THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National Polytechnique de Toulouse (Toulouse INP)*

Présentée et soutenue le *04/05/2022* **par :**

Amal Boubaker

Performances des Protocoles de Transport dans les Constellations de Satellites

---

JURY

M. Michel Marot	IMT, Télécom Sud-Paris	Rapporteur
M. Guillaume Urvoy-Keller	Université Côte d'Azur	Rapporteur
Mme Christelle Caillouet	Université Côte d'Azur	Examinatrice
Mme Géraldine Texier	IMT Atlantique	Présidente
M. Nicolas Kuhn	Thales Alenia Space (TAS)	Examinateur
M. Renaud Sallantin	Thales Alenia Space (TAS)	Examinateur
M. Emmanuel Chaput	Toulouse INP, ENSEEIHT	Examinateur
M. André-Luc Beylot	Toulouse INP, ENSEEIHT	Examinateur

École doctorale et spécialité :
MITT : Domaine Informatique
Unité de Recherche :
IRIT
Directeur(s) de Thèse :
M. Emmanuel Chaput et *M. André-Luc Beylot*
Rapporteurs :
M. Michel Marot et *M. Guillaume Urvoy-Keller*

REMERCIEMENTS

Quelqu'un m'a dit (et qui se reconnaîtra) que je devrais garder, dans mon manuscrit, cette citation qu'on trouve dans le template original de manuscrit de thèse, ironique certes, mais la voici :

«*Faut jamais se laisser abattre*» – John F. Kennedy

Je voudrais commencer par adresser mes remerciements les plus chaleureux à mon directeur, mon co-directeur et co-encadrants de thèse, respectivement, Emmanuel Chaput, André-Luc Beylot, Cédric Baudoin, Jean-Baptiste Dupé, Nicolas Kuhn et Renaud Sallantin (dans un ordre purement alphabétique), pour leur soutien inconditionnel, leur implication sans faille et leur aide précieuse. Je leur sais gré de m'avoir encouragée tout le long de ces années de thèse. Je leur dois énormément beaucoup d'être investis et tout particulièrement encore plus durant les moments les plus difficiles sur le plan scientifique et humain. Je tiens également à leur exprimer ma reconnaissance pour les moments de taquinerie (oui parce que les moments de joie ça existe aussi dans une thèse), de débats et de partages culturels surtout avec mes directeurs de thèse.

Je remercie ensuite les deux rapporteurs, M. Michel Marot et M. Guillaume Urvoy-Keller pour leur relecture minutieuse du manuscrit et leurs retours constructifs. Je remercie également Mme. Géraldine Texier d'avoir accepté de présider le jury, ainsi que les membres du jury pour l'intérêt qu'ils ont porté à ma thèse et pour les échanges agréables pendant la soutenance.

Je tiens également à remercier Corinne Mailhes, notre maman TéSA adorée, de m'avoir aidée durant ces années de thèse et de m'avoir épaulée et d'avoir été à l'écoute à tous mes passages à TéSA même à l'improviste.

Je souhaiterais également remercier les permanents de l'équipe RMESS que j'ai pu côtoyer tout au long de ma thèse au laboratoire IRIT. Riadh Dhaou et sa famille (Salma et les enfants Aïcha et Yassine), je les remercie de m'avoir apporté tant sur le plan professionnel que sur l'aspect humain. Je me suis toujours sentie chez moi et si proche de ma Tunisie grâce à votre hospitalité et votre soutien, merci pour tout! Je tiens à remercier Béatrice Paillassa pour tous ces échanges si agréables et ses conseils précieux. Je voudrais également remercier Katia Jaffrès-Runser de m'avoir fait confiance sur des enseignements dans les moments les plus délicats de la thèse. Je tiens également à remercier Gentian Jakllari pour son sens de l'humour et une pique sympathique "with an american touch".

Je voudrais remercier particulièrement le secrétariat de l'équipe RMESS : Sylvie Armengaud Metche (plus connue sous le nom de SAM), Vanessa Adjeroud et le secrétariat de TéSA : Isabelle Vasseur, pour leur gentillesse, leur accueil chaleureux et leur encouragement tout le long de ces années de thèse.

Je tiens à remercier les ami·e·s de l'équipe RMESS du labo d'avoir fait ces années de thèse moins dures, je garderai de bons souvenirs de nos repas, sorties et tous ces moments partagés ensemble.

Je voudrais en premier lieu remercier mon co-bureau, Firmin Leroy Kateu Demlabim (plus connu sous son pseudonyme Jojo Boy) avec lequel j'ai tant partagé et passé le plus temps durant ces années de thèse. Merci pour tous ces agréables moments passés dans le fameux bureau F407. Je garderai de très bons souvenirs de nos échanges et débats de tout et de rien, de m'avoir initiée au poker (when in Vegas.. On saura où me trouver) et de notre quart d'heure philosophique quotidien, du premier concert auquel j'assiste et il ne s'agit pas de n'importe lequel : Kyo, s'il te plaît. Merci Firmin d'avoir été là pour moi et pour tous ces moments de complicité (très souvent, autour un kebab à ton vendeur préféré) !

Je voudrais remercier les ami·e·s du labo qui ont joué un grand rôle par leur amitié et qui ont été très présents pendant la dernière ligne droite : Oana, Kevin, Adrien, Aakash, Chaima, Daniel, Romain, Justin, Mohamed, (San) Francisco, Youssouf et toutes/tous celles/ceux que j'oublie.

Je tiens à remercier mon amie Selma Zamoum que j'ai côtoyée au labo TéSA et avec laquelle j'ai partagé et je partage encore d'agréables moments dans le labo tout comme en dehors de ses murs. Merci à ma sœur de combat et de militantisme féministe, pour les débats les livres partagés et tes conseils de grande sœur !

Je voudrais remercier ma professeur des années école d'ingénieurs Sihem Guemara pour ses encouragements continus et d'avoir toujours cru en moi durant ces années de thèse.

Je voudrais remercier mes ami·e·s, ma deuxième famille, de m'avoir soutenue et de m'avoir surtout supportée, moi et mes sautes d'humeur. Je vais tout d'abord commencer par Manou (sans oublier Abdou), ma grande sœur qui m'a toujours épaulée, soutenue et cru en moi. Merci d'avoir été là pour moi durant cette thèse et d'avoir partagé avec moi mes moments de joie et de m'avoir aidée à surmonter mes moments de doute. Je vous souhaite, à Abdou et toi, une très belle aventure et vie à deux ! Je tiens également à remercier du fond du coeur Asma, ma sœur et ma confidente. Je ne suis pas certaine qu'un paragraphe puisse te rendre justice, je vais m'y essayer quand même. Merci pour tous ces moments de folies partagés dans et en dehors du labo, tu as toujours su être là pour moi et à l'écoute quand j'en avais besoin et me donner de la force même si on le sait bien entre thésardes cela puise dans son propre capital d'énergie. Merci d'être l'amie qui rit à mes blagues malgré elle et d'avoir séché mes larmes quand les maux pesaient sur les mots. Pour tout cela je t'en suis reconnaissante.

Je voudrais remercier Mimi, Manoula, Hejoura, Samouha et Alex pour toutes les sorties et moments de partage que je chéris. Votre compagnie me fait toujours du bien.

Je souhaiterais également remercier ma bande d'amies Arbiette, Sissi, Lawza, Ons, Salma et Wissal, promis je ne vais plus rater nos sorties/voyages sous aucun prétexte. La thèse est finie, je n'ai plus d'excuses. Merci pour vos encouragements continus et votre soutien inconditionnel.

Je tiens également à remercier Mims, tu as été témoin de tout mon combat, et ta présence a été vitalisante !

Merci Abour infiniment de m'avoir soutenue et pour ces moments de rires complices ! Merci d'avoir gardé le lien de cette amitié en vie malgré la distance qui nous sépare !

Je voudrais remercier du fond du coeur mes amies Sonson et Dorra (l'exploratrice) pour leur bonne compagnie, sachez que notre amitié m'est très chère et que je suis reconnaissante pour vos encouragements et nos moments volés pour quelques instants à la thèse à mes passages irréguliers et

brefs à Paris.

Je tiens à remercier Beshbesh pour ton soutien et tes encouragements tout le long de cette thèse, et de m'avoir fait pour la première fois une marraine à Booby!

Je voudrais vous remercier les gars Mehdi, Hamma et Skan, pour votre amitié et encouragement!

Je tiens également à remercier Farouk, sans qui sans doute je n'aurais probablement pas fait cette thèse. Merci Farouk pour ta gentillesse, tes conseils de grand frère et d'être toujours à l'écoute!

Je voudrais remercier mes ami·e·s Ahmed, Sissi, Wacef et Lassad de l'Association Tunisienne de l'Espace (TUNSA) pour cette belle aventure! Merci pour votre soutien et vos encouragements!

Je voudrais remercier Mme. Estelle Beaulieu-Dufils qui a fait un bout de chemin dans cette thèse avec moi. Merci pour votre accompagnement et pour les coups de pouce dont j'avais besoin pour finir ce que j'ai commencé, d'aller jusqu'au bout et de venir à bout! Merci d'avoir cru en moi quand il m'était plus facile de croire au père Noël. Grâce à vous je me sens en mesure de faire face et relever tous les défis, hâte de voir ce que la vie me cache et de vous en parler.

Pour ceux qui ont toujours été là avec moi, la famille, ma maman chérie qui m'a accompagnée durant cette thèse vacillante et forte en émotion. Merci d'avoir eu la patience et les mots justes pour apaiser ma peine et me réconforter quand il le fallait. À chaque passage à Toulouse, si brefs ils étaient, ta présence perdure et égaye mon existence pour des semaines après. Je ne mesure toujours pas la chance de t'avoir à mes côtés, en tant que mère, confidente et amie!

Merci papa chéri de m'avoir soutenue tout le long de ces années, d'avoir toujours été là pour moi et d'avoir eu la patience de m'entendre râler. Merci papa pour ton oreille attentive et tes encouragements permanents.

Je vous dédie, maman et papa chéris, le fruit de toutes ces années d'études de la maternelle jusqu'ici. Merci d'exister et de me donner la force de continuer mon chemin, je vous suis reconnaissante.

Je voudrais remercier également toute la famille : mon frère, ma grand-mère, mes tantes et oncles, et ma marraine, pour leur soutien et encouragements!

Et puis enfin à toute personne qui a semé une graine petite qu'elle soit, merci du fond du cœur.

RÉSUMÉ

Les constellations de satellites ont pris un nouvel essor ces dernières années avec des projets particulièrement ambitieux qui ont suffi à raviver la flamme de la communauté de recherche. Un des problèmes cruciaux est alors l'étude de l'adéquation des protocoles principalement conçus et utilisés dans un contexte terrestre vis-à-vis de ces communications par satellite.

Par le passé, par exemple, des versions de Transmission Control Protocol (TCP) ont été spécialement conçues pour les réseaux de satellites [1]-[5]. Néanmoins, ces travaux pourraient se révéler obsolètes, en raison des évolutions récentes de TCP et en particulier de la refonte complète du contrôle de congestion.

La question à laquelle nous nous sommes attaqués dans cette thèse est la suivante : les versions récentes de TCP *e.g.* TCP Cubic (CUBIC) et Bottleneck Bandwidth and Round-trip propagation time (BBR) sont-elles ou seront-elles capables de répondre aux besoins dans un tel environnement ?

Nous avons évalué les différences entre les piles TCP du passé et celles qui sont actuellement déployées. Nous décrivons en particulier l'évolution de l'utilisation des protocoles du point de vue de la couche transport de CUBIC et de TCP BBR.

Nous avons identifié les sources de variation de délai dans les constellations de satellites pour en étudier l'impact et la fréquence. Les variantes modernes de TCP s'en accommodent en particulier pour les flux longs.

Nous nous sommes ensuite intéressés à l'équité entre les flux utilisant ou non des variantes différentes. Nous avons mis en évidence certains niveaux d'iniquité dans les contextes hétérogènes assez uniformes à ceux trouvés dans le contexte terrestre.

L'ensemble de ces études a été mené au travers de simulations à événements discrets mais aussi d'émulations permettant d'obtenir des résultats plus réalistes.

ABSTRACT

Satellite constellations have taken a new impetus in recent years with even more ambitious future projects. The momentum has lasted long enough to raise the interest of the research community in addressing the adequacy of protocols mainly designed and used for terrestrial networks, to these satellite communications.

There are thus versions of TCP specially designed for satellite networks [1]-[5]. Nevertheless, these previous works could turn out to be obsolete, in particular due to the recent versions of TCP based, for instance, on hybrid-type congestion control algorithms.

The question we tackled in this thesis is : are the recent versions of TCP, such as CUBIC and BBR, able to meet the needs in such an environment?

We evaluated the differences between past and currently deployed TCP stacks. We gave an overview of the evolution of the use of protocols from a transport layer point of view of CUBIC and BBR.

We identified the sources of delay variation in satellite constellations in order to study their impact and frequency. Modern variants of TCP accommodate this, especially for long flows.

We then looked at the fairness between the flows with or without different variants. We highlighted some levels of unfairness in the heterogeneous contexts that are fairly consistent with those found in the terrestrial context.

All of these studies were conducted through discrete event simulations but also emulations in order to obtain more realistic results.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

3GPP 3rd Generation Partnership Project. 12, 13

ACK ACKnowledgement. xiii, 19–23, 29–31, 36, 38, 45, 53, 54

AIMD Additive-Increase/Multiple-Decrease. 22–24

AINE Advanced IP Network Emulator. 43

AQM Active Queue Management. 35, 36, 87

B5G Beyond-5G. 12, 13

BBR Bottleneck Bandwidth and Round-trip propagation time. v, vii, x–xv, 2, 3, 24, 27–31, 34–36, 38, 41, 45, 46, 63–71, 73–87, 101

BDP Bandwidth-Delay Product. xv, 22, 23, 27, 28, 30, 31, 36, 37, 45, 47, 49, 50, 58–60, 64, 68, 70, 74, 76–79, 81–84, 87

BER Bit Error Rate. 41, 45

BIC Binary Increase Control. 24, 26

BTLBW Bottleneck Bandwidth. 27–31

CA Congestion Avoidance. 21–23, 25, 26, 35, 58

CBR Constant Bit Rate. 40

CC Congestion Control. 24, 33

CNES Centre National d'Études Spatiales. 43

CWND Congestion Window. xiii, xiv, 20–22, 24–27, 29–33, 45, 47, 49, 50, 57–64, 67–70, 77, 78, 81

CUBIC TCP Cubic. v, vii, x–xv, 2, 3, 23–28, 30, 31, 34, 38, 41, 46, 49, 50, 52, 54, 56–70, 73, 74, 76–87, 101

DCTCP Data Center TCP. 31, 73, 87

DVB-RCS Digital Video Broadcasting - Return Channel via Satellite. 42, 43

DVB-S Digital Video Broadcasting - Satellite. 42, 43

DELACK Delayed Ack. 29, 36

DIFFSERV Differentiated Services. 43

DUPACK Duplicate Ack. 21–23, 38

ECN Explicit Congestion Notification. 23, 29–31, 35, 73, 87

ESA European Space Agency. 87

EWMA Exponentially Weighted Moving Average. 30

FCC Federal Communications Commission. 15

GEO GEostationary Orbit. ix, 1, 7–9, 12, 16, 32, 40, 41, 64, 87

GMAT General Mission Analysis Tool. 43

GMR GEO Mobile Radio. 12

GPS Global Positioning System. 6

GQUIC Google QUIC. 87

GSM Global System for Mobile Communications. 12

GSO GeoSynchronous Orbit. 1

GSaaS Ground Segment as a Service. 13

GW Gateway. 11, 43, 45–47, 49, 50, 53, 54, 56, 75

HEO Highly Elliptical Orbit. ix, 7

HMM Hidden Markov Model. 24

HTTPS HyperText Transfer Protocol Secure. 2

HTTP HyperText Transfer Protocol. 2

IETF Internet Engineering Task Force. 30, 87

IP Internet Protocol. 19, 43

ISL Inter-Satellite Link. ix, 5, 11–18, 37–40, 42, 45, 47, 49, 50, 64, 65, 76

ITU International Telecommunication Union. 41

IW Initial Window. 20, 21, 44, 53

IoT Internet of Things. 13

L4S Low Latency Low Loss Scalable Throughput. 31, 73, 87

LEO Low Earth Orbit. ix, 1, 8, 9, 12, 13, 15, 16, 37–43, 56, 64

LFN Long Fat Networks. x, 23, 24, 27, 37

LOS Line-Of-Sight. 14

LTE Long Term Evolution. 12

M2M Machine to Machine. 13

MBMS Multimedia Broadcast Multicast Service. 12

MEO Medium Earth Orbit. ix, 8, 9, 12, 13, 15, 16, 40, 42

MPTCP Multipath TCP. 36

MTT Mean Transfer Time. xv, 56

NASA National Aeronautics and Space Administration. 43

NGSO Non-Geostationary Satellite Systems. 12

NR New Radio. 13

NS-2 Network Simulator 2. xi, xiii, 42–44, 46, 49, 50, 58, 60, 63–65, 70, 71, 85

NS-3 Network Simulator 3. 42

NTN Non-Terrestrial Networks. 12

OISL Optical Inter-Satellite Link. 14

OMNET++ Objective Modular Network Testbed in C++. 42

OS3 Open Source Satellite Simulator. 42

OWD One-way Delay. xii–xv, 40, 43, 58–60, 65, 67, 68, 70, 74–84, 86

OPENSAND Open Satellite Network Demonstrator. xi, xiv, 43, 44, 46, 63–71, 73, 76, 85

PEP Performance Enhancing Proxy. 32

QUIC Quick User Datagram Protocol Internet Connections. 2, 3, 87

QOE Quality of Experience. 45

QOS Quality of Service. 12, 43

RFC Request For Comments. 20, 24, 26, 38, 73, 87

RTO Retransmission TimeOut. x, 20–22, 38

RTT Round-Trip Time. x, xii, xiv, xv, 1, 9, 19, 20, 22–24, 26–30, 32, 33, 35–38, 49, 59, 64, 67, 68, 70, 74, 76, 77, 80–84, 86, 87

RTPROP Round-Trip propagation time. 27, 29, 31

RWND Receiver Advertised Window. 28

SACK Selective ACK. 36–38, 41, 50, 64, 76

SMACSAT System and MAC Satellite Simulator. 42

SNS3 Satellite Network Simulator 3. 42, 43

SS Slow Start. 20, 21, 23

ST Satellite Terminal. 75

TCP Transmission Control Protocol. v, vii, x, xi, xiii, xv, 1–3, 5, 19–27, 30–42, 44–46, 49, 56–58, 61–63, 65, 66, 68, 71, 73, 77, 85–87, 101

TFO TCP Fast Open. 36

TLS Transport Layer Security. 2, 36

TOS Type Of Service. 34

TTC Tracking, Telemetry and Command. 6

UDP User Datagram Protocol. 36

UMTS Universal Mobile Telecommunications System. 12

VLEO Very Low Earth Orbit. 1, 15

WI-FI Wireless Fidelity. 31

ENB evolved Node B. 12

SSTHRESH Slow Start Threshold. 20, 21, 25, 30

CHAPITRE 1

INTRODUCTION

En 1945, Sir Arthur C. Clarke, ingénieur écrivain de science-fiction, a prédit qu'il serait un jour possible de communiquer d'un bout à l'autre de la planète grâce à un réseau de trois satellites [6]. Cette prémonition s'est réalisée, dans les années 1970, avec l'introduction de constellations de satellites géosynchrones — GeoSynchronous Orbit (GSO) sur une orbite géostationnaire — GEostationary Orbit (GEO), à 36000km d'altitude. Vers la fin des années 1990, on voit apparaître les premières constellations de satellites non-géosynchrones — non GSO en orbite basse — Low Earth Orbit (LEO), avec une altitude entre 500km et 2000km.

Depuis, tant de chemin a été parcouru.

À la fin des années 2010, la course à l'espace (*Space Race*) est encore une fois de mise pour permettre l'accès à l'Internet par satellite à faible coût et une latence beaucoup plus faible motivée par le besoin fort des utilisateurs d'une couverture Internet à haut débit mondiale inaccessible avec les seules solutions terrestres [7].

Les constellations de satellites ont pris un nouvel essor ces dernières années avec des projets ambitieux. La communauté de recherche s'est donc penchée sur l'adéquation des protocoles principalement conçus et utilisés dans un contexte terrestre pour des communications par satellite. On parle désormais de constellations de satellites qui comptent des centaines de satellites (OneWeb) voire de méga-constellations avec des milliers de satellites (Starlink, Kuiper) évoluant dans une orbite très basse — Very Low Earth Orbit (VLEO) dans le but de combler les lacunes des réseaux terrestres en garantissant plus d'accessibilité et en couvrant plus de zones blanches.

La qualité d'expérience utilisateur de bout-en-bout repose en grande partie sur les performances des protocoles de transport. Leurs évolutions, notamment de Transmission Control Protocol (TCP), allaient de pair avec les constellations de satellites. Plusieurs études de recherche ont été menées dans les années 90 sur les performances dans les constellations de satellites à faible bande passante. Ainsi, TCP, conçu à l'origine pour fonctionner dans un réseau câblé avec un Round-Trip Time (RTT) faible, est sensible dans un contexte satellite aux variations de délai et aux RTT importants [8]-[12]. Parallèlement, des améliorations ont été proposées pour assurer un fonctionnement fiable et efficace de TCP sur les réseaux satellite [13]-[15]. Le comportement de TCP dans des constellations de satellites a ainsi été largement étudié [11], [12], [16], [17]. Il existe des versions spécialement conçues pour les

réseaux de satellites [1]-[5]. Néanmoins, ces travaux pourraient se révéler obsolètes, en particulier en raison des évolutions récentes de TCP et de ses nouvelles versions reposant sur des algorithmes de contrôle de congestion de type hybride.

