacité des solutions de bout-en-bout classiques telles que TCP dans les communications par satellite s'étaient sensiblement réduit grâce à l'amélioration des algorithmes de contrôle de congestion des nouvelles versions de TCP et notamment Cubic et Compound. Ces dernières permettent en effet d'atteindre des performances quasiment similaires à celles des TCP-PEP dès que la taille du flux est importante en améliorant en particulier leur faculté à réagir après une perte et en réduisant leur dépendance au RTT. Ainsi, "seules" les connexions courtes et celles de taille intermédiaire continuent à pâtir de l'utilisation de solutions de bout-en-bout.

Ce constat ne s'applique pas uniquement au contexte satellitaire, mais s'étend à l'ensemble des communications. En effet, la performance des flux courts continue de dépendre principalement du protocole de slow start qui contrairement aux algorithmes d'évitement de congestion n'a pas évolué. Certains mécanismes lui ont toutefois été associés afin d'en améliorer la performance mais aucun d'entre eux n'a réussi à obtenir l'approbation générale, ayant tous en commun de n'être efficaces que sur certains types ou états particuliers de réseaux, et de dégrader la performance moyenne dans les autres configurations.

Or les flux courts, c'est-à-dire ceux dont la taille n'excède pas 15KB soit environ 10 seg-

ments TCP, représentent plus de 90% des connexions mises en œuvre. Apporter une solution à l'amélioration des performances des connexions courtes est donc un problème de tout premier ordre qui peut non seulement permettre une meilleure efficacité des communications en général mais également résoudre le problème des communications par satellite en limitant fortement l'assujettissement aux solutions spécifiques de type TCP-PEP.

Initial Spreading est la solution que nous avons proposée pour répondre à ce problème. Ce mécanisme de démarrage rapide des connexions TCP est fondé conjointement sur l'augmentation de la fenêtre Initiale et sur le mécanisme de Pacing. Il a fortement évolué au cours de nos expériences pour offrir des performances optimales quel que soit le type et l'état du réseau emprunté. Initial Spreading s'inscrit ainsi parfaitement dans la volonté annoncée de tendre vers un réseau de communication unique à performance élevée.

8.1.1 Poids des bursts sur la performance moyenne

Dans un premier temps, nous avons isolé les rafales (ou bursts) et démontré grâce à nombreuses simulations et une modélisation analytique précise des débuts de connexion TCP leurs répercussions sur la transmission d'une connexion. En effet, si la RFC 6928, qui préconise l'utilisation d'une IW de 10 segments afin d'émettre la totalité de 90% des connexions en un RTT, permet d'obtenir la meilleure performance possible pour une connexion seule, elle pâtit fortement de la concurrence avec d'autres flux. En environnement congestionné, le burst initial peut ainsi détériorer la performance à un point tel qu'une connexion utilisant une IW de seulement 3 segments est susceptible d'offrir de meilleures performances qu'une connexion avec une IW de 10 segments.

Pour autant, ces bursts demeurent nécessaires au fonctionnement de TCP, en permettant notamment un partage équitable du débit disponible entre les flux concurrents, ce qui exclut de fait une solution telle que le Pacing.

L'Initial Spreading est issu de la compréhension fine des bursts et de leurs répercussions.

8.1.2 Initial Spreading, une réponse aux faiblesses de TCP

La dernière version de l'Initial Spreading tient compte des différentes contraintes d'utilisation en environnement réel mises en lumière par nos implantations et expérimentations successives pour fournir une solution de bout-en-bout particulièrement efficace aux problèmes évoqués précédemment.

Ainsi, le nouvel algorithme de l'Initial Spreading lui permet de s'adapter aux caractéristiques du réseau traversé tout en gardant à l'esprit que celles-ci ne sont que partiellement connues et demeurent non-prévisibles. Cela lui permet notamment de se prémunir de potentielles pertes lors de l'établissement de la connexion, tandis que sa capacité d'adaptation lui permet de minimiser la latence ajoutée dans un environnement sans congestion tout en conservant son efficacité dans les autres cas de figure.