Ces dernières années, le trafic Internet a été façonné par la popularité de l'application et les choix de l'utilisateur. Il n'est pas si évident que cela d'avoir des pronostics précis autour d'une thématique sujette à des variations brutales [18], [19]. En 2019 selon le rapport *The Global Internet Phenomena Report* de *Sandvine Inc.*, société d'intelligence applicative et réseau, le *streaming* vidéo vient en premier dans le classement des catégories d'applications utilisées dans le monde avec 55.44%. Le *WEB* est à 10.14%. Le Top 5 est regroupé dans le Tableau 1.1 [20]. Concernant les parts des plateformes de *streaming* vidéo dans le trafic global, elles se répartissent comme suit dans le Tableau 1.2. *HyperText Transfer Protocol (*HTTP*) Media Stream* représente 13.76% du trafic, *Netflix* 12.87% et *YouTube* 8.69% [20]. Les applications HTTP reposent actuellement soit sur Quick User Datagram Protocol Internet Connections (QUIC) (si c'est du HTTP3) soit sur TCP/Transport Layer Security (TLS) (pour l'HyperText Transfer Protocol Secure (HTTPS)) [21], [22]. *Netflix* utilise soit du TCP Cubic (CUBIC) soit du *New Reno* [23]. *YouTube* utilise l'un des deux en fonction du protocole de transport choisi par le logiciel côté client : TCP (à notre connaissance, c'est du Bottleneck Bandwidth and Round-trip propagation time (BBR) [23]) ou QUIC, généralement (en fait, BBR peut même être utilisé pour du QUIC) [24].

TABLE 1.1 Part de la catégorie d'application dans le trafic global en 2019-2020- Top 5 [20]

2019		**2020**	
***Streaming* vidéo**	55.44%	***Streaming* vidéo**	57.64% (+2.20%)
WEB	10.14%	***Social networking***	10.73% (+1.78%)
Social networking	8.95%	***WEB***	8.05% (-2.09%)
File sharing	8.51%	***Marketplace***	4.97% (-0.93%)
Marketplace	5.90%	***Messaging***	4.94% (+1.15%)

TABLE 1.2 Parts de l'application dans le trafic total en 2019-2020- Top 3 [20]

2019			**2020**		
HTTP *media stream*	13.76%	QUIC/TCP	***YouTube***	15.94% (+7.25%)	QUIC/BBR
Netflix	12.87%	CUBIC/New Reno	***Netflix***	11.42% (-1.45%)	CUBIC/New Reno
YouTube	8.69%	QUIC/BBR	**HTTP**	6.57% (-2.96%)	QUIC/TCP

En 2020, en période de confinement le *streaming* vidéo est toujours en tête du classement avec 57.64% (+2.20%) [20], comme nous pouvons le voir dans le Tableau 1.2. Concernant les parts des plateformes de *streaming* vidéo du trafic global, elles se répartissent comme représenté dans le Tableau 1.1. *YouTube*, utilisant du BBR et/ou du QUIC, est passé à la première place en 2020 avec 15.94% et *Netflix*, utilisant du CUBIC et/ou du *New Reno*, garde toujours sa 2ème place avec 11.42%. HTTP, utilisant du QUIC et/ou du TCP, marque son entrée dans le Top 3 avec 6.57% [20]. À notre connaissance, les tendances d'utilisation des protocoles de la couche 4 transport n'ont pas changé sur

la période.

En 2021, les prévisions annoncent un trafic engendré de même nature du moins pour les mobiles [25]. La plupart des applications de réseaux sociaux déploient du QUIC (Instagram et Facebook [26], les navigateurs *web* également (Chrome et Safari [26]), tandis que les plateformes de streaming *vidéo* sont plutôt en TCP.

Dans le Tableau 1.2, on voit que BBR (dont la version n'est pas précisée, mais probablement la toute dernière) et CUBIC figurent parmi des algorithmes de contrôle de congestion de TCP les plus prisés par les applications utilisées. Cette tendance est également confirmée dans [23] et se voit sur la Figure 2.11.

La question à laquelle nous nous sommes attaqués dans cette thèse est la suivante : les versions récentes de TCP *e.g.* CUBIC et TCP BBR sont-elles ou seront-elles capables de répondre aux caractéristiques très spécifiques des communications par satellite à orbite basse?

Dans les études passées sur TCP [12], [16], la décomposition des variations de délai en différentes sources et l'impact de chacune d'entre elles sur un transfert fondé sur TCP fait défaut. De plus, compte tenu de certaines évolutions récentes de la pile TCP et de l'augmentation du débit offert par les constellations de satellites, les conclusions de ces travaux doivent être potentiellement reconsidérées.

Nous avons tout d'abord identifié les sources de variation de délai dans les constellations de satellites pour en étudier l'impact et la fréquence.

Nous avons par la suite évalué les différences entre les piles TCP du passé et celles qui sont actuellement déployées : CUBIC qui utilise un algorithme *loss-based* — fondé sur les pertes, et TCP BBRv1 qui utilise un algorithme hybride qui repose sur l'estimation du délai et de la capacité du réseau.

Nous avons ensuite abordé la question de l'équité. En effet, les différentes variantes de TCP, qui sont encore largement déployées, peuvent présenter des performances hétérogènes. Nous nous sommes donc attaqués à la question de l'équité entre une variante candidate des TCP existants CUBIC et une variante qui fait son entrée BBRv2. Elles sont désormais majoritaires pour transporter le trafic Internet.

Tout au long de cette thèse, nous avons mené nos campagnes d'évaluation à l'aide de simulation mais aussi d'un outil d'émulation qui est plus réaliste.

CHAPITRE 2

ÉTAT DE L'ART

2.1 Objectif

Les constellations de satellites déploient leur flotte, principalement avec des liens inter-satellites — Inter-Satellite Link (ISL)s, permettant plus de connectivité, et essentiellement dans la bande *Ka*, offrant ainsi plus de fréquences utilisables et une meilleure directivité des faisceaux satellites. Elles offrent ainsi une couverture omniprésente et mondiale avec un minimum d'infrastructure au sol, ce qui permet à des solutions *broadband* d'atteindre même les zones reculées où le déploiement de réseaux purement terrestres aurait un coût prohibitif. Dans ce qui suit nous allons aborder l'environnement satellite, ses spécificités qui lui confèrent des avantages et des inconvénients par rapport aux réseaux terrestres et ses domaines d'application. Par la suite, nous ferons le parallèle avec l'évolution du protocole TCP et la mesure dans laquelle son adaptation est en adéquation avec le contexte satellite sans enfreindre l'esprit de TCP.

2.2 Environnement satellite

Dans cette sous-section, nous allons nous intéresser au contexte satellite, la définition d'un satellite, ses domaines d'application et l'architecture des systèmes de communication. Ensuite, nous nous focaliserons plus spécifiquement sur l'orbitographie et les constellations de satellites. Nous présenterons finalement l'impact de leurs caractéristiques intrinsèques sur la variation de délai.

2.2.1 *Qu'es aquò* un satellite

Les satellites (artificiels) sont apparus vers la fin des années 1950, matérialisant l'aboutissement d'une idée ambitieuse de l'ingénieur visionnaire Arthur C. Clarke, avec le développement des télécommunications. Un satellite est une entité électronique envoyée dans l'espace sur une orbite et une altitude bien précises variant d'un satellite à un autre selon la mission (fonctionnalités à assurer), avec une durée de vie, communiquant avec la Terre au travers d'antennes, et dont le cycle de vie normal se termine soit par une phase de mise en orbite de rebut, soit par une désorbitation puis une entrée dans

l'atmosphère.

2.2.1.1 Domaines d'application des satellites

On distingue trois types de satellites : les satellites scientifiques, les satellites commerciaux et les satellites militaires. Les domaines d'application les plus importants sont :

— **télécommunication :** ils constituent la première application commerciale. Ils permettent d'assurer la transmission de données, les communications téléphoniques et la diffusion de programmes de télévision.
Le satellite peut travailler de pair avec les réseaux terrestres, par exemple pour du *backhauling*.
Telstar 1 fut le premier satellite de télécommunication, en 1962.

— **télédétection :** ils sont conçus pour des missions d'observation de la Terre.
ERTS 1 (plus tard renommé *Landsat* 1) est le premier satellite conçu spécifiquement pour la télédétection.

— **positionnement :** ils permettent de donner la position d'un utilisateur à la surface de la Terre, en l'air ou dans l'espace.
Le premier système positionnement par satellite Global Positioning System (GPS) a été développé par les États-Unis. Il est opérationnel depuis 1995.

— **militaire :** pour des buts militaires ou gouvernementaux, ces satellites permettent les applications mentionnées précédemment par exemple à des fins d'espionnage.
Spoutnik 1, le premier satellite de l'ère spatiale est compté parmi les premiers satellites militaires, en 1957.

Nos travaux de thèse se positionnent dans le contexte des satellites de télécommunication.

2.2.1.2 Système de communication par satellite

La Figure 2.1 permet d'illustrer le système de communication et son interconnexion avec les entités terrestres.

Le système satellite est composé d'un segment spatial — *space segment*, un segment de contrôle — *control segment* et un segment sol — *ground segment* :

— **Segment spatial :** il contient un ou plusieurs satellites actifs ou de réserve organisés en constellation.

— **Segment de contrôle :** il comprend toutes les installations au sol pour le contrôle et la surveillance des satellites, également appelé stations de Tracking, Telemetry and Command (TTC), et pour la gestion du trafic et des ressources associées à bord du satellite.

— **Segment sol :** il est constitué de toutes les stations sol de trafic. Selon le type de service considéré (*Point-to-point*, *Broadcast/multicast*, *Collect*, *Mobile*), ces stations peuvent être de taille différente, de quelques centimètres à plusieurs dizaines de mètres.

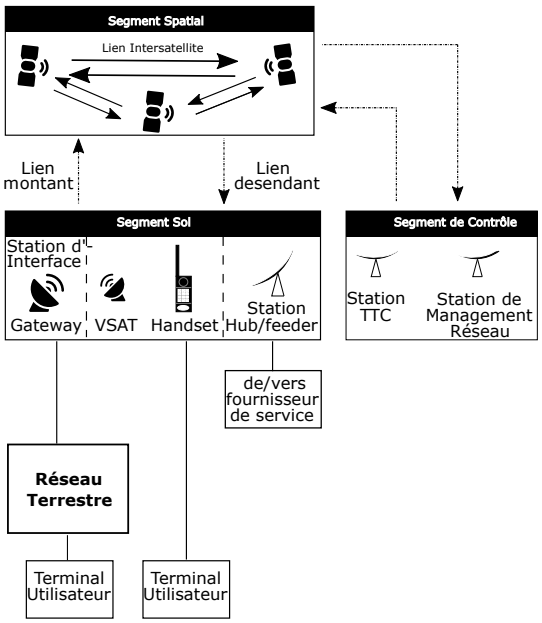


FIGURE 2.1 Système de communication par satellite, interconnexion avec les entités terrestres [27]

2.2.2 Orbitographie satellite

Une orbite est une trajectoire régulière et répétitive qu'un objet dans l'espace emprunte autour d'un autre objet plus massif.

La géométrie orbitale a un effet considérable sur la conception du réseau satellite. Elle influence la couverture et la diversité des satellites, les considérations de propagation physique telles que les contraintes de puissance et les bilans de liaison. La topologie dynamique du réseau qui en résulte ainsi que la latence et le temps de propagation aller-retour sont particulièrement importants [28].

2.2.2.1 Altitude

Sur la Figure 2.2, nous distinguons les différentes orbites vues par leur altitude.

2.2.2.1.1 Highly Elliptical Orbit (HEO)

C'est une orbite elliptique à forte excentricité, se référant généralement à une orbite autour de la Terre pour la couverture des zones de latitude élevée [30].

Parmi les orbites les plus connues, il y a l'orbite *Molniya*. Il s'agit d'un type d'orbite conçu pour assurer la couverture des communications et de la télédétection [30].

2.2.2.1.2 GEostationary Orbit (GEO)

Les satellites en orbite géostationnaire (GEO) tournent autour de la Terre au-dessus de l'équateur d'ouest en est en suivant la rotation de la Terre — qui dure 23 heures 56 minutes et 4 secondes — en

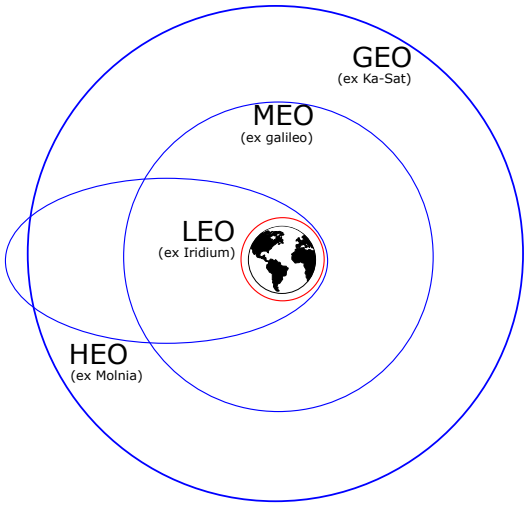


FIGURE 2.2 Quelques exemples d'orbites dans un plan hypothétique commun [29]

se déplaçant exactement à la même vitesse que la Terre. Les satellites en orbite GEO semblent donc être "stationnaires" au-dessus d'une position fixe. Afin de suivre parfaitement la rotation de la Terre, l'altitude d'une orbite GEO est de l'ordre de 35786km [31].

L'orbite GEO est utilisée par les satellites de télécommunication. Elle peut également être utilisée par les satellites de surveillance météorologique [31].

2.2.2.1.3 Medium Earth Orbit (MEO)

L'orbite Medium Earth Orbit (MEO) comprend un large éventail d'orbites situées entre les orbites LEO et GEO. Elle est utilisée par une grande variété de satellites ayant des applications très diverses [31]. Elle est très couramment utilisée par les satellites de navigation, comme le système européen Galileo [31].

2.2.2.1.4 Low Earth Orbit (LEO)

Une orbite terrestre basse (LEO) est, comme son nom l'indique, une orbite relativement proche de la surface de la Terre. Elle se situe normalement à une altitude inférieure à 1000km, mais peut descendre jusqu'à 160km [31].

Les satellites LEO ne doivent pas toujours suivre une trajectoire particulière autour de la Terre de la même manière — leur plan peut être incliné. Cela signifie qu'il y a plus de trajectoires disponibles pour les satellites LEO. C'est l'une des raisons pour lesquelles, LEO est une orbite très couramment utilisée [31].

Cependant, les satellites LEO individuels sont moins utiles pour les télécommunications, car ils se déplacent très rapidement dans le ciel et imposent donc beaucoup d'efforts pour les suivre depuis des stations au sol [31].

Au lieu de cela, les satellites de communication en orbite basse font souvent partie d'une grande combinaison ou constellation de plusieurs satellites pour assurer une couverture constante. Afin d'accroître la couverture, il arrive que des constellations de ce type, composées de plusieurs satellites identiques ou similaires, soient lancées ensemble pour créer un "réseau" autour de la Terre. Ce réseau est appelé constellation de satellites, nous les définirons plus précisément dans le prochain paragraphe. Cela permet à ces satellites de couvrir simultanément de grandes zones de la Terre en travaillant ensemble [31].

2.2.3 Constellations de satellites

2.2.3.1 Définition de constellations de satellites

Une constellation de satellites est un ensemble de satellites identiques travaillant de concert pour procurer une prestation en assurant généralement une couverture quasi-complète de la planète. Ces satellites circulent sur des orbites bien définies choisies de manière à ce que leurs couvertures au sol respectives se complètent [32].

Les satellites en orbites MEO et LEO sont souvent déployés en constellations de satellites, car la zone de couverture fournie par un seul satellite ne couvre qu'une petite zone qui se déplace lorsque le satellite se déplace avec une vitesse angulaire élevée nécessaire pour maintenir son orbite. De nombreux satellites MEO ou LEO sont nécessaires pour maintenir une couverture continue sur une zone. Cela contraste avec les satellites géostationnaires, où un seul satellite, situé à une altitude beaucoup plus élevée et se déplaçant à la même vitesse angulaire que la rotation de la surface de la Terre, assure une couverture permanente d'une vaste zone.

Pour certaines applications, en particulier la connectivité numérique, l'altitude plus basse des constellations de satellites MEO et LEO offre des avantages par rapport à un satellite géostationnaire, avec des affaiblissements de propagation — *path losses* moins importants (ce qui réduit les besoins en énergie et les coûts) ainsi qu'une latence plus faible. Le RTT est de l'ordre de 500ms en GEO, de 125ms pour un satellite MEO et de 30ms pour un système LEO, par exemple.

Notons qu'une approche multi-orbite ou hybride est possible permettant de tirer profit de chacune de ces orbites retenues.

Le renouveau des constellations, et plus particulièrement LEO, est flagrant avec le coût de plus en plus réduit des lanceurs réutilisables (*i.e.* ceux de *SpaceX*). Par ailleurs, il y a un besoin croissant de couverture omniprésente avec des services gourmands en bande passante. Cela allégera la charge sur le segment sol et ajoutera de la robustesse aux liens utilisateurs. L'approche hybride (née vers les années 2000 [33], [34]) a également connu un regain d'intérêt [35].

2.2.3.2 Géométrie orbitale

Les constellations de satellites ont été conçues de manière à avoir le nombre minimum de satellites, dans des plans réguliers, similaires, pour une couverture optimale du sol [10].

Les orbites circulaires sont utilisées afin de garantir une couverture au sol constante, ou une taille fixe de l'"empreinte" du satellite tout au long de l'orbite [10]. Les satellites en orbite circulaire peuvent

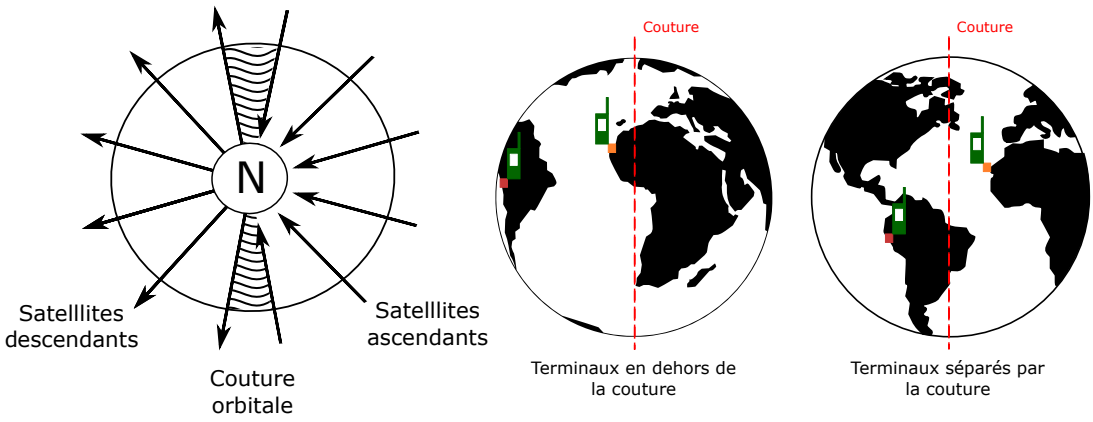


FIGURE 2.3 Position de la couture orbitale dans les constellations de satellites polaires ou quasi-polaires, en particulier *Iridium NEXT* et pour des positions de terminaux par rapport à la couture

fournir une couverture continue d'une zone du sol située en dessous ("empreinte"), mais la zone couverte par le satellite se déplace au fur et à mesure que le satellite se déplace sur son orbite. L'altitude de ces orbites est choisie en fonction de considérations physiques et géométriques, notamment la puissance du signal, le temps de visibilité du satellite, la zone de couverture et l'évitement des ceintures de rayonnement de Van Allen [28].

Il existe, donc, deux types de géométrie de constellations : [10]

1. **Constellation *Walker star*** : Constellation de satellites polaires et quasi-polaires (constellation π passant par les deux pôles)

   Cette constellation est constituée de plans orbitaux inclinés à un angle constant, proche de 90 °, par rapport au plan de référence (l'équateur). Leurs angles d'ascension, et l'espacement entre eux, remplissent 180 °de l'équateur, d'où l'appellation de constellation-π. Les constellations quasi-polaires, avec une "couture" orbitale entre les plans ascendant et descendant sont appelées constellations *Walker star patterns*, on peut les distinguer sur la Figure 2.3. Le nom d'"étoile" vient du fait que, sur une projection polaire, les plans orbitaux se croisent pour former une étoile.

   Comme les plans orbitaux adjacents se croisent aux pôles, les satellites correspondants sont plus proches et donc la couverture est bien disparate selon la latitude. Le long de l'équateur la séparation orbitale entre les plans est la plus grande, la couverture est comblée par les chevauchements des "empreintes" — "*footprints*" pour une couverture complète et continue.

   On peut donc distinguer 3 types de liens (inter-satellite) :
   — Intra-orbital : lien maintenu entre deux satellites successifs appartenant à un même plan orbital.
   — Inter-orbital : lien maintenu entre deux satellites successifs appartenant à deux plans orbitaux adjacents prograde.
   — *Cross-seam* : lien maintenu entre deux satellites appartenant à deux plans orbitaux adjacents

rétrogrades (les satellites sont ascendants et descendants, de part et d'autre).

L'utilisation des ISLs présente des avantages importants, comme la réduction des communications entre les satellites et les Gateway (GW)s, la réduction du nombre de GWs, l'équilibrage de charge entre les GWs et la prévention des pannes de GWs.