L'Initial Spreading présente les garanties suivantes :

1. Des performances notoirement supérieures à celles des autres mécanismes bout-en-bout quel que soit le type de réseau traversé et son encombrement :
   - Un gain de performance supérieur à 30% dans la transmission de flux courts en environnement congestionné par rapport aux mécanismes actuels ;
   - Des performances excellentes dans la transmission de flux courts dans un réseau sans congestion.
2. Des performances proches de celles des TCP-PEP dans les communications pas satellite. Les performances ainsi obtenues associées aux très nombreux avantages résultant de l'utilisation d'une solution bout-en-bout efficace sans distinction du réseau parcouru, remettent en cause l'utilisation inconditionnelle des solutions de type TCP-PEP.
3. Et finalement, une implantation aisée qui facilite et favorise son déploiement.

8.2 Perspectives

8.2.1 Perspectives à court terme

Nous avons évoqué dans les chapitres précédents la complexité de proposer et valider une optimisation d'un protocole tel que TCP qui voit sa performance dépendre de ses interactions avec d'autres protocoles de la pile mais également avec les connexions concurrentes. En proposant un large panel de simulations, modélisation analytique et expérimentations réelles, nous avons non seulement affiné notre mécanisme afin qu'il tienne compte des répercussions de ces nombreuses interactions, mais nous avons également assis la légitimité et la crédibilité de l'Initial Spreading.

Néanmoins, les scénarios que nous avons envisagés restent limités, par exemple par le nombre d'utilisateurs, de flux concurrents et le nombre de segments parcourus. Afin de s'assurer de l'efficacité de l'Initial Spreading dans l'ensemble des scénarios possibles, il serait particulièrement intéressant de réaliser des tests à grande échelle semblables à ceux menés par Google pour légitimer l'augmentation de l'IW [35].

Par ailleurs, nous avons mis en avant le caractère peu intrusif de l'Initial Spreading, suggérant que cette faculté garantissait la pérennité de son utilisation. L'évaluation des performances de l'Initial Spreading pourrait être menée dans le cadre d'une utilisation conjointe avec les optimisations protocolaires présentées dans le chapitre 2.

De plus, une étude complémentaire sur l'influence de la couche MAC sur l'efficacité de l'Initial Spreading serait également intéressante pour affiner les valeurs des paramètres tels que T_{max} qui permet de fixer la limite haute de la durée d'espacement. Il serait également intéressant de tester notre mécanisme dans le cas d'utilisateurs terminaux connectés à d'autres types de technologies telles que le WiFi. Les couches basses du protocole 802.11 risquent en effet de

modifier le comportement et l'efficacité de l'Initial Spreading, et il est possible qu'il faille modifier le T_{max} proposé.

Finalement, forts des derniers résultats que nous avons eus avec l'Initial Spreading, nous allons pouvoir modifier notre "Internet Draft" et nous rapprocher de sa standardisation.

8.2.2 Perspectives à plus long terme

La prise en compte à titre individuel de chacune des couches basses pouvant affecter le comportement de l'Initial Spreading n'est pas une solution aisée ni une solution définitive. En effet, ces dernières vont continuer à évoluer et de nouvelles études devront alors être menées afin que l'Initial Spreading conserve un maximum d'efficacité.

Une solution qui nous semble très prometteuse pourrait ainsi consister à modifier la relation entre le buffer IP et la couche MAC en utilisant les perspectives exhaltantes qu'offrent les nouvelles techniques de gestion actives des files d'attentes mais également l'association FQ/Pacing décrite au chapitre 6. On pourrait par exemple lisser le comportement des segments en sortie du bloc constitué de la couche réseau et de la couche MAC, ce qui permettrait à l'Initial Spreading de ne plus avoir à se soucier des comportements des couches basses.

Par ailleurs, ces nouvelles techniques permettent également d'envisager la généralisation de l'utilisation de l'Initial Spreading à l'ensemble de la connexion. Ainsi, en ayant la possibilité d'appliquer des traitements différents pour chaque flux mais également pour chaque segment d'un même flux, il semble envisageable de trouver des algorithmes capables de limiter les conséquences négatives des bursts tout en ne tombant pas dans les travers du Pacing.

Finalement, nous n'avons traité jusqu'à présent que des possibilités relatives à l'utilisation des ces nouveaux mécanismes au niveau des entités d'extrémité de la connexion, c'est-à-dire en prenant garde de ne considérer que la configuration la plus réaliste, car la plus aisément déployable. Néanmoins, leur utilisation généralisée à l'ensemble des routeurs et combinée avec un espacement adéquat permettant l'appréhension des bursts, peut offrir de formidables possibilités en termes de qualité de service, permettant notamment la différenciation des flux en fonction de leur taille, de leurs destinations, ...

Liste des communications

Conférences internationales avec comité de lecture

[1] R. Sallantin, E. Chaput, E. P. Dubois, C. Baudoin, F. Arnal, and A.-L. Beylot, 'On the sustainability of PEPs for satellite Internet access", in *30th AIAA International Communications Satellite System Conference (ICSSC)*, 2012, AIAA, pp. 1–8.

[2] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A.-L. Beylot, "Initial spreading: A fast start-up tcp mechanism", in *2013 IEEE 38th Conference on Local Computer Networks, LCN 2013*, 2013, pp. 492–499.

[3] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A.-L. Beylot, "A tcp model for short-lived flows to validate initial spreading", in *IEEE 39th Conference on Local Computer Networks, LCN 2014*, 2014, à paraître.

Contributions aux instances de standardisation

[1] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A. Beylot, "Safe increase of the tcp's initial window using initial spreading", Working Draft, IETF, Internet-Draft draft-irtf-iccrg-sallantin-initial-spreading, Sep. 2014.

Articles Soumis

[1] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A.-L. Beylot, "A TCP Model for Short-Lived Flows to Validate Initial Spreading", soumis à *International Journal on Satellite Communications and Networking*, Wiley.

[2] R. Sallantin, et al. *, "MUSE - Mission to the Uranian System: Unveiling the evolution and formation of ice giants", soumis à *Advances in Space Research*, Elsevier, Minor Revision

* Ce travail a été réalisé dans le cadre d'une école d'été de l'Agence Spatiale Européenne (Summer School of Alpbach) avec 14 autres doctorants européens (3 semaines réparties entre Août et Novembre 2012).

Bibliographie

[1] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the performance of tcp pacing," in *IEEE Conference on Computer Communications, INFOCOM 2000, IEEE*, vol. 3, 2000, pp. 1157–1165 vol.3.

[2] R. Sallantin, E. Chaput, E. P. Dubois, C. Baudoin, F. Arnal, and A.-L. Beylot, "On the sustainability of PEPs for satellite Internet access," in *30th AIAA International Communications Satellite System Conference (ICSSC)*. AIAA, 2012.

[3] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A.-L. Beylot, "Initial spreading: A fast start-up tcp mechanism," in *2013 IEEE 38th Conference on Local Computer Networks, LCN 2013*, 2013, pp. 492–499.

[4] ——, "A tcp model for short-lived flows to validate initial spreading," in *IEEE 39th Conference on Local Computer Networks, LCN 2014*, 2014, à paraître.

[5] R. Sallantin, C. Baudoin, E. Chaput, F. Arnal, E. Dubois, and A. Beylot, "Safe increase of the tcp's initial window using initial spreading," Working Draft, IETF, Internet-Draft draft-irtf-iccrg-sallantin-initial-spreading, Sep. 2014.

[6] G. Fairhurst, A. Sathiaseelan, H. S. Cruickshank, and C. Baudoin, "Transport challenges facing a nextgeneration hybrid satellite internet." *International Journal on Satellite Communications and Networking*, vol. 29, no. 3, pp. 249–268, 2011.

[7] A. Allman and S. Floyd, "Increasing tcp's initial window," RFC 3390, IETF, Proposed Standard, Oct. 2002.

[8] V. Paxson and A. Allman, "Computing tcp's retransmission timer," RFC 2988, IETF, Proposed Standard, Nov. 2000.