Les vitesses relatives élevées des satellites se déplaçant dans des plans adjacents rendent plus difficile le maintien des ISLs à des latitudes élevées en raison de l'effet Doppler accru, de la vitesse de *tracking* plus élevée et de la nécessité de permuter les voisins et de rétablir les liaisons lorsque les plans orbitaux se croisent aux pôles. Les *cross-seam* ISLs peuvent ne pas être maintenus en raison des vitesses d'approche élevées (effectivement deux fois la vitesse orbitale) des satellites ascendants et descendants.

Aux pôles, le chevauchement des *footprints* des satellites entraînera une couverture multiple et des interférences de signaux pour les systèmes d'attribution de fréquences simples, ce qui nécessitera la désactivation de certaines empreintes ou de certains faisceaux ponctuels.

2. **Constellation rosette** : (constellation-2π)

   Le principal inconvénient des constellations *Walker Star* est la couture, du point de vue des réseaux de communication, et c'est pour cela que la topologie rosette a été proposée [36]. Cette constellation est constituée de plans orbitaux inclinés à un angle constant, généralement inférieur à 90°, par rapport à un plan de référence (l'équateur). L'espacement régulier des angles droits des nœuds ascendants sur les 360° de longitude signifie que les plans ascendants et descendants des satellites et leur couverture se chevauchent continuellement, au lieu d'être séparés comme avec la *Walker star*. D'où le nom 2π.

   Bien que l'entrelacement et l'alignement de phase minutieux des plans de couverture puissent être adoptés par les rosettes, *e.g. Globalstar*, pour réduire le nombre de satellites nécessaires, leurs orbites inclinées imposent généralement des contraintes plus sévères sur les ISLs. Les vitesses relatives, les exigences de poursuite et l'effet Doppler sont globalement accrus par rapport aux liens inter-plan pour les orbites polaires. Aucune constellation rosette n'a encore été mise en œuvre avec des ISLs.

   Il n'y a pas de couverture au-delà d'une certaine latitude qui dépend de la valeur de l'angle d'inclinaison ; les constellations en rosette inclinée négligent généralement la couverture polaire, tout en offrant le plus haut degré de couverture aux latitudes moyennes.

Il convient de souligner que la combinaison de deux constellations-π de satellites opposées avec le même angle d'inclinaison donne une constellation-π. Mais il est peu probable que les ISLs entre les étages des deux *Walker star* puissent communiquer en raison des exigences imposées à l'équipement d'acquisition et de poursuite, des vitesses relatives élevées des satellites, ainsi que de la fréquence élevée de *handover* et de l'effet Doppler.

2.2.3.3 Applications des constellations de satellites par télécommunication

Les constellations de satellites peuvent être utilisées dans de nombreux contextes applicatifs tels que la navigation et l'observation de la terre. Nous allons nous focaliser sur les satellites de

télécommunication.

Les constellations de satellites ont marqué leur entrée avec l'introduction des services analogiques de première génération, comme la voix et autres applications à faible débit de données, principalement dans les scénarios maritimes, (*i.e. Inmarsat*) et ce indépendamment des réseaux terrestres.

Au début des années 1990 avec l'arrivée de la 2G, les communications par satellite ont été exploitées pour fournir des services tels que la voix aux personnes voyageant à bord d'avions ainsi que pour assurer une couverture dans certaines zones terrestres. Entre-temps, les constellations de satellites Non-Geostationary Satellite Systems (NGSO) (*e.g.*, *Iridium NEXT* et Globalstar) ont attiré l'attention de la communauté des chercheurs en raison de leur capacité à fournir une couverture satellite mondiale. Toutefois, il s'est avéré qu'elles étaient coûteuses pour concurrencer les solutions GEO et les réseaux cellulaires [37]. Néanmoins, là où les réseaux terrestres sont coûteux à déployer, les GEO ont conquis des domaines de niche.

En dépit de ces soucis réglementaires (satellites de type solutions propriétaires), l'idée de combiner les réseaux satellite et terrestres a émergé en raison des coûts élevés, de la couverture limitée des réseaux terrestres et de la faible exploitation des satellites. Cela a été au départ avec le Global System for Mobile Communications (GSM) via un lien GEO Mobile Radio (GMR), puis par la suite, pour la 3G Universal Mobile Telecommunications System (UMTS).

* **3G UMTS :** La 3G a marqué la fin de la concurrence entre réseaux terrestres et satellite et le début de la collaboration. Cela a été la première étape vers la convergence des systèmes mobiles et *broadband* en offrant des services Multimedia Broadcast Multicast Service (MBMS) à des groupes d'utilisateurs.
  L'intégration des constellations de satellites NGSO introduisant les ISLs a permis une couverture globale avec un moindre délai que les GEOs et, dans une certaine mesure, avec un moindre coût que les réseaux terrestres. Elles ont donc imprégné les générations ultérieures d'accès réseaux.
* **Long Term Evolution (LTE) :** Les satellites d'une constellation peuvent jouer le rôle d'une station sol (evolved Node B (eNB) en dénomination 4G). Les terminaux LTE communiquent directement avec le satellite sans intermédiaire. Certaines études ont proposé des modifications au standard pour le rendre compatible avec un lien satellite. [38] considère une constellation de satellites MEO du type O3B, tandis que [39] s'intéresse à l'utilisation d'une méga-constellation LEO pour un système intégrant LTE.
* **5G, Beyond-5G (B5G) et 6G :** Les systèmes de communications mobiles 5G offrent une qualité de service Quality of Service (QoS) élevée, une grande capacité et une latence extrêmement faible. Néanmoins, les réseaux terrestres traditionnels ne peuvent pas fournir un accès internet à tout le monde, en particulier dans les avions ou dans les régions rurales (zones blanches). Parallèlement, les Non-Terrestrial Networks (NTN), plus particulièrement les systèmes de constellations de satellites, permettent de relever les défis liés à la fourniture de la connectivité. Pour bénéficier des économies d'échelle de l'écosystème 5G et avec le regain d'intérêt pour la *broadband* fournie par les constellations de satellites LEO, l'industrie des satellites s'est engagée dans le processus 3rd Generation Partnership Project (3GPP) pour intégrer les réseaux

satellite dans la 5G New Radio (NR) et son évolution en B5G [40]. Cela s'est matérialisé pour la première fois par le 3GPP dans Rel-15. Des activités plus récentes ont pris le relais pour les Rel-16 et Rel-17 afin de définir des scénarios et des paramètres de déploiement [41], [42]. Dans [43], des cas d'utilisation ont été étudiés, dans des domaines verticaux, B5G et 6G avec des constellations de satellites LEO, ainsi que les différents défis, contraintes et exigences pour réussir cette intégration [44].

* **Internet of Things (IoT)/Machine to Machine (M2M) :** La quasi-omniprésence des constellations de satellites et surtout la faible altitude de certains types de constellations permettent d'avoir des applications pour les réseaux d'IoT (*Space* IoT) et M2M [45].
  Le projet Mustang (Machine-to-machine Utilisant le Satellite et le Terrestre pour des Applications de Nouvelle Génération) lancé par Airbus Defence and Space a étudié des extensions de couverture Sigfox avec des satellites LEO [46], [47].
  Semtech, Inmarsat et Lacuna Space utilisent quant à eux la technologie LoRa à bord de satellites LEO [47].
* **Edge, fog and cloud computing :** L'idée derrière l'*edge* et le *fog computing* est d'avoir accès aux ressources à proximité (pour des latences et des coûts réduits), à destination d'applications nécessitant un temps de réponse court telles que la réalité augmentée et la réalité virtuelle, ou encore les futurs réseaux de communications [48], [49].
  Avec la multitude de projets de constellations en cours de lancement en LEO, leur omniprésence et une faible latence grâce aux ISLs pour desservir les utilisateurs, l'intégration du satellite dans l'*edge* avec l'introduction de l'*on-board processing* dans les satellites est parfaitement envisageable.
  Les acteurs du cloud tel que Microsoft, avec sa plateforme Azure, a lancé *Azure Space*, en Octobre 2020, en partenariat avec les acteurs du spatial tels que *SpaceX* (méga-constellation *Starlink*) et SES (constellation de satellites MEO *O3b mPOWER*) pour offrir des services de *cloud computing* [50]. La concurrence est rude avec le lancement du *cloud computing unit* d'Amazon *Aerospace and Satellite Solutions* après le succès de l'*AWS Ground Station* (projet dévoilé en 2018) auprès de la communauté spatiale, telle que la NASA, et d'opérateurs satellites, tel qu'*Iridium NEXT*, offrant un Ground Segment as a Service (GSaaS) [51], [52]. D'autres projets visant à assurer des services de *satellite-based cloud computing* voient le jour ou bien sont en cours tels que le projet tout récemment dévoilé : *LEOcloud* [53] et le projet *Spacebelt* de *Cloud Constellation* [54].

2.2.3.4 Projets de constellations de satellites

2.2.3.4.1 Marché des constellations de satellites

Les constellations de satellites LEO offrant une connectivité globale ont connu de lourds échecs dans les années 1990. Parmi les projets *Globalstar*, *Iridium*, *Odyssey* et *Teledesic* seuls *Globalstar* et *Iridium* ont abouti, en raison des coûts élevés et de la demande limitée. Les industriels étaient alors réticents vis-à-vis des constellations de satellites LEO. Les déboires de *LeoSat* n'ont fait que confirmer

les craintes. Depuis, les projets ont mûri d'un point de vue technologique mais aussi le marché y est prêt : des services exigeant de hauts débits et de faibles latences sont apparus avec des besoins en couverture large. Le succès de la relance tient à certains facteurs tant sur le plan technologique qu'économique qui sont, bien évidemment, étroitement liés : réduction des coûts de lancement, de fabrication des engins spatiaux, des équipements au sol et des équipements utilisateurs [55], [56].

2.2.3.4.2 Les constellations de satellites

Dans le Tableau 2.2, nous avons regroupé certains projets de constellations de satellites et dans ce qui suit nous explicitons cette classification.

Orbite : Nous avons montré l'importance du choix de l'orbite pour bien mener la conception d'une constellation de satellite, d'où la présence d'une colonne dédiée à ce paramètre décisif.

ISLs : Les constellations de satellites à basse altitude ont souvent une densité importante de satellites. Les satellites sont donc assez proches pour avoir une Line-Of-Sight (LOS) avec leurs voisins. L'atout primordial est donc de pouvoir maintenir les liens inter-satellites ce qui permet d'alléger la charge et les demandes sur la station sol, et de proposer une couverture avec plus de chemins candidats donc de créer une certaine diversité. Toutefois, toutes les constellations n'en disposent pas car cela exige plus de complexité à bord du satellite *e.g.* des antennes classiques dans le cas des ISLs et laser dans le cas des Optical Inter-Satellite Link (OISL)s ou encore un routage à bord du satellite. Nous nous intéresserons dans ce qui suivra aux constellations de satellites ayant recours aux ISLs.

Bande de fréquences : Du fait de la variété des applications et des fonctionnalités que peut assurer un satellite mais encore une constellation de satellites, plusieurs bandes de fréquences peuvent être utilisées d'où le besoin d'une standardisation. Généralement, les bandes de fréquences plus élevées permettent d'accéder à des largeurs de bande plus importantes, mais sont également plus sensibles à la dégradation du signal due à l'évanouissement lié à la pluie — *rain fading*. Les bandes de fréquences basses sont de plus en plus saturées, en raison de l'utilisation, du nombre et de la taille accrus des satellites. La bande de fréquences satellite utilisée s'étend sur l'intervalle $1-40$GHz.

Type de service : Nous les avons évoqués précédemment.

Nombre de satellites : Le nombre de satellites dans une constellation permet d'avoir une idée sur sa topologie, l'étendue de sa couverture (avec le nombre de plans orbitaux) et sa dynamique. C'est un paramètre fondamental pour la conception d'une constellation de satellites, en fonction du service ; il est étroitement lié au choix de l'orbite et donc de l'altitude.

Nombre de spots par satellite : Les spots satellites font référence aux faisceaux distincts de l'empreinte du satellite — *footprint* (principe similaire aux cellules d'un réseau cellulaire terrestre). Les spots permettent d'implanter le mécanisme de réutilisation de fréquences et d'avoir des schémas plus colorés selon les méthodes d'accès, et donc de pouvoir desservir plus d'utilisateurs et de répondre à plus de besoins.

Parmi les constellations de satellites en cours d'exploitation dans le Tableau 2.2, on compte deux projets qui ont survécu au fiasco des années 1990 à savoir *Globalstar* et *Iridium*. Ils ont fait peau neuve

et sont de retour avec de nouvelles générations en orbite LEO. *Iridium* s'appelle désormais *Iridium NEXT*.

Par la suite, après un regain d'intérêt catalysé tant par le besoin que par une maturité technologique permettant une réduction importante des coûts, plusieurs projets de constellations commencent à se concrétiser pour la plupart en orbite LEO, voire VLEO, avec plus de densité de satellites allant de quelques centaines jusqu'à plusieurs milliers. Une nouvelle génération d'*O3B* : l'*O3B mPower*, et *ViaSat* comptent se lancer en orbite MEO annonçant ainsi une concurrence rude en offrant une connectivité performante et omniprésente le tout à un prix abordable.

Néanmoins, certains projets de constellations, mentionnés dans le Tableau 2.2 ci-contre, n'ont pas vu le jour, faute de financement. Quant à *Boeing*, il est venu à la rescousse de *OneWeb* en lui cédant ses autorisations de la Federal Communications Commission (FCC) en 2017.

Le besoin en haut-débit à prix abordable et les services désormais cruciaux gourmands en bande passante, ne cesse de s'accroître avec la fracture numérique qui se creuse. Par exemple la crise du covid-19 a mis les infrastructures internet à rude épreuve.

Pour tout ce qui suit, la constellation de satellites *Iridium NEXT* va servir de modèle : elle est en orbite circulaire, quasi-polaire (constellation-π) LEO, disposant d'ISLs, pour des services de télécommunication et elle est complètement déployée.

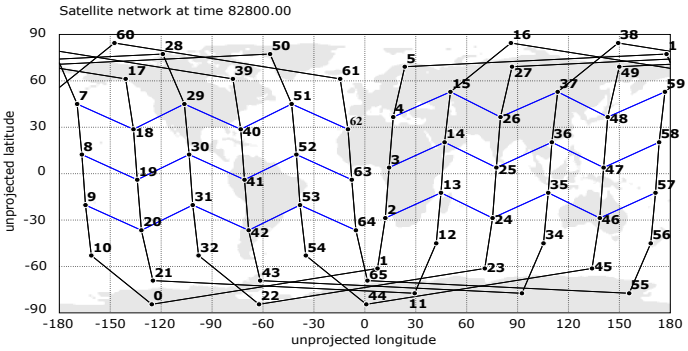


FIGURE 2.4 Tracé de la topologie d'*Iridium NEXT* [57]

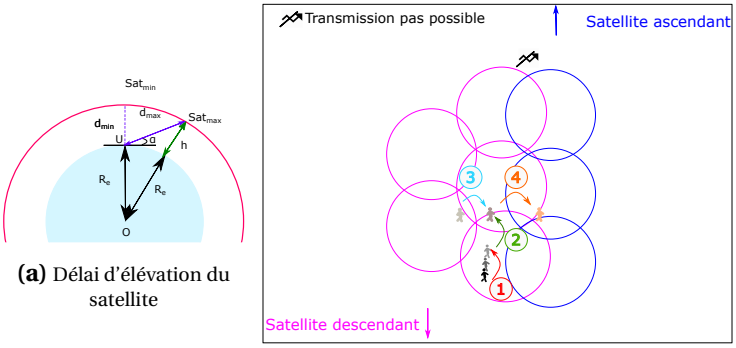
Dans la Figure 2.4 sur une carte non projetée, on peut distinguer les 6 plans orbitaux, les 11 satellites par plan et 4 ISLs par satellite.

En général, les satellites d'*Iridium NEXT* maintiennent 4 ISLs comme on peut le voir sur la Figure 2.4 : 2 intra-plan (en noir) qui sont toujours maintenus et 2 liens inter-plans intermittents (en bleu). Les liens inter-plans sont désactivés près des pôles à cause de la rotation à grande vitesse des satellites, et sur la "couture" à cause des satellites en contre-rotation qui se chevauchent à grande vitesse.

Le Tableau 2.1 résume les paramètres orbitaux de la constellation de satellites *Iridium NEXT* qui servira par la suite de référence.

Paramètre	Valeur
Altitude (km)	780
Plan orbital	6
Satellites par plan	11
Inclinaison (°)	86.4
Séparation inter-plan (°)	31.6
Séparation à la couture (°)	22
Phasing intra-plan	oui
Phasing inter-plane	oui
ISLs par satellite	4
Cross-*seam* ISLs	non

TABLE 2.1 Paramètres d'*Iridium NEXT*



(a) Délai d'élévation du satellite

(b) Les différentes transitions de variation de délai

FIGURE 2.5 Délai d'élévation du satellite et différentes transitions de variations de délai

2.2.4 Impact de l'environnement des constellations de satellites

La dynamique inhérente de la topologie de constellations de satellites (MEO ou LEO) introduit des configurations hétéroclites. Bien que les constellations de satellites LEO soient connues pour avoir un délai de propagation plus faible que les constellations MEO ou GEO, elles présentent des variations de délai plus importantes dues au mouvement des satellites par rapport à la station et au terminal sol, et dues à la topologie de la constellation elle-même. Ce phénomène peut être décomposé en plusieurs facteurs [74] que nous allons décrire maintenant.

2.2.4.1 Délai dû à la variation d'élévation

Lorsque le satellite se déplace par rapport au terminal au sol, l'angle d'élévation du satellite varie, ce qui entraîne une variation de la distance oblique terminal terrestre-satellite. Il en résulte une variation de délai de propagation. Cette variation peut être calculée à l'aide du schéma simplifié de la Figure 2.5 et en appliquant de simples formules trigonométriques.

Dans le cas présenté, le délai de propagation varie dans l'intervalle $[2,6\text{ms}\,;\,8,2\text{ms}]$.

TABLE 2.2 Constellations de satellites récentes [43], [58]-[73]

Constellations de satellites en cours d'exploitation

Constellation	Opérationnelle en	Orbite	ISL	Bande	Type de service	# de satellites	# de spots par satellite
- *Iridium NEXT* (Thales Alenia Space, Orbital ATK)	2018	LEO (quasi-polaire 780 km)	4	- Ka (inter-satellite) - L (uplink and downlink)	Low-speed data communications	- 66 - 6 plans orbitaux - 11 /plan	- 48 Mobile Satellite Service (MSS) beams +2 (feeder links) - dirigeable
- O3B 1st Generation (Thales Alenia Space)	2014	MEO (8000km)	ND	Ka	Trunking (cruise ship)	20	- 10 faisceaux/sat - 12 antennes dirigeables
- Globalstar 2nd Generation (Thales Alenia Space)	2010-2013	LEO (1414km)	Aucun	L-S-C	Satellite phone and low-speed data communications	24	- 16 spot de faisceaux/sat - spot de faisceau fixe

Constellations de satellites à venir

Constellation	Opérationnelle en	Orbite	ISL	Bande	Type de service	# de satellites	# de spots par satellite
- Kuiper (Amazon)	Demi-déploiement by 2026	LEO (590km, 610km, 630km)	ND	Ka	Broadband (400 Mbit/s	3236 (98 plans orbitaux)	ND
- Starlink (SpaceX)	2020-2025	VLEO (345.6km, 340.8km, 335.9km)	ND	V	Broadband	7518	- Dirigeable (phased array)
		LEO (1150km, 1110km, 1130km, 1275km, 1325km)		Ku/Ka		4425	
- Hongyan (CASC)	2023	LEO (900km,	ND	L, Ka	Broadband and navigation (100 Mbit/s)	60	ND
		ND	ND			270	ND
- O3B mPOWER (O3B 2nd Generation) (Boeing)	2022	MEO (8000km)	Aucun	Ka	Trunking (cruise ship)	7	4000+ faisceaux/sat (30000 spot faisceaux)
- Telesat Canada (Airbus, SSTL, SS/Lora)	2021	LEO (polaire 1000km et inclinée 1248km)	max 4	Ka	Proche fibre optique/câble	- 117-512 (120 by 2021) - 6 plan polaire : 12+ /plan - 5 plans inclinés : 9+ /plan	45 (16+ faisceaux utilisateurs dirigeables & 2 faisceaux dirigeables de la gateway)
- OneWeb (OneWeb, Airbus, Virgin)	2020	LEO (1200km)	Aucun	- Ku (sat user link) - Ka (sat gateway link)	Broadband	- 720 - 18 plan orbitaux quasi-polaires	- 16 faisceaux utilisateurs - non-dirigeables
		MEO (8500km)	ND	V	Broadband	1280	ND
- ViaSat (Viasat Inc.)	2020	MEO (polaire 8200km)	ND	Ka	Broadband	24 (ViaSat 1 & 2) 8 plans	ND

Constellations de satellites annulées

Constellation	Annulée en	Orbite	ISL	Bande	Type de service	# de satellites	# de spots par satellite
- LeoSat (Thales Alenia Space)	Novembre 2019	LEO (1400km)	4	Ka	Proche fibre optique/câble	78-108	12 faisceaux de spot dirigeables
- Boeing (Boeing Satellite)	Décembre 2017	LEO (1200km)	Aucun	V	Broadband	1396-2956	ND

2.2.4.2 Délai dû au *handover* intra-orbital

Lorsque le satellite passe en dessous du masque d'élévation du terminal, la communication est transférée au satellite suivant qui pourrait se trouver dans le même plan et qui répond au critère d'être au-dessus du masque d'élévation. Pour les terminaux fixes, ce phénomène se produit toutes les 10 minutes pour *Iridium NEXT* qui possède 6 plans orbitaux et 11 satellites par plan orbital, comme indiqué dans le Tableau 2.1. Il en résulte de fréquentes variations de délai.