[9] S. Floyd and T. Henderson, "The newreno modification to tcp's fast recovery algorithm," RFC 2582, IETF, Proposed Standard, Apr. 1999.

[10] S. Floyd, T. Henderson, and A. Gurtov, "The newreno modification to tcp's fast recovery algorithm," RFC 3782, IETF, Proposed Standard, Apr. 2004.

[11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "Tcp selective acknowledgment options," RFC 2018, IETF, Experimental, Oct. 1996.

[12] V. Jacobson, "Congestion avoidance and control," in *ACM Conference of the Special Interest Group on Communications, SIGCOMM 1988.* ACM, 1988, pp. 314–329.

[13] ——, "Modified tcp congestion avoidance algorithm," *end2end-interest mailing list*, 1990.

[14] M. Allman, V. Paxson, and W. Stevens, "Tcp congestion control," RFC 2581, IETF, Proposed Standard, Apr. 1999.

[15] M. Allman, V. Paxson, and E. Blanton, "Tcp congestion control," RFC 5681, IETF, Proposed Standard, Sep. 2009.

[16] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "Compound TCP: A Scalable and TCP-friendly Congestion Control for High-speed Networks," in *4th International workshop on Protocols for Fast Long-Distance Networks, PFLDNet*, 2006.

[17] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *SIGOPS Operating System Review.*

[18] C. Caini and R. Firrincieli, "Tcp hybla: a tcp enhancement for heterogeneous networks," *International Journal Satellite on Communications and Networking*, vol. 22, 2004.

[19] Opensand: A satellite telecommunication system emulation platform. [Online]. Available: http://opensand.org/

[20] D. Lin and H. T. Kung, "Tcp fast recovery strategies:analysis and improvements," in *IEEE Conference on Computer Communications, INFOCOM '98. IEEE*, vol. 1, 1998, pp. 263–271 vol.1.

[21] C. Caini and R. Firrincieli, "End-to-end tcp enhancements performance on satellite links," in *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, 2006, pp. 1031–1036.

[22] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-Speed and Long Distance Networks," in *IEEE International Conference on Computer Communications, INFOCOM 2006, IEEE*, 2006, pp. 1 –12.

[23] S. Osada, T. Yokohira, W. Hui, K. Okayama, and N. Yamai, "Performance improvement of tcp using performance enhancing proxies - effect of premature ack transmission timing on throughput -," in *IEEE 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, 2005. IEEE*, 2005, pp. 7–12.

[24] S. Osada, W. Hui, T. Yokohira, Y. Fukushima, K. Okayama, and N. Yamai, "Throughput optimization in tcp with a performance enhancing proxy," in *ICCT '06. International Conference on Communication Technology, 2006*, 2006, pp. 1–6.

[25] Y. Nishida, W. Hui, H. Matsumoto, T. Yokohira, and Y. Fukushima, "Retransmission control in tcp with a performance enhancing proxy," in *10th International Conference on Advanced Communication Technology, 2008. ICACT 2008*, 2008, pp. 1881–1886.

[26] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 2002*. ACM, 2002, pp. 89–102.

[27] F. Farooqui, "Dynamic improvements for intelligent performance enhancement proxy," in *IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC 2010, IEEE*, 2010, pp. 2494–2498.

[28] E. Dubois, J. Fasson, C. Donny, and E. Chaput, "Enhancing tcp based communications in mobile satellite scenarios: Tcp peps issues and solutions," in *5th IEEE Advanced satellite multimedia systems conference (asma) and the 11th IEEE signal processing for space communications workshop (spsc)*, 2010, pp. 476–483.

[29] G. Ciccarese, M. De Blas, L. Patrono, P. Marra, and G. Tomasicchio, "An ipsec-aware tcp pep for integrated mobile satellite networks," in *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2004. IEEE*, vol. 4, 2004, pp. 2362–2366 Vol.4.

[30] D. Isci, F. Alagoz, and M. Caglayan, "Ipsec over satellite links: a new flow identification method," in *2006 IEEE International Symposium on Computer Networks. IEEE*, 2006, pp. 140–145.

[31] F. Arnal, T. Gayraud, C. Baudoin, and B. Jacquemin, "Ip mobility and its impact on satellite networking," in *Advanced Satellite Mobile Systems, 2008. ASMS 2008. 4th*, Aug 2008, pp. 94–99.

[32] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda, "Is it still possible to extend tcp?" in *Conference on Internet Measurement Conference, ACM SIGCOMM 2011*. ACM, 2011, pp. 181–194.

[33] F. Peng, A. Cardona, K. Shafiee, and V. Leung, "Tcp performance evaluation over geo and leo satellite links between performance enhancement proxies," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012, pp. 1–5.

[34] S. Ramachandran, "Web metrics: Size and number of resources." *http://code.google.com/speed/articles/web-metrics.html*.

[35] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, and N. Sutin, "An Argument for Increasing TCP's Initial Congestion Window," *SIGCOMM Computer Communication Review, ACM*, vol. 40, no. 3, 2010.

[36] The need for speed. [Online]. Available: http://www.technologyreview.com/files/54902/GoogleSpeed_charts.pdf

[37] "Transmission control protocol," RFC 793, IETF, Experimental, Jan. 1981.

[38] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan, "Tcp fast open," in *Seventh COnference on Emerging Networking EXperiments and Technologies, CoNEXT 2011*. ACM, 2011, pp. 21:1–21:12.

[39] Y. Cheng, J. Chu, S. Radhakrishnan, and A. Jain, "TCP Fast Open," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-tcpm-fastopen-02, Oct. 2012.

[40] V. Paxson, M. Allman, C. J, and M. Sargent, "Computing tcp's retransmission timer," RFC 6298, IETF, Proposed Standard, Jun. 2011.

[41] M. Allman, K. Avrachenkov, U. Ayesta, J. Blanton, and P. Hurtig, "Early retransmit for tcp and stream control transmission protocol (sctp)," RFC 5827, IETF, Proposed Standard, Apr. 2010.

[42] N. Dukkipati, N. Cardwell, Y. Cheng, and M. Mathis, "Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail Losses," Working Draft, Tech. Rep., 2013.

[43] M. Scharf, "Performance Evaluation of Fast Startup Congestion Control Schemes," in *IFIP Conference on Networking, NETWORKING 2009,Springer, Lecture Notes in Computer Science, LNCS*, 2009, vol. 5550, pp. 716–727.

[44] ——, "Comparison of end-to-end and network-supported fast startup congestion control schemes," *Computer Networks*, vol. 55, no. 8, pp. 1921–1940, 2011.

[45] A. Allman and S. Floyd, "Quick-start for tcp and ip," RFC 4782, IETF, Proposed Standard, Jan. 2007.

[46] D. Liu, M. Allman, S. Jin, and L. Wang, "Congestion control without a startup phase," in *International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports, PFLDnet 2007*, 2007.

[47] M. Allman, F. S, and C. Partridge, "Increasing tcp's initial window," RFC 2414, IETF, Proposed Standard, Sep. 1998.

[48] U. Ayesta and K. Avrachenkov, "The effect of the initial window size and limited transmit algorithm on the transient behavior of tcp transfers," in *15th ITC Internet Specialist Seminar*, 2002.

[49] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, "Increasing tcp's initial window," RFC 6928, IETF, Experimental, Jan. 2013.

[50] J. Gettys, "Iw10 considered harmful," *Internet draft*, 2011.

[51] V. Padmanabhan and R. Katz, "Tcp fast start: A technique for speeding up web transfers," in *IEEE GLOBECOM Internet MiniConference, IEEE 1998, pp. 41-46*, 1998.

[52] J. Mathis, M.and Semke, J. Madhavi, and K. Lahey, "The rate-halving algorithm for tcp congestion control," *Internet draft*, 1999.

[53] V. Visweswaraiah and J. Heidemann, "Improving restart of idle tcp connections," *Technical Report TR97-661, University of Southern California*, 1997.

[54] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, 1999.