2.2.4.3 Délai dû au *handover* inter-orbital

La rotation de la terre sur son axe ou le déplacement du terminal terrestre le long de la longitude font que la couverture du terminal terrestre est transférée du satellite de couverture actuel à un autre dans le plan orbital adjacent. Ce transfert entraîne un changement de route et donc fort probablement une variation de délai. Pour des terminaux de communication fixes, le transfert inter-orbital se produit généralement environ toutes les 2 heures pour *Iridium NEXT*.

2.2.4.4 Délai dû au *seam handover*

Une particularité des constellations de satellites polaires ou quasi-polaires est que les satellites du dernier et du premier plan orbital n'ont généralement aucune liaison entre eux (sans liens cross-*seam*). Lorsque des satellites situés sur les deux côtés de la couture — *seam* sont sollicités, des changements de route se produisent. En général, le trafic est re-routé vers les satellites qui se trouvent au-dessus du pôle. La séparation par la couture de deux terminaux fixes communicants se produit au maximum trois fois et au minimum deux fois sur 24 heures, en fonction de leur position par rapport à la couture. Quant à la durée de cette phase, elle dépend de la séparation longitudinale des terminaux communicants.

2.2.4.5 Délai dû aux changements d'ISLs

Pour les constellations de satellites polaires ou quasi-polaires, les liaisons inter-orbitales sont désactivées aux pôles à cause des satellites en rotation rapide qui se croisent. Cela impose un re-routage et entraîne donc des variations de délai.

2.2.5 Conclusion

Dans la Figure 2.5b, nous distinguons les différentes sources de variations de délai dans la constellation à savoir l'élévation, les *handovers* intra- et inter-orbitaux et les variations de délai dues à la couture, respectivement notées par 1, 2, 3 et 4. Nous pouvons voir les différentes transitions de 4 des 5 sources de variations de délai en fonction du mouvement de la terre et de la topologie de la constellation de satellites, pour un terminal fixe représenté ici par une figure humaine. La cinquième source (changements d'ISLs) ne peut pas être représentée ici.

La question qui reste donc en suspens est l'étude de l'impact de ces variations de délai très variées sur les communications de bout-en-bout.

2.3 Évolution de la couche transport

Nous allons nous intéresser maintenant à la couche transport en survolant les contextes, principes et algorithmes historiques de TCP.

Par la suite, nous ferons une brève taxonomie des différents algorithmes de contrôle de congestion destinés aux communications terrestres et ensuite des algorithmes conçus pour un contexte satellite. Et enfin, nous décrirons les problèmes de TCP tant généraux que dans un contexte satellite.

2.3.1 Principe de Transmission Control Protocol (TCP)

2.3.1.1 Contexte

Le modèle Internet (TCP/Internet Protocol (IP)) a été intégré par la communauté en 1983. Il est souvent représenté sous forme de sablier, où la taille est au niveau de la couche réseau, IP étant la composante de base sur laquelle les autres couches reposent, permettant ainsi l'hétérogénéité des applications et la mise en œuvre avec plusieurs solutions protocolaires.

Plusieurs travaux de recherche et de standardisation ont été entrepris sur les différentes couches du modèle TCP/IP, nos travaux s'intéressent plutôt à la couche transport.

La couche réseau permet l'intégration et l'interopérabilité des différentes couches, IP *on everything*, avec une abstraction commune (et bien d'autres fonctionnalités : routage et *forwarding* des paquets). La couche transport, quant à elle, fournit des abstractions de la communication aux applications telles que le multiplexage, la fiabilité (totale/partielle), le contrôle de flux/congestion etc.

TCP est un protocole de la couche transport en mode connecté de bout-en-bout créé dans les années 1980 très largement amélioré, depuis. Il permet la transmission fiable d'informations entre deux utilisateurs.

TCP repose sur un modèle de boîte noire où l'émetteur, une fois l'étape de *handshake* finie, transmet les données vers le récepteur sans connaissance préalable du réseau en termes de capacité et de délai. Le seul *feedback* que l'émetteur puisse avoir s'effectue par des accusés de réception — ACKnowledgement (ACK)s. Les pertes sont donc indiquées implicitement via ces ACKs.

2.3.1.2 Objectif

L'émetteur TCP a pour objectif de maximiser l'utilisation de la bande disponible et donc d'avoir le débit le plus élevé durant la transmission. L'introduction de l'option *timestamp* a permis de conserver une mesure actualisée du RTT. Le RTT représente une métrique qui permet à l'émetteur de dessiner une image plus fidèle de l'état du réseau. Un deuxième objectif de TCP s'ajoute alors, il s'agit de minimiser le délai de bout-en-bout durant une communication.

Le point de fonctionnement idéal serait d'avoir un débit maximal avec un délai minimal. Ce point correspond au point de fonctionnement optimal de Kleinrock [75]. Cela correspond à une configuration dans laquelle les données en vol occupent la bande disponible. Quand ce point est franchi, le *buffer* se remplit. Le délai augmente alors, jusqu'à ce que la limite du *buffer* soit atteinte.

Cette limite dépassée, des pertes commencent à survenir. Les versions classiques de TCP prennent ces pertes comme signe de congestion et commence à réagir.

2.3.1.3 Problème

La connaissance que l'émetteur TCP a du lien s'élabore au fur et à mesure que la communication évolue. Quand l'émetteur tente de sonder le lien en envoyant les données à l'aveugle en attendant à chaque fois les ACKs du récepteur pour ajuster son modèle d'émission, le *buffer* (du lien déterminant et limitant, qu'on appelle le plus souvent *bottleneck*) déborde. Le remplissage de la file d'attente et son affluence se traduisent, par conséquent, par une augmentation du délai de bout-en-bout et une diminution du débit. Au départ, du point de vue émetteur, cela se reflétait par des pertes, mais à cela s'est ajoutée la connaissance de la mesure du délai via le RTT grâce à l'option *timestamp*. TCP utilise ces événements qu'il interprète en tant que notifications de congestion pour établir le débit de la connexion et assurer aux utilisateurs le meilleur débit instantané possible. En effet, il ne possède pas une compréhension complète des spécificités du réseau utilisé, et en particulier le type des liens, le débit disponible ou la congestion réelle éventuelle.

2.3.1.4 Principe

Entre l'établissement et la fermeture de la connexion, le modus operandi de TCP est réparti sur deux états principaux assurant le contrôle de flux et de congestion :

- le début de connexion ;
- l'état permanent.

2.3.1.4.1 Début de connexion

Handshake : L'établissement de connexion est à l'initiative de l'émetteur sous forme de poignée de main — 3-*way handshake* qui dure 1.5× RTT.

Slow Start (SS) : La connexion est établie et le RTT connu. Dans cette étape, TCP essaie de sonder le réseau via le mécanisme du Slow Start (SS) jusqu'à la détection d'un événement de congestion ou au dépassement de la Slow Start Threshold (SSTHRESH), qui détermine si la connexion doit continuer en SS ou passer à l'état permanent [76]. Il est important de noter qu'en SS, TCP commence par émettre le contenu de l'Initial Window (IW) dont la taille a été fixée au départ entre 2 et 4 segments par la Request For Comments (RFC) 3390 [77] puis passée à 10 segments avec la RFC 6928 expérimentale [78] (avec un retour à la recommandation existante de la RFC 3390 lorsque des problèmes de performance sont détectés). À chaque ACK reçu l'émetteur augmentera ainsi d'une unité la taille de sa Congestion Window (CWND), c'est-à-dire du nombre potentiel de segments qu'il peut envoyer sans attendre de nouvel accusé. À chaque RTT, l'émetteur va donc doubler sa capacité d'émission [76].

La reprise sur erreur : Durant cette étape, les pertes sont détectées par l'émetteur. Initialement, c'était par le biais de l'expiration du temporisateur sur la non-réception d'un ACK, le Retransmission TimeOut (RTO) dont la durée a été fixée à 3 secondes par la RFC 2988 [79] puis à 1 seconde par la RFC

6298 [78]. L'expiration du temporisateur entraîne le retour définitif dans l'état précédent (*i.e.* en SS) avec une IW égale à 1 [76].

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 d'une manière plus efficace [76]. Le mécanisme de Duplicate Ack (DUPACK), quand le récepteur envoie à chaque réception d'un segment l'ACK du dernier segment reçu de manière ordonnée, permet d'avertir l'émetteur qu'il y a eu réception de manière déséquencée ou perte d'un segment. Cela se matérialise par la réception d'un ACK en tout point équivalent au précédent [76].

Fast Retransmit : À la réception de 3 DUPACKs, nombre jugé suffisant pour que l'émetteur comprenne que les segments n'arriveront pas de façon déséquencée et que la perte n'était pas due à une congestion majeure du réseau mais à un événement isolé, il retransmet instantanément le prochain segment attendu et effectue une transition vers l'état stable [76].

Fast Recovery : Au passage à l'état stable la CWND subit une diminution significative pour prévenir une potentielle saturation du réseau. *Fast Recovery* permet donc d'atténuer l'effet de la réduction de la fenêtre : pour chacun des DUPACKs reçu la taille de la CWND sera incrémentée d'une unité et ce jusqu'à la réception de l'accusé du segment retransmis [76].

Par conséquent, lorsqu'un segment est perdu, TCP réagit de la sorte : [76]

- soit il est en mesure de récupérer la perte via *Fast Retransmit*, auquel cas il entrera dans l'état permanent ;
- soit il ne le peut pas, et dans ce cas, il reprend en SS après expiration du RTO.

2.3.1.4.2 État permanent

Dans cet état la connexion évolue à un débit élevé, après que des événements de congestion ont été détectés. TCP y est entré après avoir dépassé la SSTHRESH ou lors d'une sortie réussie du mécanisme de *Fast Retransmit*.

Dans cet état, qui est crucial pour les performances de TCP, les algorithmes de Congestion Avoidance (CA) interviennent et ont fait l'objet d'améliorations progressives. Les différentes versions de TCPs portent généralement le nom de leur algorithme de CA [76].

Les algorithmes d'évitement de congestion CA sont toujours un champ de bataille. Ils permettent de faire du contrôle de congestion et donc de détecter et de sortir de la congestion dans le but de reprendre au mieux possible et au plus vite l'utilisation de la capacité de la bande tout en maintenant un comportement proactif assez conservateur pour ne pas submerger le réseau et monopoliser son utilisation. L'idée au départ était d'éliminer le phénomène de *congestion collapse*. Une autre ambition a été ajoutée ensuite, visant à utiliser efficacement les ressources réseau disponibles dans différents types d'environnements (filaire, sans fil, à haut débit, à long délai, etc.)

La Figure 2.6 de l'évolution de la CWND de TCP *New Reno* montre bien la transition entre les différentes phases en retraçant différents événements sur une connexion.

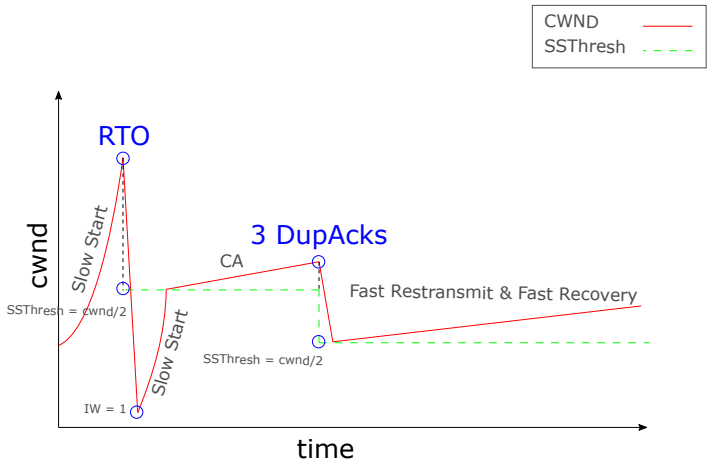


FIGURE 2.6 Évolution de la CWND de *New Reno*

2.3.1.5 Algorithmes historiques

Les algorithmes classiques de contrôle de congestion de TCP tels que *Tahoe* [80], *Reno* [81] et *New Reno* [82] utilisent les pertes de paquets pour détecter la congestion [83]. À la perte d'un paquet, soit lorsque le temporisateur RTO expire, soit après trois DUPACKs consécutifs, le mécanisme Additive-Increase/Multiple-Decrease (AIMD) réduit fortement la CWND. Alors que *Tahoe* est très conservateur, en partant d'une CWND de 1 segment et en entrant dans la phase de *Slow Start*, *Reno* et *New Reno* ne réduisent la CWND que de la moitié dans le cas de trois DUPACKs, comme le montre la Figure 2.6. *New Reno* présente une petite modification, mais qui se révèle importante, de l'algorithme *Reno* au niveau de la phase de *Fast Recovery* : il reste dans ce mode tant que il n'y a pas eu de réception des ACKs de tous les segments perdus. Lorsqu'il y a une perte de plusieurs segments d'une même "rafale", à la réception d'un accusé de réception partiel, le segment perdu suivant est renvoyé, sans sortir du mode *Fast Recovery*, contrairement à *Reno* qui sort de ce mode dès la réception d'un ACK non dupliqué [84]. Suite au *Fast Retransmit* donc, *New Reno* augmente sa fenêtre de façon incrémentale, d'une unité à chaque RTT jusqu'à la prochaine détection d'un événement de congestion, comme illustré dans la Figure 2.6. Il aura donc besoin de $\frac{\text{CWND}}{2}$ RTTs avant de récupérer le débit avant réduction. Ses performances seront donc fonction du RTT avec une accélération du débit inversement proportionnelle à la durée de ce dernier.

2.3.2 TCP dans le terrestre

2.3.2.1 Chronologie des variantes TCP

Les CAs ont fait l'objet de plusieurs modifications et améliorations dans le but d'apporter plus d'adaptabilité selon les types de liens : liens avec de gros Bandwidth-Delay Product (BDP) ; liens avec pertes ; liens à débit variable, et selon les aspects à consolider : équité ; avantages pour les flux courts ; vitesse de convergence. Il existe plusieurs philosophies de conception d'algorithmes de contrôle de

congestion, avec différentes approches de détection et de réaction à un événement de congestion. Certaines taxonomies des mécanismes de contrôle de congestion fort intéressantes ont été faites [83], [85]. Nous allons énumérer les principales philosophies de conception du contrôle de congestion, leurs avantages et inconvénients. Les principaux mécanismes présentés dans cette section sont résumés dans le Tableau 2.3.

Les premiers algorithmes de contrôle de congestion reposaient sur la perte et sont donc appelés *loss-based*. Par la suite, certains algorithmes utilisent le délai comme signe de congestion et sont appelés *delay-based*. Les algorithmes *capacity-based* permettent quant à eux de suivre les changements dans la capacité de la bande. Certains algorithmes tirent le meilleur parti des *loss-based* et *delay-based* ou une combinaison des *capacity-based* et *delay-based*, ils sont donc appelés *hybrides*. Avec l'introduction et l'utilisation courante du *machine learning*, les algorithmes de contrôle de congestion ne font pas exception. Il existe donc des variantes de TCP (néanmoins, pas encore déployées) qui utilisent les outils de *machine learning* et sont dites *learning-based*.

2.3.2.2 Loss-based

2.3.2.2.1 Variantes Loss-based

Les premiers algorithmes de contrôle de congestion se fondaient sur les pertes de paquet comme symptôme de congestion et adoptaient une stratégie AIMD [83]. Comme nous l'avons précisé précédemment, les algorithmes de contrôle de congestion de TCP *loss-based* tels que Tahoe, Reno ou encore New Reno reposent sur les pertes comme *feedback* sur l'état du lien. Les pertes se produisent quand le *buffer* du goulot d'étranglement est plein. TCP interprète ces pertes comme signe de congestion, bien qu'elle puisse survenir avant le débordement du *buffer*.

2.3.2.2.2 CUBIC

Principe. Les mécanismes *loss-based* classiques sont extrêmement inefficaces dans les LFNs (BDP plus de 100kbit). TCP CUBIC [87] est un algorithme de contrôle de congestion développé pour résoudre les problèmes des mécanismes précédents avec les LFNs. Il abandonne l'AIMD pur pour une fonction plus complexe. À partir de la dernière perte, il applique une AIMD puis, comme son nom l'indique, une augmentation suivant une fonction cubique, comme le montre la Figure 2.7.

Slow Start (SS). En SS, CUBIC implante une variante appelée *hybrid slow start*. Cette variante garde elle aussi le mécanisme *slow start* standard de TCP permettant de passer en évitement de congestion (CA) sans subir un nombre trop élevé de pertes de paquets. *Hybrid slow start* utilise deux éléments d'information — la longueur des trains d'ACK et l'augmentation des délais des paquets. En mesurant la longueur des trains d'ACK, un émetteur TCP déduit approximativement le nombre maximum de paquets en vol. L'augmentation des délais pour les premiers paquets dans chaque RTT pendant le *slow start* indique que le chemin est très probablement congestionné.

***Fast Recovery et* CA.** À la détection d'une perte (ce qui correspond à $t = 0$ dans l'équation 2.1) par DUPACK ou ACKs avec Explicit Congestion Notification (ECN)-Echo flags, CUBIC sauvegarde

TABLE 2.3 Évolution des algorithmes de contrôle de congestion de TCP pour un contexte général [83]

TCPs pour un contexte général

Type	Algorithme	Année	Standardisation	Déploiement	Mécanisme de Congestion Control (CC)	Avantages	Inconvénients
Loss-based	New Reno [82]	1999	Standard RFC 2582 (1999) (RFC 3782 (2004), RFC 6582 (2012))	Défaut noyau Linux (<2.6.8) et FreeBSD	AIMD	Convergence et équité, prouvées	Inefficace en Long Fat Networks (LFN)s
	BIC [86]	2004	-	Défaut noyau Linux (>2.6.8-2.6.19)	Fonction binary search increase	Plus grande efficacité	Trop aggressif
	CUBIC [87]	2008	Informational RFC 8312 (2018)	Défaut noyau Linux 2.6.19 (2006), Noyau Windows 10 (2017), macOS (2014), FreeBSD	Fonction cubique CWND	Équitable	Grand # de retransmissions et problèmes d'équité
Delay-based	Vegas [88]	1994	-	-	Changements du RTT comme signe de congestion	Moins de retransmissions, moins de latence	Dominé par les flux loss-based
	LEDBAT [89]	2010	Experimental RFC 6817 (2012)	BitTorrent, mises à jour SW Apple et Windows 10	Délai supplémentaire pour les flux à haute priorité	N'affecte pas les autres flux	Limité au trafic à faible priorité
Capaciy-based	Westwood [90]	2001	-	-	Estimation du débit pour diminuer la CWND	Bon pour les liens sans fil et à perte	Pas de contrôle de la latence
	Sprout [91]	2013	-	-	Modèle de capacité fondé sur HMM	Faible délai, personnalisable	Nécessite un buffer par flux
Hybride	Compound [92]	2006	-	Windows Vista et 8, Windows Server 2008, Noyau Linux < 2.6.17	Somme des fenêtres de Reno et Vegas	Rapide dans les LFNs, équitable envers CUBIC	Pas de contrôle de la latence
	BBR [93]	2016	Experimental RFC (2017/2021)	Trafic Google (BBRv2), BBRv1 disponible sur les différents OS	Mesure de la capacité et du RTT	Débit élevé, faible délai	Problèmes d'équité et de mobilité
Learning-based	Remy [94]	2013	-	-	Politique fondée sur Monte Carlo	Il atteint sa capacité avec un délai faible	Problèmes d'équité avec les autres TCPs

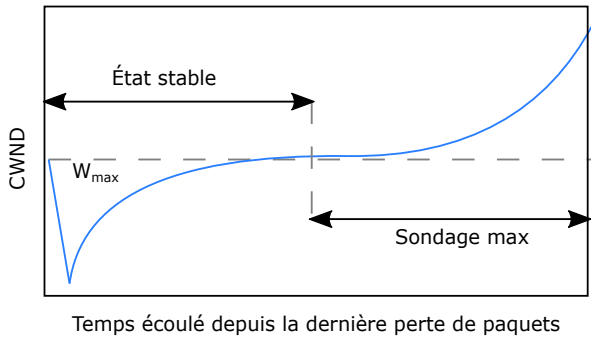


FIGURE 2.7 Évolution de la CWND de CUBIC [83]

la valeur de la dernière CWND dans W_{max}, et l'équation 2.1 devient donc $W(t=0) = (1-\beta)W_{max}$. CUBIC réduit donc sa fenêtre de CWND par un facteur $1-\beta$ (β constante de diminution typiquement à 0.3). La nouvelle CWND sera à 70% de la valeur précédente. Il conserve les phases de *fast retransmit* et *fast recovery* habituelles. La SSTHRESH prend les valeurs CWND$(1-\beta)$ et max(SSTHRESH, 2).

Après être passé de la *fast recovery* à la CA, CUBIC commence à augmenter la fenêtre en utilisant le profil concave de la fonction cubique. Cette fonction est définie pour avoir son plateau (point d'inflexion) à W_{max} comme le montre la Figure 2.7, de sorte que la croissance concave se poursuit jusqu'à ce que la taille de la fenêtre devienne W_{max}. Par la suite, la fonction cubique se transforme en un profil convexe de croissance de la fenêtre.

L'évolution de la fenêtre (CWND) de CUBIC utilise la fonction suivante :

$$W(t) = C(t-K)^3 + W_{max} \tag{2.1}$$

où :

- W_{max} est la taille maximale de la fenêtre avant la dernière réduction ;
- **t** est le temps écoulé depuis la dernière réduction ;
- **C** est le facteur de mise à l'échelle ;
- **K** est la durée que la fonction ci-dessus met pour augmenter W à W_{max} lorsqu'il n'y a pas eu de pertes. K = $\sqrt[3]{\frac{W_{max}\beta}{C}}$, β est un facteur constant de diminution appliqué à la réduction de la fenêtre à l'instant de la perte.

Timeout. En cas de *timeout*, CUBIC suit le standard TCP pour réduire CWND, mais définit SSTHRESH comme dans le cas d'une perte de paquets.

Fast Convergence et équité. Pour améliorer la vitesse de convergence de CUBIC, une heuristique a été introduite. Lorsqu'un nouveau flux rejoint le réseau, les flux existants dans le réseau doivent céder une part du débit pour permettre au nouveau flux de se développer. Pour accélérer cette libération du débit par les flux existants, le mécanisme appelé *fast convergence* a été introduit. Avec la *fast convergence*, lorsqu'un événement de perte se produit, avant une réduction de la fenêtre de congestion, le protocole se souvient de la dernière valeur de W_{max} avant de mettre à jour W_{max} pour l'événement de perte en cours. Appelons la dernière valeur de W_{max}, W_{last_max}. Lors d'un événement

de perte, si la valeur courante de W_{max} est inférieure à sa dernière valeur, W_{last_max}, cela indique que le point de "saturation" de ce débit est en train de se réduire à cause du changement du débit disponible. Ce flux doit libérer plus de débit en réduisant encore W_{max}. Cette action prolonge le temps nécessaire à ce flux pour augmenter sa fenêtre car la réduction de W_{max} force le flux à atteindre le plateau plus tôt. Cela donne plus de temps au nouveau flux pour rattraper la taille de sa fenêtre. La *Fast Convergence* est conçue pour les environnements avec plusieurs flux CUBIC. Avec un seul flux CUBIC et sans autre trafic, la *fast convergence* devrait être désactivée. Nous avons remarqué néanmoins que dans l'implantation ce n'est pas le cas. C'est pour le moins compréhensible, cela implique une saturation (par abus de langage pour dire congestion) à W_{max}, qui est égale à CWND_en_CA, < W_{last_max} implique encore une réduction de W_{max} à $W_{max}(2-\beta)/2$.

Adoption, limites et déploiement. Binary Increase Control (BIC) a été jugé trop agressif et inéquitable envers les anciens flux [95], CUBIC est maintenant largement déployé [96]. C'est l'algorithme de contrôle de congestion par défaut dans le noyau Linux depuis la version 2.6.19. Il a fait l'objet d'une RFC [97]. Il peut atteindre des performances élevées sans être inéquitable pour les flux utilisant les anciens mécanismes de contrôle de congestion et les flux CUBIC.

2.3.2.3 Delay-based

Le seul signal de congestion que les mécanismes *loss-based* perçoivent est, comme son nom l'indique, la perte de paquets. Cela remplit le *buffer* du lien *bottleneck* et peut créer ce qu'on appelle un *bufferbloat* [83].

Une solution possible à ce problème est le contrôle de congestion *delay-based* : les pertes peuvent être un signe de congestion, mais la congestion peut se produire bien avant que le *buffer* ne soit plein, et sa détection précoce peut augmenter la réactivité du contrôle de congestion. Il peut alors faire un *back off* dès que le RTT augmente de manière substantielle et éviter des baisses brutales du débit et une latence élevée.

Les protocoles dits *delay-based* incluent donc l'information sur le délai comme un signe potentiel de congestion, avec une adaptation de la fenêtre de congestion aux exigences de fonctionnement.

TCP *Vegas* [98] a été la première implantation utilisant le délai comme signe de congestion. Après une phase de *Slow Start* plus douce [99], il ajuste la fenêtre de congestion au débit attendu du canal. Le canal est supposé être sous-utilisé si le RTT est proche du minimum mesuré. Dès que le RTT dépasse un certain seuil, le protocole effectue un *back off*.

Une taxonomie plus aboutie et détaillée des algorithmes *delay-based* peut être trouvée dans [83].

En général, les mécanismes *delay-based* sont plus stables, retransmettent moins de segments et ont des latences plus faibles que les mécanismes *loss-based*, avec un débit similaire dans des conditions réalistes. Cependant, leur adoption a été bloquée en raison de leur faible agressivité. La majorité des serveurs d'Internet utilisent un mécanisme *loss-based* de contrôle de congestion, et la séparation du trafic *loss-based* du trafic *delay-based* est extrêmement coûteuse et nécessite des modifications importantes. Les protocoles *delay-based* n'ont jamais été déployés à grande échelle.

2.3.2.4 Capacity-based

TCP *Westwood* [90] était un mécanisme de contrôle de congestion de 2001 qui suivait explicitement la largeur de bande estimée afin de modifier de manière adaptative la CWND après une perte, tout en conservant le schéma d'augmentation additive de TCP *Reno* [83]. *Westwood* adapte la CWND à la capacité mesurée de la connexion après les pertes et obtient une meilleure performance [100] que les schémas *loss-based* dans ces conditions, ainsi que dans les réseaux filaires.

L'idée d'un contrôle de congestion *capacity-based* a été récemment relancée par *Sprout* [91], un protocole de bout-en-bout conçu pour les réseaux cellulaires qui vise à être un compromis entre l'obtention du débit le plus élevé possible et la prévention des longs délais de paquets dans une file d'attente du réseau. Plus de détails sur l'algorithme peuvent être trouvés dans [83].

2.3.2.5 Hybride

2.3.2.5.1 Variantes hybrides

Certains mécanismes de contrôle de congestion cherchent à combiner deux approches pour bénéficier de leurs avantages respectifs. *Compound* TCP [92] est un exemple bien connu, car il a été l'algorithme par défaut sur toutes les machines *Microsoft Windows* avant que CUBIC ne le surclasse. *Compound* utilise la somme d'une CWND *delay-based* et d'une CWND *loss-based Reno* comme CWND, et en tant que tel, il peut traiter les LFNs mieux que les mécanismes purement *loss-based* tout en maintenant une mesure d'équité envers eux. En effet, sa fenêtre croîtra plus rapidement qu'une fenêtre purement *loss-based*, mais le comportement du protocole reviendra au contrôle de congestion *loss-based* lorsqu'il détectera une congestion [83].

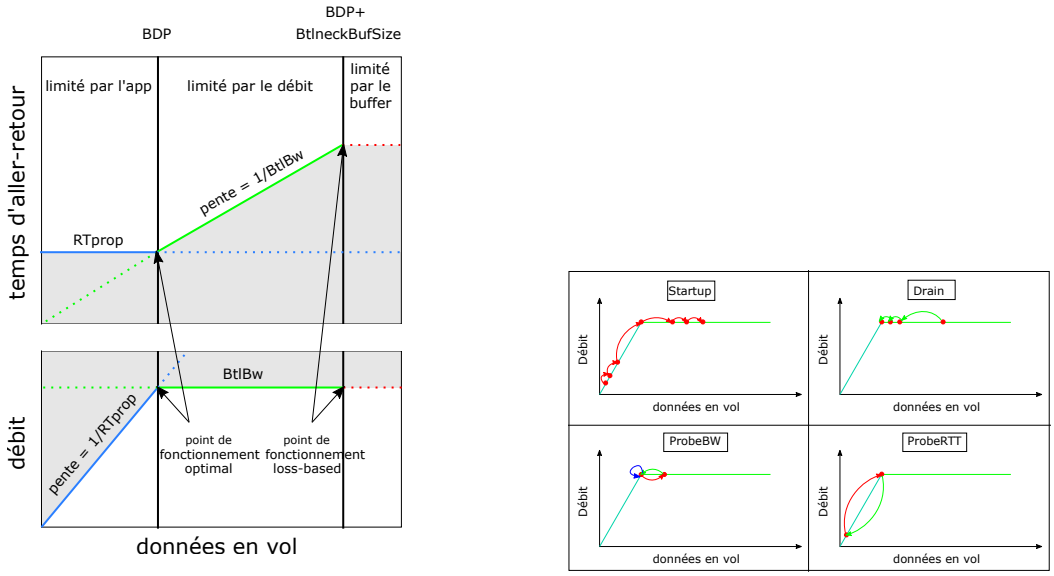
D'autres algorithmes sont plus détaillés dans [83].

2.3.2.5.2 Bottleneck Bandwidth and Round-trip propagation time (BBR)

Principe. Le tout dernier des mécanismes hybrides de contrôle de congestion est sans aucun doute TCP BBR avec ses deux versions en 2016 la v1 [93], [101] et en 2019 la v2 [102] de *Google*, qui adopte une approche différente en essayant d'estimer à la fois le débit et le RTT [93] comme son nom peut l'indiquer. Il essaie de maintenir un équilibre entre débit et délai en atteignant le point optimal de fonctionnement [75] en maintenant des données en vol égales au BDP. Toute augmentation supplémentaire des données en vol ne fait qu'augmenter le délai sans aucun avantage en termes de débit, comme le montre la Figure 2.8a.

Objectif. L'objectif principal de BBR est de garantir que le *bottleneck* reste saturé mais non congestionné, ce qui permet d'obtenir un débit maximal avec un délai minimal. BBR estime le RTT minimal — Round-Trip propagation time (RTPROP) en vidant périodiquement le *buffer* durant *ProbeRTT* et sonde le débit disponible durant *ProbeBW* — Bottleneck Bandwidth (BTLBW). Ce mécanisme est illustré dans la Figure 2.8b.

Machine à états. BBR utilise une machine à états de sondage séquentiel qui alterne entre le



(a) Principe de base de fonctionnement de BBR [93] **(b)** Les phases de fonctionnement de l'algorithme de BBR [103]

FIGURE 2.8 Principe de fonctionnement et phases de fonctionnement de BBR

sondage pour plus de débit et le sondage pour un RTT plus faible. L'algorithme de BBR repose sur une machine à états qui comporte quatre phases différentes [101] : Démarrage — ***Startup***, Drainage — ***Drain***, et le régime permanent — ***steady-state*** composé de deux états Sondage du débit — ***Probe Bandwidth***, et Sondage du RTT — ***Probe* RTT** comme illustré dans la machine à états des Figures 2.8b et 2.9 :

- **Startup** : La première phase d'entrée, comme le montrent les Figures 2.8b et 2.9, adopte le comportement exponentiel de CUBIC en doublant le débit d'émission à chaque RTT pour estimer BTLBW. Le cap de données en vol de 3 BDP de BBR peut créer jusqu'à 2 BDP de file d'attente supplémentaires. BBR effectue une recherche binaire du débit. Cela permet de trouver BTLBW très rapidement.BBR essaie de déterminer si le lien est plein en recherchant un plateau dans l'estimation de BTLBW. S'il constate qu'il y a plusieurs RTT où les tentatives de doubler le débit n'aboutissent en fait qu'à une faible augmentation (moins de 25% sur 3 RTTs consécutifs), il estime qu'il a atteint BTLBW et sort donc de *Startup* pour entrer dans *Drain*. Dans les faits, on choisit une valeur de 3 RTTs afin d'avoir des preuves solides que l'émetteur ne détecte pas un plateau du débit d'émission qui a été temporairement imposé par la Receiver Advertised Window (RWND). Cela permet donc aussi au récepteur de régler sa fenêtre de réception. Mais comme cette observation est retardée d'un RTT, une file d'attente a déjà été créée au niveau du *bottleneck* et une fois que le lien est plein, la file d'attente est à ~ 2BDP (vu que le cap de données en vol est de 3 BDP).
- **Drain** : Dans cette phase, BBR essaie de se débarrasser rapidement de la file d'attente créée dans la phase de *Startup*, en un seul RTT. Lorsque le nombre de paquets *inflight* correspond au BDP estimé, c'est-à-dire que BBR estime que la file d'attente a été entièrement vidée mais que

le lien est encore plein, BBR quitte la phase *Drain* et entre, si possible, dans *Probe Bandwidth*. Par la suite, dans ce qui est appelé *steady state*, un flux BBR n'utilise que l'une des deux phases : *Probe Bandwidth* ou *Probe* RTT.

- **Probe Bandwidth** : BBR entre dans la phase de *Probe Bandwidth* (où il passe la grande majorité de son temps, ≈ 98%) dans laquelle il cherche à obtenir plus de débit disponible avec de faibles délais de mise en file d'attente, et une convergence vers un partage équitable du débit. Ceci est effectué, en *gain cycling*, sur 8 cycles, chacun durant normalement un RTPROP. Tout d'abord, un *pacing_gain* (coefficient multiplicateur d'adaptation du débit d'émission) est fixé à 1.25, ce qui permet d'obtenir plus de débit, puis à 0.75 pour vider les files d'attente créées. Pour les 6 cycles restants, BBR fixe le *pacing_gain* à 1, pour garder une file d'attente courte et une utilisation totale de la bande. Il faut noter que si le *gain cycling* fait varier la valeur du *pacing_gain*, le *cwnd_gain* (coefficient multiplicateur d'adaptation de la CWND) reste constant à 2 pour permettre à BBR de poursuivre ses émissions au débit d'émission estimé, même lorsque les ACKs sont retardés d'un RTT (Delayed Ack (DELACK)s ou *Stretched* ACKs). Les cycles se répètent s'il n'y a pas eu de pertes ou besoin d'estimer une nouvelle valeur de RTPROP.

  Si aucune nouvelle valeur minimale de RTPROP n'a été mesurée pendant plus de 10s en dehors de la phase *Probe* RTT, BBR entre en phase de *Probe* RTT.
- **Probe RTT** : Pendant cette phase, la CWND est réduite à 4 paquets pour vider toute la file d'attente éventuelle et obtenir une estimation réelle du RTT. Cette phase est maintenue pendant 200ms + RTT. RTPROP est mis à jour et reste valable pendant 10 secondes [104]. BBR passe ensuite dans la phase *Startup* ou *Probe Bandwidth*, selon qu'il estime que le lien est déjà rempli ou non.

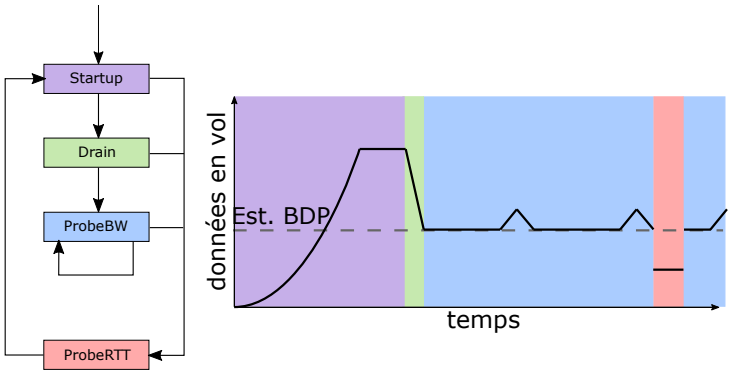


FIGURE 2.9 Machine à états de BBR [102]

***Limites de* BBR*v1*.** BBRv1 présente quelques limitations telles que son agressivité, son iniquité en cas de RTTs différents et envers les flux *loss-based* [104]-[108].

BBR*v2* dont la toute dernière version *draft* [109] vient de voir le jour au moment de la rédaction du manuscrit, tente de traiter le problème d'iniquité envers les flux *loss-based*. Le modèle utilisé par BBRv1 est augmenté pour inclure des informations sur la perte de paquets et des informations provenant de l'ECN. Dans BBRv2, les paramètres de modèle de BBRv1 BTLBW, RTPROP sont donc

augmentés par *max_aggregation* et *max_inflight* (fondés sur les pertes et ECN). BTLBW et *inflight* peuvent être très dynamiques en fonction du trafic. BBRv2 maintient à la fois des estimations à court et à long terme pour chacun de ces paramètres.

Les objectifs restent les mêmes pour ces estimations : débit élevé, même en cas de perte aléatoire modérée, et faible pression sur les files d'attente (faibles délais de mise en file d'attente, faibles taux de perte de paquets).

Pour maintenir ces objectifs deux mécanismes sont implantés pour :

- S'adapter le plus souvent rapidement pour maintenir l'équilibre des flux et laisser de la marge de manœuvre, en mesurant {*bw*, *inflight*}*_lo* qui représentent les limites à court terme en utilisant les derniers signes de *delivery*/perte/ECN.
- Sonder périodiquement au-delà de l'équilibre des flux pour regarder de manière robuste des débits plus élevés, des données en vol en mesurant {*bw*, *inflight*}*_hi* qui représentent les valeurs max à long terme mesurées avant les signes de congestion (perte, ECN).

Les variables d'états *inflight* à court *inflight_lo* et long terme *inflight_hi* respectivement, correspondent à l'équivalent de SSTHRESH et la fenêtre maximale de congestion observée jusqu'à présent CWND_{max}. Ce qui permet de laisser la place aux autres flux partageant le lien [110].

Les différence majeures entre la version 1 et la 2 annoncées lors de la présentation de l'Internet Engineering Task Force (IETF) 104 [102], ont été regroupées dans ce Tableau 2.4.

Plus de précisions sur l'algorithme de BBRv2 et les différences majeures avec la v1 se trouvent dans [102], [109].

Bien que BBRv2 puisse parfois conduire à un débit inférieur à BBRv1, il est généralement considéré comme ayant un meilleur débit utile — *goodput* [111].

En 2019 dans la présentation de l'IETF pour annoncer la version BBRv2 [102], il a été montré que la nouvelle version remédie à la plupart des défaillances de l'ancienne.

***Limites de* BBR*v2*.** Les problèmes connus, jusqu'à présent, et qui ont été soulevés [112] sont l'équité intra- et inter-protocolaire en fonction de la capacité du *buffer*. [110], [113] présentent une évaluation expérimentale de BBRv2, en utilisant *Mininet* et un banc de test. Les résultats montrent que la coexistence entre BBRv2 et CUBIC est améliorée (mais l'iniquité persiste) par rapport à celle entre BBRv1 et CUBIC. Ils montrent également que BBRv2 atténue le problème d'iniquité du RTT observé dans BBRv1, ainsi que l'iniquité et l'agressivité avec des *buffers* inférieurs à 1 BDP.

2.3.2.6 Learning-based

Avec l'introduction et l'utilisation frénétique du *machine learning*, les algorithmes de contrôle de congestion ne font pas exception [114]. Il existe donc des variantes de TCP qui utilisent de tels outils. Il existe déjà plusieurs mécanismes fonctionnels dans la littérature [83].

TCP *Remy* [94] est le premier exemple de contrôle de congestion *machine learning-based*. Le modèle prend en entrée le rapport entre le RTT le plus récent et le RTT le plus bas mesuré pendant la connexion, une Exponentially Weighted Moving Average (EWMA) des temps d'arrivée des derniers ACKs et une EWMA des temps d'émission de ces mêmes paquets. Le mécanisme en lui-même est essentiellement fondé sur une simulation de *Monte Carlo*, et présente certaines limites, puisqu'il

TABLE 2.4 Différences majeures entre BBRv1 et v2 [102]

	BBRv1	**BBRv2**
Paramètres du modèle de la machine à états	BTLBW, RTPROP	BTLBW, RTPROP, *max_aggregation, loss_rate*
Pertes	ND	Valeur seuil du taux de perte
Condition de sortie de Startup	*Slow-start* jusqu'à un plateau de débit	*Slow-start* jusqu'à un plateau de débit ou taux d'ECN/perte > seuil
Équité	Faible débit pour les flux *loss-based* tels que *Reno*/CUBIC partageant un *bottleneck* avec des flux en rafale.	Adaptation du sondage de la bande passante pour une meilleure coexistence avec *Reno*/CUBIC.
Réponse aux pertes	Indifférence aux pertes ; taux élevé de perte de paquets si la file d'attente du *bottleneck* < 1.5×BDP.	Utilisation des pertes comme un signal explicite, avec un plafond de taux de perte cible explicite.
ECN	ND	Utilisation du système ECN de type DCTCP/L4S, si disponible, pour aider à maintenir des files d'attente courtes.
Réponse à l'agrégation	Faible débit pour les chemins à haut degré d'agrégation > 1×BDP de données ou ACKs (*e.g.* WI-FI).	Estimation du degré récent d'agrégation (via *max_aggregation*) pour correspondre à un débit proche de celui de CUBIC.
ProbeRTT	Sondage toutes les 10s et CWND réduit à 4 paquets.	Sondage toutes les 5s et réduction du CWND à 50% du BDP.
Durée du sondage	Durée fixée à 8 cycles.	Durée adaptative (*probe_wait* valeur aléatoire entre 2 et 3s).

nécessite une certaine connaissance préalable du scénario du réseau et fonctionne sur l'hypothèse de base que tous les flux concurrents l'utilisent.

Une discussion plus détaillée sur les considérations de conception de nouveaux mécanismes de contrôle de congestion fondés sur le *machine learning* peut être trouvée dans [115] et un autre papier aussi important avec des simulations à l'appui [116]. Plus d'algorithmes *learning-based* sont détaillés dans [83].

2.3.3 TCP dans les satellites

TCP a bien évidemment été conçu pour les réseaux terrestres filaires. Les possibilités de l'utilisation de TCP dans le contexte satellite se caractérisent soit par la conception de versions TCP dédiées ou bien par l'adaptation de celles existantes. Vers les années 1990 et 2000, plusieurs efforts de recherche

ont été entrepris en ce sens dans les communications par satellite en particulier avec le regain d'intérêt dans les constellations de satellites [1], [3]-[5], [8], [13]-[15], [117]. Dans ce qui suit, nous allons aborder quelques variantes de TCP proposées spécifiquement pour les communications par satellite.