[55] N. Cardwell, S. Savage, and T. Anderson, "Modeling tcp latency," in *IEEE Conference on Computer Communications, INFOCOM 2000. IEEE*, vol. 3, 2000, pp. 1742–1751 vol.3.

[56] B. Sikdar, S. Kalyanaraman, and K. Vastola, "Analytic models for the latency and steady-state throughput of tcp tahoe, reno, and sack," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 959–971, 2003.

[57] M. Mellia and H. Zhang, "Tcp model for short lived flows," *IEEE Communications Letters, IEEE*, vol. 6, no. 2, pp. 85–87, 2002.

[58] U. Ayesta, K. Avrachenkov, E. Altman, C. Barakat, and D. P, "Multilevel approach for modeling short tcp sessions," INRIA technical report, Tech. Rep. 4705, 2003.

[59] K. Zhou, K. Yeung, and V.-K. Li, "On bursty packet loss model for tcp performance analysis," in *2005 Workshop on High Performance Switching and Routing, 2005. HPSR.*, 2005, pp. 292–296.

[60] P. Dirnopoulos, P. Zeephongsekul, and Z. Tari, "Modeling the burstiness of tcp," in *12th IEEE Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. MASCOTS 2004 IEEE*, 2004, pp. 175–183.

[61] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–145, 2000.

[62] J. Nagle, "Congestion control in ip/tcp internetworks," RFC 896, IETF, Experimental, 1984.

[63] The bufferbloat projects. [Online]. Available: http://www.bufferbloat.net/

[64] C. Baudoin and F. Arnal, "Overview of platine emulation testbed and its utilization to support dvb-rcs/s2 evolutions," in *2010 5th Advanced satellite multimedia systems conference (ASMA) and the 11th signal processing for space communications workshop (SPSC), IEEE*, 2010, pp. 286–293.

[65] C. Caini, R. Firrincieli, and D. Lacamera, "Pepsal: a performance enhancing proxy designed for tcp satellite connections," in *63rd Vehicular Technology Conference. VTC 2006-Spring. IEEE*, vol. 6, 2006, pp. 2607–2611.

[66] E. Chaput, A. L. Beylot, and C. Baudoin, "Packet scheduling over dvb-s2 through gse encapsulation," in *IEEE Global Telecommunications Conference, 2008. GLOBECOM 2008. IEEE*, 2008, pp. 1–5.

[67] J. B. Dupe, E. Chaput, C. Baudoin, C. BES, A. Deramecourt, and A. L. Beylot, "Scheduling over dvb-s2," in *IEEE Global Telecommunications Conference, 2014. GLOBECOM 2008. IEEE*, 2014, à paraître.

Résumé

Dans cette thèse, nous proposons un mécanisme appelé Initial Spreading qui permet une optimisation remarquable des performances de TCP pour les connexions de petites tailles, représentant plus de 90% des connexions échangées dans l'Internet.

Cette solution est d'autant plus intéressante que pour certaines technologies telles qu'un lien satellite, le temps d'aller retour particulièrement long est très pénalisant, et des solutions spécifiques ont du être implantées qui empêchent l'intégration du satellite dans un système de communication plus large.

Nous montrons que l'Initial Spreading est non seulement plus performant, mais surtout plus général car pertinent dans toutes les situations. De plus, peu intrusif, il ne compromet aucune des évolutions de TCP passées ou à venir.

Mots clés : TCP; Satellite ; Optimisation ; Performance; Congestion

Abstract

In this Ph.D. Thesis, we propose a mechanism called Initial Spreading that significantly improves the TCP short-lived connexions performance, and so more than 90% of the Internet connections.

Indeed, if regular TCP without our mechanism can be considered as efficient for terrestrial networks, its behavior is strongly damaged by the long delay of a satellite communication. Satellite community developed then some satellite specific solutions that provide good performance, but prevent the joint use of satellite and other technologies.

We show therefore that Initial Spreading is not only more efficient than regular solutions but enables also the use of an unique protocol whatever the context. Moreover, being non-intrusive, it is suitable for past and future TCP evolutions.

Keywords: TCP; Satellite ; Optimisation ; Performance; Congestion