2.3.3.1 Loss-based

TCP *Noordwijk* [118], est une version *cross-layer* de TCP qui exploite les particularités des liaisons par satellite. La principale caractéristique de TCP *Noordwijk* est qu'il transmet les données en rafales — *bursts* à la capacité maximale du goulot d'étranglement — *bottleneck*, en supposant une grande mémoire tampon — *buffer* au niveau du *bottleneck* (comme c'est souvent le cas pour les liaisons par satellite) et sans tenir compte des flux concurrents. L'équité est assurée par un ordonnancement approprié des *bursts* de transmission. Une proposition récente, appelée TCP *Wave* [120], étend le protocole en supprimant les aspects *cross-layer* et en adaptant l'algorithme à tout type de liaison.

TCP *Hybla* [119] a été proposé pour remédier aux faiblesses de certaines versions de TCP telles que *Reno* et *New Reno* dans un environnement où elles sont confrontées à de longs RTT, et donc rendre TCP indépendant du RTT durant la phase de contrôle de congestion. Son objectif est 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 *pacing* des segments pour atténuer les conséquences sur le réseau de l'émission d'un nombre de segments plus important [76].

L'utilisation d'*Hybla* a permis de réelles améliorations du débit pour les communications par satellite [121].

TCP *Hybla* présente de bonnes performances sur les liaisons GEO par rapport aux autres protocoles. Cependant, cette variante présente quelques inconvénients qui ont réduit son intérêt pour la communauté "réseau" [76], [122]. On lui reproche son manque d'équité envers les flux utilisant des algorithmes de contrôle de congestion différents. Ce manque d'équité exige que les liens satellites soient isolés du reste du réseau, avec des Performance Enhancing Proxy (PEP), la partie satellite n'autorisant que les flux *Hybla*. Le RTT de référence, exigé dans l'algorithme de *Hybla*, 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.

2.3.3.2 Capacity-based

TCP *Peach* [2] a été proposé pour les réseaux satellite afin de contrecarrer l'effet des délais. Il introduit les notions de *Sudden Start* et de *Rapid Recovery*, qui remplacent les schémas habituels de *Slow Start* et de *Fast Recovery*. Il envoie également des segments fictifs dans le réseau pour mesurer sa capacité.

TCP *Peach* présente de bonnes performances sur les réseaux satellite sans dégradation de l'équité [122].

TABLE 2.5 Évolution des algorithmes de contrôle de congestion de TCP pour les satellites

TCPs pour les satellites

Type	Algorithme	Année	Standardisation	Déploiement	Mécanisme de CC	Avantages	Inconvénients
Loss-based	Noordwijk [118] (Wave) [118]	2008 (2017)	-	-	Burst-based adaptation	Equitable et efficace	RTT très volatile
	Hybla [119]	2004	-	-	Indépendant du RTT, avec un flux de référence pour RTT	Augmentation + rapide de la CWND pour compenser la réduction du débit due à un long RTT	Problèmes d'équité, RTT référence considéré égal à RTT classique terrestre, si $RTT_{mesure} < RTT_{ref}$: Hybla n'a aucun effet
Capacity-based	Peach [2]	2001	-	-	*Sudden Start*, *Rapid Recovery* et segments fictifs	Equité maintenue sur les liens satellites	Modification du récepteur : champ TOS

Cependant, il nécessite une légère modification du récepteur afin de prendre en compte le champ Type Of Service (TOS) et de gérer ensuite la priorité des segments fictifs.

2.3.4 Chronologie des variantes TCP

Après avoir décrit les différents types d'algorithmes de contrôle de congestion de TCP tant dans un contexte terrestre que dans un contexte satellite une frise chronologique est représentée sur la Figure 2.10. L'écosystème et le paysage de TCP ont bien changé au fil du temps. On voit, sur la

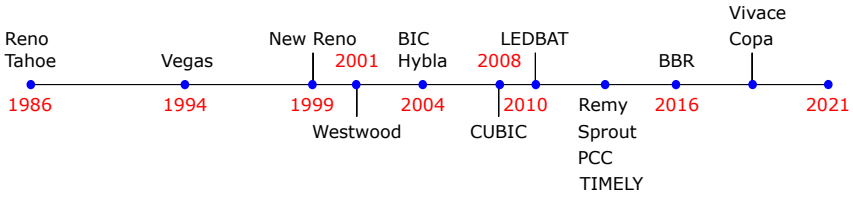


FIGURE 2.10 Évolution des algorithmes de contrôle de congestion de TCP [23]

Figure 2.11, que BBR et CUBIC figurent parmi les algorithmes de contrôle de congestion de TCP les plus prisés par les applications. Cette tendance est également confirmée dans [23].

Dans la famille *loss-based*, c'est désormais CUBIC qui domine et de façon similaire, c'est BBR qui se détache pour les algorithmes hybrides. Nous allons donc les choisir en tant qu'algorithmes candidats pour la suite de la thèse.

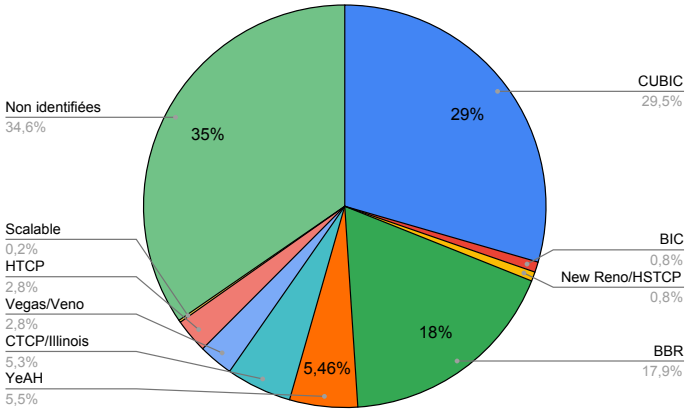


FIGURE 2.11 Répartition des variantes de TCP [23] en 2019 à Paris, la tendance est quasi la même pour d'autres endroits

2.3.5 Problèmes de TCP dans le terrestre

2.3.5.1 Congestion

Les ressources du réseau sont limitées, notamment le temps de traitement des routeurs mais surtout le débit des liens. Lorsque ces équipements se trouvent submergés cela crée des congestions. Ce phénomène se manifeste par une réduction de la qualité de service. Une augmentation du trafic

(de flux) provoque un ralentissement global du réseau avec un risque de paralysie et d'effondrement, si aucune contre-mesure n'a été prise.

Cet événement se caractérise par des retransmissions intempestives, des paquets non remis, des fragmentations, des délais et/ou gigues.

Solutions :

- Les techniques de contrôle de congestion et d'évitement de la congestion (CA) de TCP permettent de moduler l'entrée du trafic et donc d'éviter l'effondrement du réseau.
- Une autre solution serait d'utiliser l'option ECN. Cette modification explicite de congestion doit être mise en place avec de l'Active Queue Management (AQM) une extension qui ajoute un mécanisme de contrôle permettant au routeur de notifier les extrémités communicantes de la possibilité d'un événement de congestion.

D'autres mécanismes ne relevant pas non plus de la couche transport permettent d'atténuer et prévenir la congestion tels que l'AQM, entreprise au niveau du *Network scheduler*, qui réordonne ou abandonne sélectivement les paquets en cas de congestion.

2.3.5.2 Equité

La commutation de paquets induit un partage dynamique des ressources du réseau. L'équité est atteinte lorsqu'un flux ne reçoit qu'une part juste du débit disponible du lien. Ceci présente un problème pour les protocoles de transport en raison de la grande diversité des solutions TCP qui coexistent. La diversité de chemin dans le réseau implique une inégalité des délais et par suite des RTTs. L'étude de l'équité intra- et inter-protocolaires a fait l'objet de plusieurs études [123] - [125].

Solution : l'utilisation des techniques et mécanismes de *scheduling* d'AQM permet de garder un lien équitablement réparti. Il est important de noter que pour certaines variantes de TCP la taille et le dimensionnement du *buffer* joue un rôle essentiel dans la garantie de l'équité, notamment pour BBR.

2.3.5.3 Latence

Certaines applications sont indifférentes aux délais mais dernièrement l'attention est plutôt tournée vers celles qui sont exigeantes en termes de latence. La latence dans TCP est principalement due aux éléments suivants :

- **L'établissement de la connexion :** dans TCP pour initier une connexion entre deux extrémités cela se passe en 3 temps appelés le 3-*way handshake* (SYN, SYN-ACK, et ACK) qui comptent pour 3/2 du RTT ce qui, pour des flux *short-lived* ou des réseaux avec un RTT considérable, est contraignant au vu des délais induits.
- **La *bufferisation* :** les *buffers* coûtent de moins en moins chers, on a souvent tendance à les surdimensionner. Cela peut sembler rationnel et efficace en contre-mesure préventive à la congestion et les rejets de paquets qui en découlent. Cependant, cela n'a pas donné l'effet escompté mais bien le contraire. Les temps de séjour (*bufferisation*) peuvent devenir considérables au travers de ce qu'on appelle *bufferbloat* et la gigue explose [126] - [129].
- **L'*overhead* :** Comme TCP fonctionne en mode connecté pour permettre la fiabilisation de la

connexion, il y a un en-tête plus important de 20B, contrairement à User Datagram Protocol (UDP) qui est en mode non-connecté et n'a donc que 8B d'en-tête. De plus, TCP utilise les ACKs pour faire évoluer sa fenêtre de congestion. Ils induisent un *overhead* important.

Solutions :

- **Établissement de connexion :** Parmi les solutions qui permettent de réduire le temps d'établissement de connexion il y a le TCP Fast Open (TFO) [130] qui est un mécanisme optionnel dans TCP permettant aux points communicants qui ont établi une connexion TCP complète dans le passé d'éliminer un RTT du *handshake* et d'envoyer des données immédiatement. Il est particulièrement bénéfique sur les réseaux à forte latence où le temps avant le premier octet — *time-to-first-byte* est critique et pour les flux courts. TFO permet, le cas échéant, aux données d'être transportées dans les paquets SYN et SYN-ACK et consommées par le récepteur pendant le *handshake* initial de la connexion. On économise ainsi jusqu'à un RTT par rapport au TCP standard. Ceci est très similaire au 0-RTT de TLS 1.3. Une solution possible pourrait être le Multipath TCP (MPTCP) [131] qui permet d'avoir plusieurs connexions TCP ouvertes sous forme de sous-flux où chaque sous-flux représente une connexion TCP classique, avec simultanément l'éclatement que cela offre. MPTCP peut utiliser efficacement plusieurs chemins dans une seule connexion de transport et maintenir une connexion logique établie lorsque l'adresse d'une extrémité communicante change. Outre les gains en latence et de débit obtenus grâce au multichemin, il est possible d'ajouter ou de supprimer des liens lorsque l'extrémité communicante entre ou sort de la couverture sans interrompre la connexion TCP de bout-en-bout. Ce transfert simplifié est géré par abstraction dans la couche transport, sans aucun mécanisme spécial au niveau du réseau ou de la couche de liaison.
- **La *bufferisation* :** Une solution à la *bufferisation* excessive est le *pacing* pour éviter les transmissions sporadiques et pour faire correspondre le taux d'arrivée des paquets au taux de départ du lien, et donc éviter de provoquer des pertes au niveau des paquets *bufferisés*. Une autre solution consiste en une conception judicieuse du *buffer* ; plusieurs travaux s'y sont attaqués et continuent avec les évolutions des différentes variantes de TCP [126]-[128]. La plupart ont montré qu'une taille du *buffer* fixée au BDP n'est pas une bonne pratique. Il n'y a pas besoin d'autant de mémoire pour subvenir aux besoins des flux ni compromettre l'occupation de la totalité du débit. Néanmoins, certaines variantes de TCP, notamment BBR dépendent étroitement du dimensionnement du *buffer* et cela peut entraîner des soucis d'iniquité. De toute évidence, l'utilisation d'une AQM est toujours une bonne pratique pour limiter et prévenir les débordements de *buffer* et ainsi éviter le *bufferbloat*.
- **L'*Overhead* :** La réduction de l'*overhead* ou plutôt l'optimisation de l'utilisation de la bande avec les différents mécanismes tels que DELACK et Selective ACK (SACK) peuvent constituer de bonnes pistes.

2.3.6 Problèmes de TCP dans les satellites

Malgré les pistes précédentes, il subsiste des phénomènes qui risquent d'entraver le bon fonctionnement et les performances des protocoles de la couche transport [77], [83], [132]-[135], notamment

avec des constellations de satellites.

Les constellations de satellites sont caractérisées par d'importantes variations de délais que ce soit en raison du routage au travers de la constellation elle-même dans le cas d'une connectivité ISL via des liaisons radio ou optiques ou dans une moindre mesure à la variation d'élévation du satellite. Ces phénomènes, ajoutés aux perturbations liées à la couche accès (gestion des ressources et *handover*) vont influencer les performances des protocoles de transport.

2.3.6.1 Long Fat Networks (LFN)

Une constellation de satellites LEO peut être considérée comme un LFN, car son BDP est supérieur à 10^5 bits. Ce BDP élevé a un fort impact sur les performances de TCP. [122]

2.3.6.1.1 Taille maximale de la fenêtre

La taille maximale de la fenêtre, qui est de 65535 octets, peut être atteinte. Ce problème a été corrigé [132], permettant des fenêtres plus grandes. Ce mécanisme est maintenant activé dans les principaux systèmes d'exploitation.

2.3.6.1.2 Accusés de réception cumulatifs

Lorsqu'une perte de paquet est détectée, TCP n'a aucune information sur la bonne réception des paquets suivants, ce qui peut conduire à des retransmissions inutiles, diminuant ainsi les performances. Ce problème a été résolu avec l'introduction de SACK, qui informe l'émetteur des paquets qui doivent vraiment être retransmis.

2.3.6.2 Variation de délai

Les satellites, et plus précisément les constellations de satellites qui font l'objet de notre étude sont sujettes à des variations de délai avec divers ordres de grandeur et fréquences d'apparition. Ceci ne tient compte que des topologies "simples", tandis que certaines nouvelles constellations de satellites mettent en jeu deux étages (si ce n'est plus) d'orbites, comme nous l'avons précisé dans le Tableau 2.2. Cela entraîne les problèmes suivants.

2.3.6.2.1 Estimation du Round-Trip Time (RTT)

Le Round-Trip Time (RTT) est difficile à estimer, ce qui entraîne une baisse des performances du *timeout* TCP. Avec une piètre estimation du RTT, TCP ne peut pas réagir correctement aux changements des conditions de trafic. En particulier lorsque des pertes de paquets se produisent et que des retransmissions ont été envoyées, il est alors difficile de savoir si la réponse concerne la première instance du paquet ou une instance ultérieure (une retransmission). L'option TCP *timestamp* a été introduite pour contrer ce problème et lever l'ambiguïté concernant l'instance du segment de

données qui a déclenché l'accusé de réception et permettant ainsi une meilleure approximation du RTT [132].

2.3.6.2.2 Déclenchement du Retransmission TimeOut (RTO)

Une autre variable fonctionnelle de TCP est le temporisateur Retransmission TimeOut (RTO). Cette variable, dont la durée a été fixée à 3 secondes par la RFC 2988 [79] puis à 1 seconde par la RFC 6298 [78], permet de détecter une perte à la fin d'un flux en utilisant le *timeout*. Si aucun ACK n'a été reçu à l'expiration du temporisateur, le paquet est considéré comme perdu et il est retransmis. Cette valeur du temporisateur est calculée et mise à jour pendant la transmission, afin de s'adapter à l'évolution du RTT et de la gigue, et de réagir au plus vite à une perte sans provoquer de retransmissions intempestives.

2.3.6.2.3 Déséquencement

Le déséquencement des paquets peut être causé par les changements de délais inhérents à la constellation de satellites. Ce phénomène provoque de multiples DUPACKs avant l'arrivée du ou des paquets manquants ou de plusieurs retranmissions. L'utilisation de l'option SACK [136] permet de réduire le nombre de retransmissions en ré-émettant que les paquets qui ont été détectés comme perdus. Cependant, SACK n'a pas d'effet sur la cause des paquets déséquencés.

2.4 Conclusion

Dans ce chapitre, nous avons abordé les différentes caractéristiques du contexte satellite. Nous avons également étudié les caractéristiques principales de TCP, les algorithmes de contrôle de congestion avec une attention spéciale à ceux conçus pour un contexte satellite. Enfin, nous nous sommes penchés sur les problèmes de TCP en particulier dans un contexte satellite. Dans ce qui suit, nous allons considérer CUBIC et BBR, dans le contexte de la constellation de satellites type que nous avons choisie. Rappelons que pour la suite de la thèse, nous allons retenir une orbite LEO circulaire quasi-polaire. Le type de constellation retenu est la constellation-π qui intègre des ISLs. Le contexte applicatif est celui des télécommunications. Donc pour tout ce qui suit, la constellation de satellites *Iridium NEXT* va servir de modèle

CHAPITRE 3

ENVIRONNEMENT SYSTÈME ET PARAMÈTRES GÉNÉRIQUES

3.1 Objectif

Nous allons désormais décrire l'environnement système de nos travaux de recherche, ce qui nous permettra de nous positionner par rapport à cet état de l'art particulièrement foisonnant.

Pour le système retenu, nous en présenterons les différents paramètres ainsi que des scénarios qui nous serviront pour toute la suite de la thèse.

Nous commencerons par la constellation choisie, ses spécificités et leurs impacts sur la variation de délai, ainsi que par les algorithmes TCP comparés.

Les différents outils utilisés seront finalement décrits.

3.2 Environnement système

3.2.1 Constellation retenue

Nous avons choisi une orbite LEO circulaire quasi-polaire. Le type de constellation retenu est la constellation-π qui détient des ISLs offrant ainsi plus de connectivité. L'application de la constellation est celle des télécommunications. Parmi les constellations de satellites existantes présentées dans le Tableau 2.2, nous avons choisi la constellation de satellites *Iridium NEXT*. Elle remplit toutes les conditions préalablement énoncées et de plus, elle est en plus complètement déployée. Cette constellation de satellites peut donc être utilisée, selon nous, comme une constellation représentative et les travaux pourraient être généralisés pour les constellations similaires en cours de déploiement ou les projets futurs.

Le Tableau 2.1 résume les paramètres orbitaux de la constellation de satellites *Iridium NEXT* qui servira comme point d'ancrage. Ses principaux paramètres sont :

- 6 plans orbitaux ;
- 11 satellites / plan orbital ;

- 4 ISLs par satellite ;
- Pas de *cross-seam* ISLs ;
- LEO à une altitude de. 780km

Comme nous l'avons précédemment mentionné, les constellations LEO sont connues pour avoir un délai de propagation plus faible que les constellations MEO ou GEO. Cependant, elles sont soumises à des variations de délai plus importantes en raison du mouvement des satellites par rapport au terminal au sol et à la topologie de la constellation elle-même.

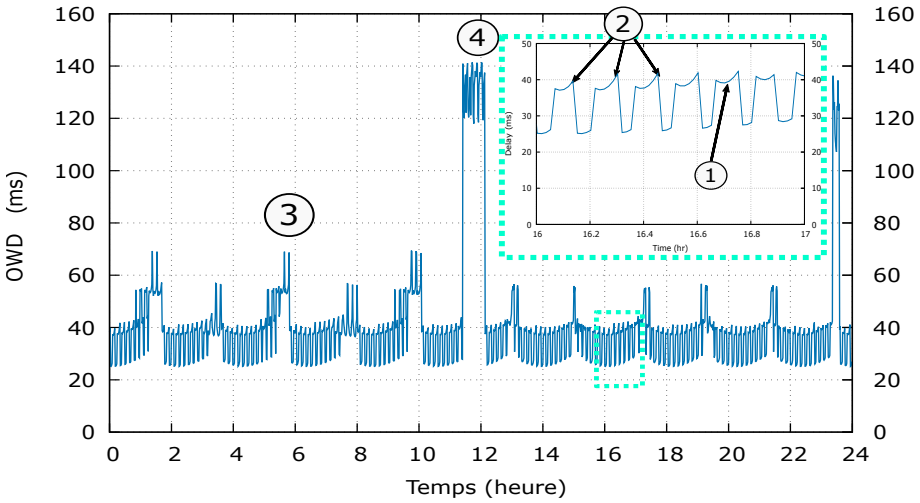


FIGURE 3.1 Évolution de l'One-way Delay (OWD) dans *Iridium NEXT*. Les principaux facteurs de variation de délai sont numérotés comme sur la Figure 2.5

Nous pouvons voir les différentes transitions des sources de variations de délai sur la Figure 2.5b. Sur la Figure 3.1, nous pouvons voir le profil du délai de propagation des paquets envoyés par une source à débit binaire constant — Constant Bit Rate (CBR). La source est un terminal sur un bateau dans l'océan Atlantique et la destination est géographiquement à Londres. Ils sont suffisamment proches sur la longitude pour que l'on puisse examiner en détail le profil de la constellation de satellites *Iridium NEXT* sur 24 heures, en se fondant sur le délai comme métrique pour le calcul de la route. Nous pouvons clairement voir l'effet des variations de délai dans la constellation. L'élévation, les *handovers* intra- et inter-orbitaux et les variations de délai dues à la couture, respectivement notées par 1, 2, 3 et 4 sont représentés sur la Figure 3.1. Nous pouvons constater que les valeurs observées confirment ce qui a été dit précédemment. Certains facteurs de variation sont fréquents mais de faible amplitude, tandis que d'autres sont moins fréquents mais d'amplitude plus importante. Par conséquent, TCP devrait traiter différemment chacune de ces sources de variation de délai. Cependant, cette figure ne peut pas montrer exactement les changements d'ISLs qui sont dus au fait que les satellites se croisent aux pôles. Cette variation de délai est le résultat de la modification du nombre de sauts due aux changements de routage. Nous pouvons voir qu'avec la topologie d'*Iridium NEXT* ; le nombre d'occurrences des événements 3 avec 4 est de 12, ce qui correspond à une constellation avec 6 plans orbitaux. Le nombre d'occurrences de l'événement 2 est de 11, ce qui correspond au nombre de satellites par plan orbital.

3.2.2 Algorithmes TCP

Pour ce qui est des algorithmes TCP, nos choix ont porté sur CUBIC et BBR. Dans l'état de l'art, nous avons parcouru les différents algorithmes de contrôle de congestion de TCP en distinguant ceux qui sont dédiés principalement aux réseaux terrestres et donc couramment utilisés de ceux qui sont conçus spécialement pour les communications par satellite mais qui pour la plupart ne sont que rarement déployés. La taxonomie des différents algorithmes de contrôle de congestion TCP a bien montré l'intérêt de se focaliser sur CUBIC (algorithme du type *loss-based*) et BBR (algorithme du type hybride).

Nous avons jugé crucial d'inclure TCP SACK. En effet, l'environnement satellite à orbite basse est sujet à plusieurs pertes pouvant se produire dans une même fenêtre. Le Bit Error Rate (BER) en GEO est de 1.25×10^{-7} [137]. En LEO, le BER est à 10^{-11} [138]. Selon les recommandations de l'International Telecommunication Union (ITU) les valeurs de BER pour la voix ne devraient pas dépasser 10^{-3} et 10^{-6} pour les services data, en 2016 [139]. Du côté récepteur nous avons retenu du TCP SACK.

Au vu de la dynamique des constellations de satellites considérées et ce qu'elles introduisent comme variation de délai, il est primordial de voir si les problèmes généraux de TCP sont accentués, en plus des différentes spécificités des variantes de TCP.

3.2.3 Outils d'évaluation retenus

Les simulateurs et les émulateurs sont des outils permettant d'appuyer la recherche notamment dans le domaine des réseaux et en particulier de constellations de satellites.

3.2.3.1 Pourquoi utiliser un environnement de simulation ?

Les outils de simulation offrent souvent un environnement facilement contrôlé, une configurabilité, une répétabilité et une évolutivité, mais la précision des mises en œuvre et des résultats de simulation peut être remise en question. Par définition, un simulateur met en œuvre des modèles de protocoles ou d'applications réels qui imitent la réalité. Les modèles de simulation doivent trouver un équilibre entre précision et complexité. La précision nécessaire dépend du cas d'utilisation : dans certains cas, le modèle de simulation peut être mis en œuvre conformément aux spécifications, mais il peut aussi être très simplifié en essayant seulement de capturer les effets d'une mise en œuvre réelle. Un niveau de précision plus élevé nécessite généralement des modèles de simulation plus complexes, ce qui augmente le temps d'exécution de la simulation.

3.2.3.2 Pourquoi utiliser un environnement d'émulation ?

Les outils d'émulation se situent entre les outils de simulation et les bancs d'essai réels. Les émulations peuvent offrir la grande précision des bancs d'essai, tout en fournissant un environnement plus contrôlé comme c'est le cas pour les simulateurs. Les implantations d'émulation imitent les systèmes réels (protocoles, applications) de telle sorte qu'elles peuvent remplacer les implantations logicielles ou matérielles réelles. Ainsi, le niveau de précision des implantations d'émulation est

(généralement) plus proche de la réalité si on le compare aux modèles de simulation. Cependant, les simulations offrent toujours les avantages de scénarios plus grands et plus faciles à construire, de la collecte facile de statistiques à l'échelle du système et de la répétabilité.

3.2.3.3 Outils de simulation et émulation

Les outils de simulation et émulation de satellites appartenant aux vendeurs, aux opérateurs et aux organismes de recherche sont souvent propriétaires, adaptés à des fins particulières et/ou fondés sur des licences commerciales.

3.2.3.3.1 Outils de simulation

L'Open Source Satellite Simulator (OS3) [140] (2013) est un simulateur de satellite reposant sur Objective Modular Network Testbed in C++ (OMNET++), complétant la *framework* INET. Il se concentre sur la mobilité des satellites, les constellations de satellites et l'inclusion de données météorologiques et de modèles de canaux, mais pas sur les protocoles de communication [141].

Network Simulator 3 (NS-3) [142] est un simulateur de réseau à événements discrets open source, principalement destiné à la recherche et à l'enseignement. NS-3 est sous licence GNU GPLv2, et il est disponible pour la recherche et le développement [143]. Satellite Network Simulator 3 (SNS3) [144] (2015) repose sur la plateforme NS-3, qui est un simulateur de réseau à source ouverte modulaire, largement utilisé, évolutif et rapide au niveau des paquets et des systèmes pour la R&D en matière de réseaux. SNS3 modélise des réseaux satellite interactifs à faisceaux multiples avec une charge utile géostationnaire en étoile transparente [141]. Cependant, à notre connaissance au début de la thèse l'outil n'intègre pas les constellations de satellites LEO/MEO et il faudra, par exemple, intégrer les profils de délais en tant que fichiers de traces extraites de simulations sous Network Simulator 2 (NS-2) pour modéliser ces types de constellations. Un modèle Digital Video Broadcasting - Return Channel via Satellite (DVB-RCS) [141] a été développé à partir de NS-2. Cependant, il lui manque certaines fonctionnalités essentielles et il ne fournit pas de modèles DVB-RCS2 ou Digital Video Broadcasting - Satellite (DVB-S)2. En outre, il est fondé sur NS-2, qui présente des problèmes d'évolutivité, de modularité, de maintenance [141].

En tant que simulateur à événements discrets, NS-2, (2000-2010) destiné à la recherche sur les réseaux, prend en charge la simulation des protocoles TCP, du routage et du multicast sur tous les réseaux (câblés et sans fil) [143]. Le fait qu'une panoplie de constellation de satellites LEO/MEO ait déjà été implantée sur cet outil de simulation avec les différentes possibilités de schémas d'ISLs, constitue un atout majeur.

Outre les solutions *open source* mentionnées ci-dessus, il existe plusieurs solutions propriétaires. Par exemple, le DLR (Centre Aérospatial Allemand) dispose d'un simulateur de satellite fondé sur OMNET++, appelé SatSim (2015), et Thales Alenia Space et le CNES ont un simulateur de satellite reposant sur NS-2 appelé System and MAC Satellite Simulator (SMACSAT) [141].

De nouveaux outils de simulation ont vu le jour depuis le début de la thèse mais pas forcément déployés ni facilement accessibles ni documentés. Par exemple, un simulateur de réseau de petits

satellites à grande échelle (en 2020) à événements discrets et fondé sur l'abstraction du traitement de la couche réseau a été développé [145]. Il a pour objectif de faciliter la mise en place de la plateforme de simulation de satellites de manière à ce qu'elle soit efficace, à faible *overhead* et évolutive. On peut encore citer le simulateur de réseau pour les grands réseaux de satellites LEO (en 2021), un outil qui se veut une fusion entre l'outil de visualisation 3D de constellations de satellites General Mission Analysis Tool (GMAT) de la National Aeronautics and Space Administration (NASA) et une version gros grain de l'outil précédemment mentionné SNS3 [146].

3.2.3.3.2 Outils d'émulation

EmuNET [147] est un émulateur de réseaux en temps réel, développé en 2004, utilisé pour tester les protocoles dans diverses conditions, telles que la limitation du débit, le délai et la gigue du réseau, le taux d'erreur binaire, différents systèmes de mise en file d'attente, etc. Il implante également une émulation de l'environnement satellite. L'inconvénient est que nous n'avons aucune information sur l'évolution du développement de l'outil actuellement. Open Satellite Network Demonstrator (OPENSAND) [148], est un émulateur *user-friendly* DVB-RCS2 et DVB-S2 développé initialement par Thales Alenia Space, le Centre National d'Études Spatiales (CNES) et Viveris Technologies. OPENSAND a été publié sous licence GPLv3 et il est disponible gratuitement. Il a également été promu par le CNES comme outil logiciel open-source de référence dans le cadre de ses activités de recherche et développement dans le domaine des systèmes et des réseaux de communication par satellite et par constellations de satellites. OPENSAND émule un système de communication par satellite DVB-RCS2/DVB-S2. Les communications par constellations de satellites sont modélisées par des fichiers OWD récupérables à partir par exemple de traces de délai comme sorties de simulation via NS-2. La plateforme peut être vue comme une boîte noire configurable fournissant un accès au réseau satellite sur chaque terminal et GW. Les utilisateurs peuvent brancher leurs stations de travail et leurs serveurs derrière les GW et/ou les terminaux et effectuer des tests à travers un réseau satellite. Il fournit des points de mesure et des outils d'analyse pour l'évaluation des performances et permet à la recherche et l'ingénierie de valider différentes fonctionnalités.

En ce qui concerne les émulateurs propriétaires, on peut citer Advanced IP Network Emulator (AINE) [149], un émulateur développé par Indra en 2008 pour l'émulation et la caractérisation en temps réel des performances et des protocoles IP et de la couche de liaison, particulièrement pour des réseaux par satellite. L'émulateur prend en compte le comportement des réseaux DVB-RCS y compris les canaux régénératifs, l'allocation des ressources et les schémas Differentiated Services (DIFFSERV), QOS, DVB-S2 et les mécanismes d'adaptation *cross-layer*. Nous n'avons pas d'informations sur la possibilité d'intégrer les constellations de satellites et sur le fait qu'il soit toujours maintenu. Un émulateur DVB-RCS2 [141], [150] a également été développé dans le cadre du projet ESA "DVB-RCS Next Generation Support and Verification Testbed" par le DLR et VeriSat en 2011. L'outil permet de reproduire la plupart des fonctionnalités d'un système satellite réel de la couche physique à la couche transport. Il est capable de prendre en charge un grand nombre de services, IPv4 et IPv6 en mode de transmission unicast et multicast sont possibles. Cependant, l'outil est limité à 3 terminaux

satellites mais pourrait être étendu. Nous n'avons pas davantage d'informations sur l'évolution et la maintenance de l'outil. NE-ONE [151] l'émulateur de réseaux de iTrinegy développé en 2017 est payant. Il permet d'intégrer les réseaux de constellations de satellites. Sur le site du produit, on constate que l'émulateur est plus tourné vers des applications commerciales que vers la recherche. SCNE (Satellite Constellation Network Emulator) [152], [153] de l'ESA développé en Septembre 2018 se fonde sur une co-simulation entre un simulateur orbital et un autre réseau le tout commandé par un orchestrateur. L'outil offre la possibilité de fonctionner en mode simulation ou émulation. C'est un émulateur de réseau satellite en temps réel pour les futures constellations. Il vise l'évaluation des performances de bout-en-bout de solutions de routage, des protocoles de transport pour du *broadband* et l'accès au Web par satellite, les caractéristiques des liens inter-satellite, et les fonctionnalités de la charge utile de télécommunications par satellite au niveau de la couche 2 et des couches supérieures.

De nouveaux outils d'émulation tels que le Satellite Communication Digital Twin [154] (2021) commencent à voir le jour. Différentes technologies existantes sont combinées pour émuler la dynamique des liens, la génération automatique de trafic et l'émulation globale de la topologie satellite.

3.2.3.4 Outils de simulation et émulation retenus

Pour l'étude en simulation, nous avons retenu NS-2. En premier lieu en effet, la topologie de la constellation de satellites retenue, *Iridium NEXT*, figure parmi les modèles intégrés. D'autre part, l'outil a atteint une maturité technologique permettant une stabilité et la portabilité du travail vers d'autres outils. Notamment les fichiers de trace en sortie peuvent être utilisés en entrée d'autres outils. Les autres outils sont le plus souvent trop spécifiques, mal documentés et la qualité des modèles implantés n'a pas été autant testée que pour NS-2.

Pour l'émulation, nous avons choisi OPENSAND qui nous permet d'avoir une modélisation assez réaliste d'une topologie et d'un environnement satellite. De plus, ce choix nous permet une certaine continuité et complémentarité avec le choix de simulateur (NS-2). En particulier, nous pouvons exploiter les fichiers de résultats du simulateur pour les déployer dans l'émulateur. De plus, cet outil est à portée de main avec une évolution en cours (début de thèse v5.1 puis passage à la version v5.2 et actuellement la v6.0 est en cours de production). Les autres outils d'émulation sont peu reconnus, peu maintenus et régulièrement peu accessibles.

3.3 Paramètres et caractéristiques génériques

Nous allons énoncer les différents paramètres génériques et les métriques qui seront étudiés dans toute la thèse, sauf mention contraire.

- ***Tailles des fichiers envoyés considérées :***
  - 9kB : ce court fichier tiendra dans une IW de 10 paquets . La taille d'un paquet est égale à 1040B sous NS-2 ;
  - 15MB : ce fichier plus grand permet à TCP de sortir du *Slow Start* ;
  - Fichier illimité (*unlimited-bulk*) : il nous permet d'évaluer le comportement de TCP dans la phase d'évitement de congestion.

Ces tailles de fichiers ont été choisies afin d'analyser le comportement de TCP dans différentes phases pour mieux comprendre l'impact des variations de délai.

- ***Métriques observées :***
  Nous nous concentrerons sur trois métriques sensibles au délai, à savoir la durée de transfert d'un fichier, le débit instantané reçu ou encore la CWND :
  * Durée de transfert d'un fichier (taille de fichier limitée) : du point de vue de la Quality of Experience (QoE), il s'agit de la mesure pour laquelle la variation de délai dans la constellation de satellites a un impact quantifié du point de vue de l'utilisateur ;
  * Débit instantané reçu du point de vue de l'application (taille de fichier illimitée) : il donne une idée plus détaillée et plus fine du comportement de TCP ;
  * CWND : cette variable d'état maintenue par l'émetteur permet de limiter la quantité de données que TCP peut envoyer avant de recevoir un ACK du point de vue du contrôle de congestion. C'est un moyen d'empêcher le lien entre l'émetteur et le récepteur d'être surchargé par un trafic trop important, d'où la pertinence d'observer cette métrique pour suivre l'état du lien.
- ***Débits :*** Nous voulions utiliser une constellation type avec ISLs avec des débits tels que offerts par *Iridium NEXT* [155] :
  * Débits Up/Down-links : 1.5 Mbit/s
  * Débits ISLs : 25 Mbit/s
- ***Capacité de la file d'attente du goulot d'étranglement :*** Elle est égale au BDP du lien. Ce choix est justifié par le fait que BBRv1 maintient un *inflight* cap à $2 \times$ BDP ; une file se crée de l'ordre d'un BDP d'où le choix d'un BDP.
- ***Métrique de routage :*** pour le routage dans la constellation de satellites nous avons choisi la métrique *hop-count*, au lieu du routage *delay-based*. Cela se traduit surtout par une consommation plus faible en puissance de calcul donc par une mise à jour et une synchronisation de l'état de la constellation de satellites pour tous les satellites plus simples et moins fréquentes. Une solution *delay-based* est extrêmement coûteuse car contrairement aux réseaux filaires ou terrestres, il y a de nombreuses fluctuations sur plusieurs échelles de temps engendrant des calculs très nombreux.
- ***Modèle :*** sans erreurs. Comme nous nous intéressons plutôt à l'impact de la variation de délai et à une sélection d'algorithmes, nous n'avons jugé pas pertinent d'introduire des erreurs vu que cela peut altérer les résultats qui nous intéressent et semble cohérent avec un BER de 10^{-11}.
- ***Choix de la* GW *Iridium NEXT :*** L'évaluation de la performance lorsque l'emplacement des terminaux et des GWs varie peut rendre l'analyse assez complexe. Comme plusieurs GWs sont disponibles, nous avons dû évaluer la pertinence du choix d'une GW satellite pour la suite de la thèse.

Nous avons effectué 50000 simulations. Dans chaque simulation, le terminal est placé à une position aléatoire et télécharge à un instant aléatoire un fichier de 9kB ou un fichier de 15MB depuis l'une des 7 GWs *Iridium NEXT*.

La Figure 3.2, représente la durée de transfert pour chacun des fichiers de 9kB et de 15MB, et ce pour chaque GW *Iridium NEXT*. Les résultats montrent qu'à l'exception des valeurs maximales qui présentent de légères différences, la distribution empirique de la durée de transfert est tout à fait semblable, quelle que soit la position de la GW.

Nous avons conclu des résultats présentés que le choix de la GW *Iridium NEXT* n'a pas beaucoup d'importance. Pour la suite de la thèse, nous allons donc considérer la GW d'Hawaï d'*Iridium NEXT*.

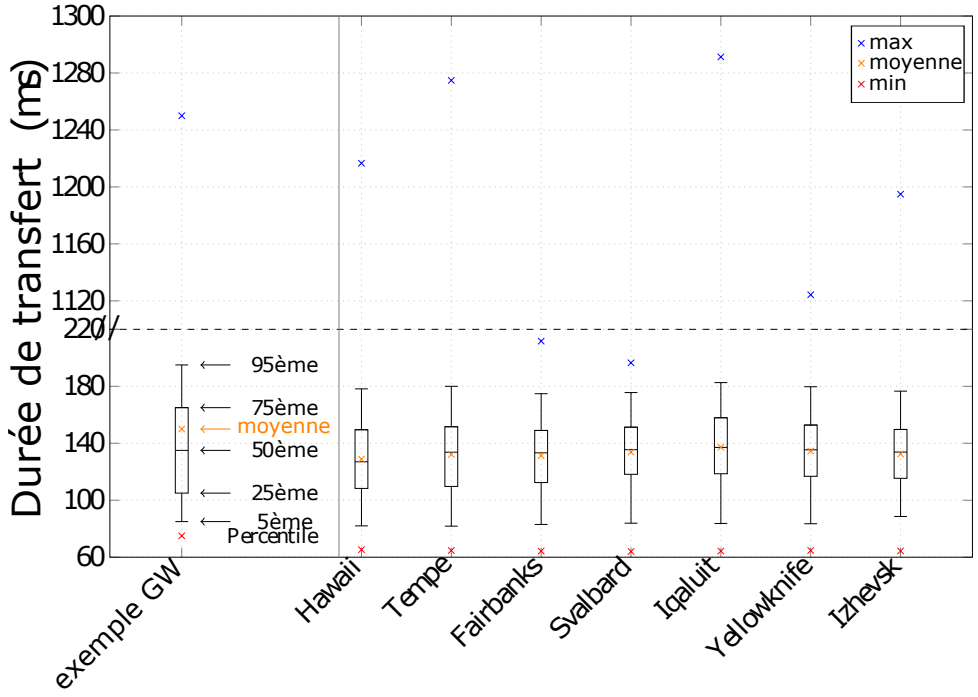
3.4 Conclusion

Dans ce chapitre, nous avons défini en premier lieu notre environnement système, la constellation retenue *Iridium NEXT*, ses spécificités, et ce que cela peut engendrer en termes de variations de délai. En second lieu, nous avons rappelé les algorithmes TCP, CUBIC et BBR, auxquels nous allons nous intéresser et les différentes options. Ensuite, nous avons abordé les outils d'évaluation par simulation (NS-2) et émulation (OPENSAND) dont nous allons nous servir pour la suite, ainsi que les paramètres génériques à retenir.

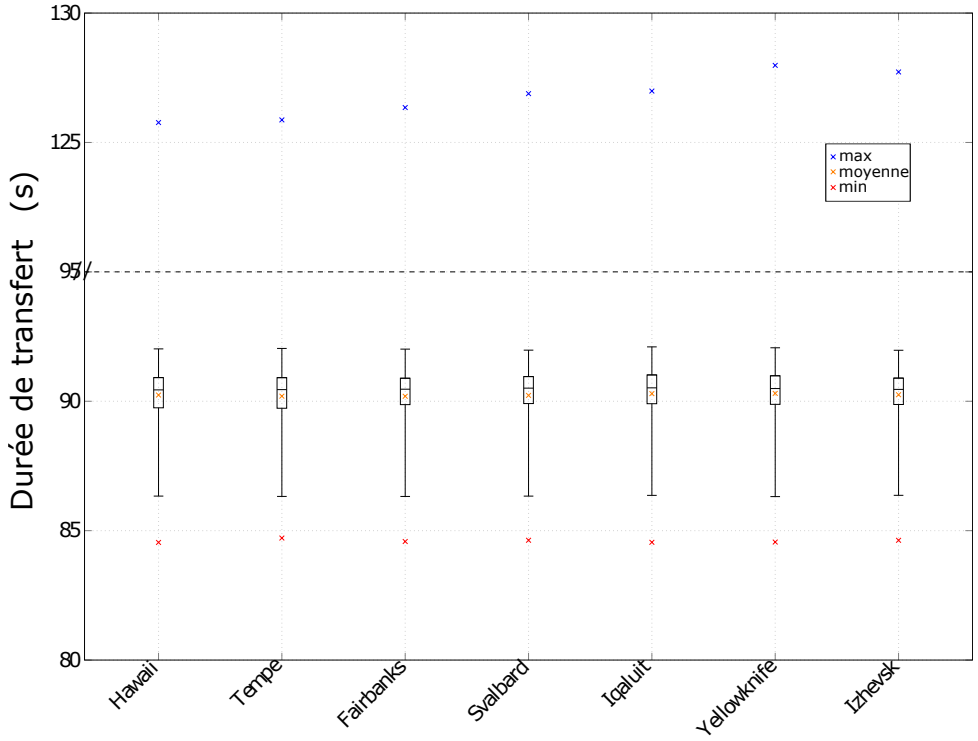
Nous avons regroupé les paramètres génériques, des scénarios présagés établis précédemment dans ce chapitre, dans le Tableau 3.1.

TABLE 3.1 Paramètres communs aux différents scénarios

Paramètres génériques						
Fichiers envoyés	**Métrique observée**	**Débits**	**Capacité de la file**	**Métrique de routage**	**Modèle**	**Choix de la GW**
-9kB -15MB -Fichier illimité (*unlimited-bulk*)	-Durée de transfert d'un fichier -Débit instantané reçu -CWND	-Up-Down-links : 1.5 Mbit/s -ISLs : 25 Mbit/s	BDP	*hop-count*	Sans erreurs	Hawaï



(a) Fichier 9kB



(b) Fichier 15MB

FIGURE 3.2 Durée de transfert pour des instants et des positions de terminaux aléatoires

CHAPITRE 4

SIMULATION DE L'IMPACT DE LA VARIATION DE DÉLAI SUR CUBIC

4.1 Objectif

Dans ce chapitre, nous allons nous intéresser à l'impact des différentes sources de variations de délai de la constellation de satellites de référence *Iridium NEXT* sur CUBIC. Pour cela, nous allons bâtir une étude de cas au travers d'un scénario de téléchargement/d'envoi de fichiers.

4.2 Outil de simulation et contexte de l'expérimentation

Ici l'objectif est d'étudier l'impact des différentes sources de variation de délai sur CUBIC durant le téléchargement/l'envoi de fichiers à 9kB et 15MB. Nous l'avons évalué en NS-2.

La variation de délai agit directement sur le transfert de fichiers et donc cela nous permettra de le percevoir concrètement. Les métriques permettant d'observer cet impact sont la durée de transfert d'un fichier, le débit "instantané" avec une granularité de 25s pour voir en gros grain ce qui se passe, la CWND et le numéro de séquence.

Nous définissons deux ensembles de débits (bas débit et haut débit, avec une différence d'un facteur 80). En général, la justification de ce choix est que nous voulons déterminer l'impact d'une constellation type avec ISLs sur TCP et ne pas limiter uniquement nos conclusions au débit offert par *Iridium NEXT*. Le bas débit correspond à des up- et down-links de 1.5 Mbit/s, comme pour Iridium Certus [155] et 25 Mbit/s pour les ISLs, comme précisé précédemment. Pour le haut débit, les up- et down-links sont de 120 Mbit/s et 2 Gbit/s pour les ISLs.

Le Tableau 4.1 regroupe les paramètres des scénarios de simulation. Un exemple illustratif du schéma expérimental peut se trouver sur la Figure 4.1.

Le BDP a été calculé en fonction du RTT que nous avons tiré par simulation entre des positions de villes différentes dispersées sur le globe avec une GW *Iridium NEXT*. Nous avons retenu le RTT le plus important pour que cela puisse inclure tous les cas.

TABLE 4.1 Paramètres des scénarios de simulation

Paramètres	Valeur
Métrique observée	Débit en réception (fenêtre de 25s) et CWND
Durée	[9 :14]h
Capacité file d'attente = BDP (paquets de 1040B)	Bas débit : 167
	Haut débit : 1024
Émetteur/Récepteur	GW Hawaï / Emplacement change
Protocole de transport	Émetteur : CUBIC
	Récepteur : SACK
Débits Up/Down-links	Bas débit : 1.5 Mbit/s
	Haut débit : 120 Mbit/s
Débits ISLs	Bas débit : 25 Mbit/s
	Haut débit : 2 Gbit/s
Nombre de flux	1
Outil	NS-2

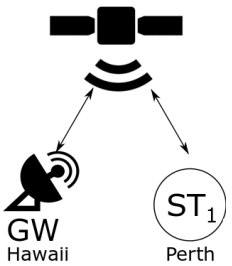


FIGURE 4.1 Schéma expérimental

4.3 Impact de la couture

Cette section est consacrée à l'analyse de l'impact du délai du *handover* de la couture sur une transmission de fichiers entre deux terminaux. Elle considère la configuration à bas débit : les résultats pour la configuration à haut débit présentent la même tendance.

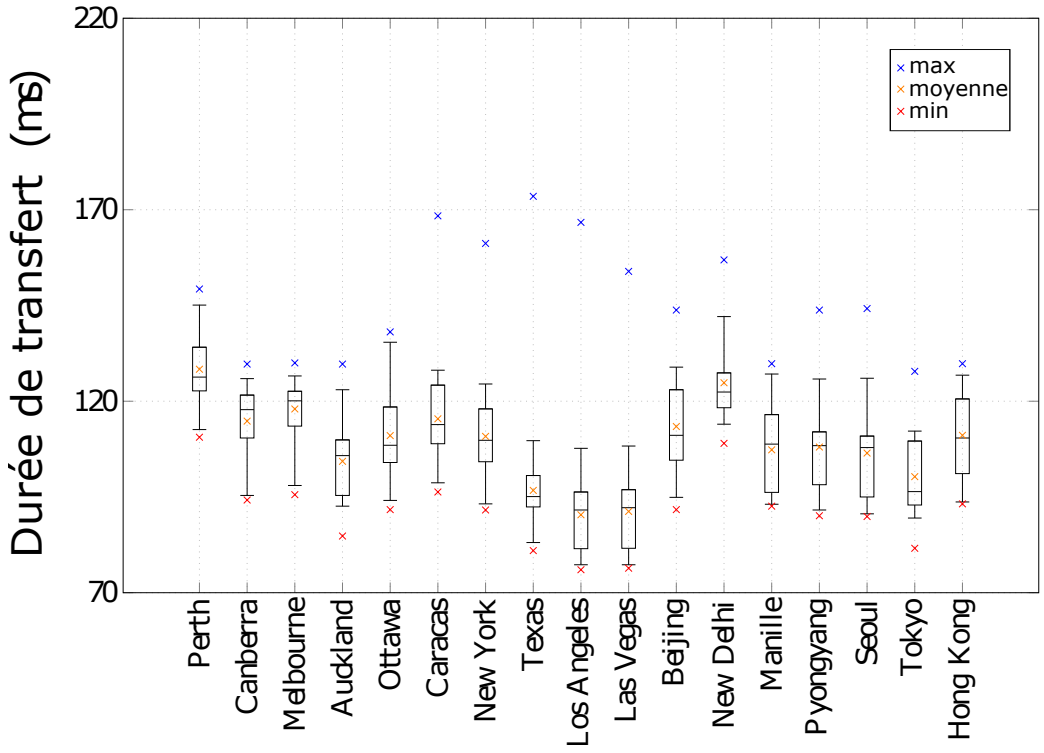
Le délai du *handover* de la couture (numéro 4) entraîne d'importantes variations de délai. Ceci est illustré dans la Figure 2.5. L'un des objectifs de cette section est d'évaluer la mesure dans laquelle le délai du *handover* de la couture est un problème pour différentes tailles de fichiers.

4.3.1 Impact de la couture sur les fichiers de 9kB

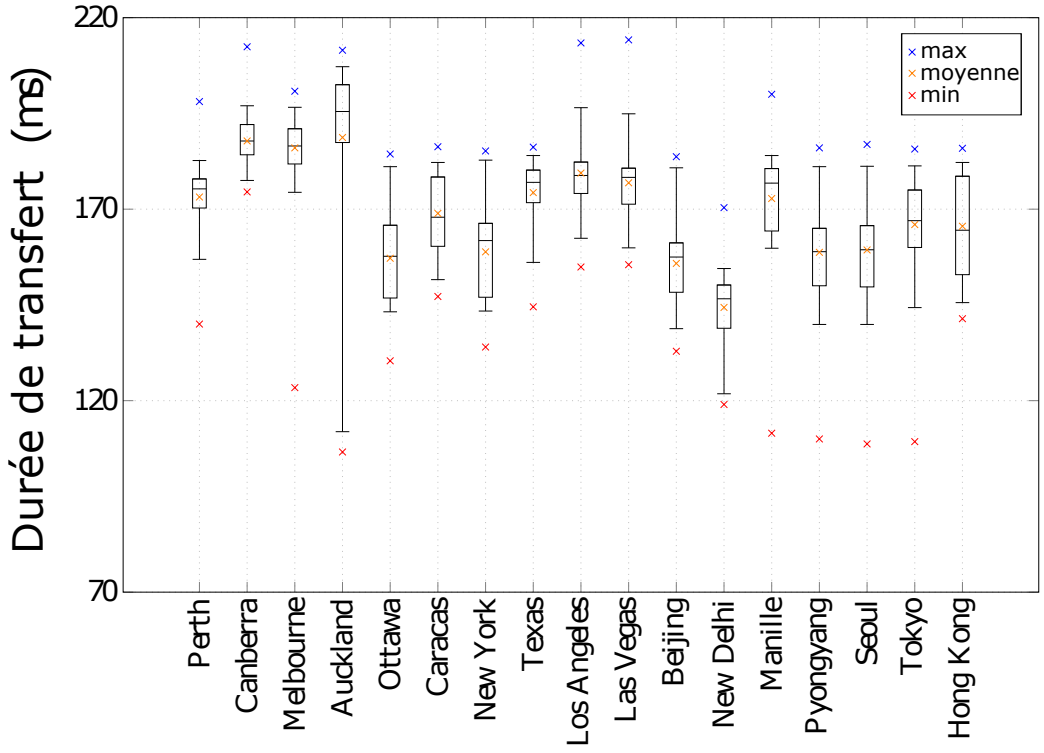
Nous avons effectué plusieurs simulations d'un transfert de fichiers de 9kB entre deux terminaux. L'un est la GW *Iridium NEXT* à Hawaï, qui est considérée comme le serveur. L'autre terminal, le client, est placé dans 17 villes différentes que nous avons choisies dispersées sur le globe. L'instant de début des simulations est tiré aléatoirement dans deux des intervalles : lorsque les terminaux ne sont pas séparés par la couture et lorsqu'ils le sont.

La Figure 4.2 représente la durée nécessaire au transfert de 9kB pour des terminaux situés dans des villes différentes, qu'ils soient séparés par la couture ou non. Nous pouvons clairement voir que

la durée de transfert moyenne lorsque les terminaux sont séparés par la couture est beaucoup plus importante que lorsqu'ils ne sont pas séparés par la couture. Par exemple, prenons le pire et le meilleur cas de figure, c'est-à-dire les villes pour lesquelles la séparation a eu le moins et le plus d'impact sur la durée de transfert moyenne. Pour New Delhi, elle passe de 124ms lorsque les terminaux ne sont pas séparés par la couture à 144ms lorsque les terminaux sont séparés par la couture, soit une augmentation de 15,6 %. Pour Los Angeles, la durée de transfert moyenne passe de 90ms lorsque les terminaux ne sont pas séparés par la couture à 179ms lorsque les terminaux sont séparés par la couture, soit une augmentation de 98 %.



(a) En dehors de la couture, transfert de fichier de 9kB



(b) Séparés par la couture, transfert de fichier de 9kB

FIGURE 4.2 Durée de transfert pour un fichier de 9kB

Afin de mieux comprendre l'impact de la couture sur le transfert de fichiers de 9kB, nous présentons dans la Figure 4.3 différents événements pour un terminal source sur un bateau dans l'océan Atlantique communiquant avec un terminal de réception à Londres. Il y a des ACKs non ordonnés ce qui est dû au fait que les ACKs envoyés plus tôt ont emprunté un chemin plus long que ceux envoyés plus tard. Sur la Figure 4.4, on peut voir que les ACK des paquets n°0 et n°1 ont emprunté un chemin de 1 saut, juste avant que les deux terminaux ne soient séparés par la couture. Lorsque la couture a séparé les terminaux, les ACKs des paquets n°2-3 ont pris un chemin de 11 sauts. Et les ACKs des paquets n°4 à 9 ont pris un chemin de 10 sauts. Ceci a aussi été constaté pour les paquets de données en envoyant dans le sens inverse pour une taille de fichier dépassant les 10 paquets (la taille de la fenêtre IW), c'est-à-dire de Londres vers une destination dans l'océan Atlantique.

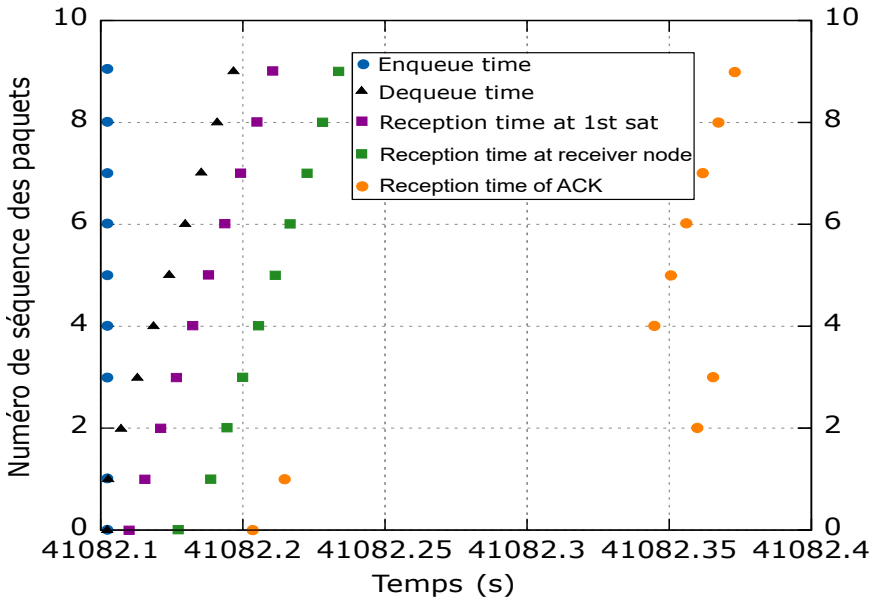


FIGURE 4.3 Transmission des données d'un fichier de 9kB durant le passage par la couture (cas d'utilisation 4 sur la Figure 3.1)

Contrairement aux résultats présentés dans la Figure 3.2, les résultats présentés dans la Figure 4.2 présentent une variation importante. Cela peut s'expliquer par le fait que les terminaux ont des positions aléatoires et que la GW a une position fixe pour la Figure 3.2 alors que les terminaux et la GW ont des positions fixes dans cette sous-section.

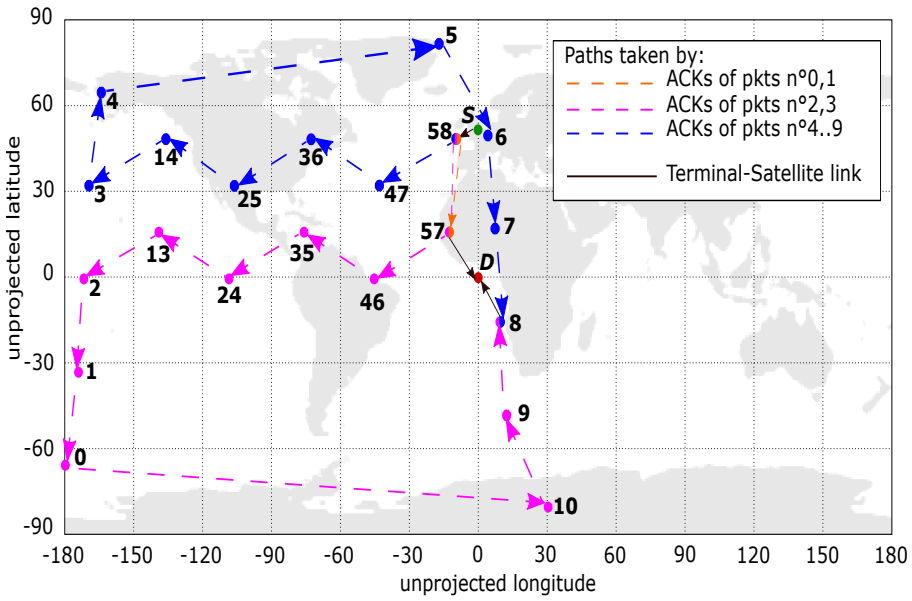


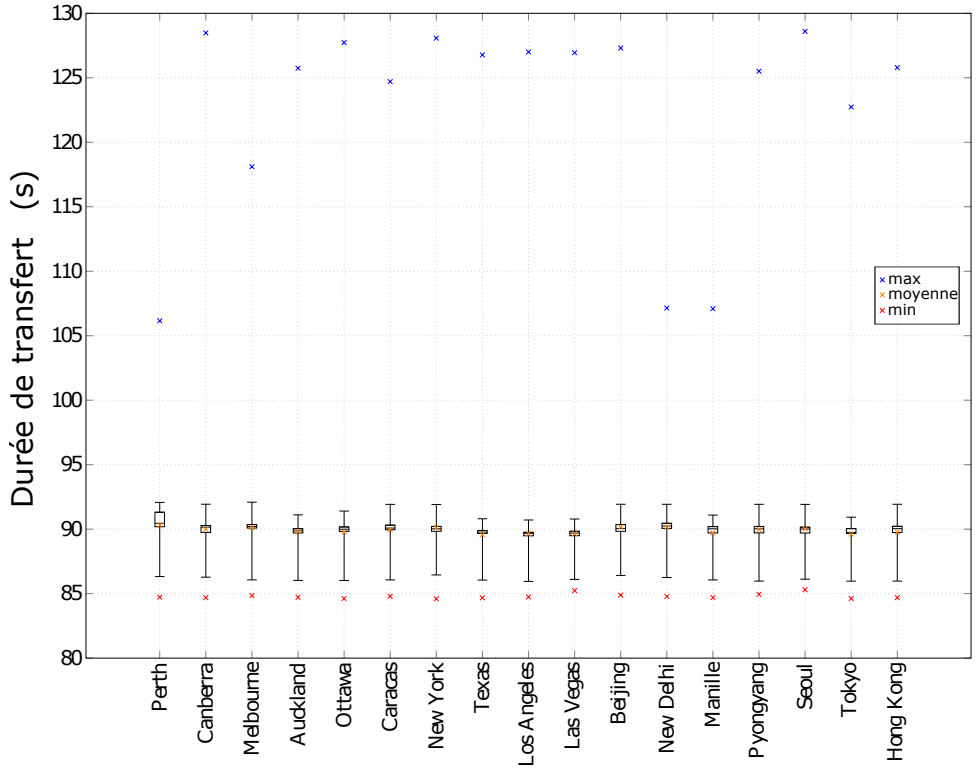
FIGURE 4.4 Chemins empruntés par tous les paquets ACK durant la transmission du fichier de 9kB au passage par la couture (cas d'utilisation 4 sur la Figure 3.1) [57]

Cette sous-section nous a permis de mesurer que, pour une GW satellite donnée et quelle que soit la position du terminal satellite, la couture a un impact important sur les transferts de petits fichiers (de l'ordre de 9kB).

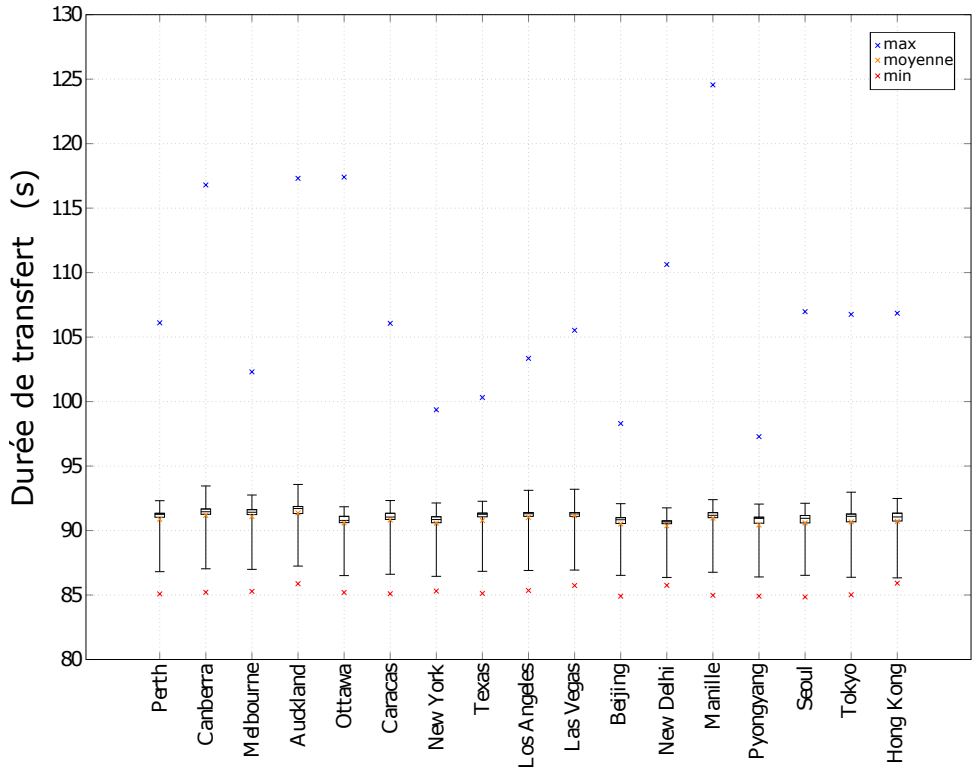
4.3.2 Impact de la couture sur les fichiers de 15MB

Cette sous-section se concentre sur l'impact du *handover* de la couture sur la transmission de fichiers de 15MB. Nous avons effectué plusieurs simulations d'un transfert de fichiers de 15MB, entre deux terminaux : l'un est la GW *Iridium NEXT* à Hawaï, qui est considérée comme serveur, et l'autre terminal, le client, est placé dans 17 villes différentes. L'instant de début des simulations est choisi aléatoirement séparés ou non par la couture.

La Figure 4.5 représente la durée nécessaire pour transférer un fichier de 15MB pour des terminaux situés dans des villes différentes, qu'ils soient séparés par la couture ou non. Il ne semble pas y avoir de différence flagrante entre les valeurs des durées de transfert.



(a) En dehors de la couture, transfert de fichier de 15MB



(b) Séparés par la couture, , transfert de fichier de 15MB

FIGURE 4.5 Durée de transfert pour un fichier de 15MB

4.3.3 Impact de la couture sur les différentes tailles de fichiers

La section 4.3.1 a montré que le délai dû au *handover* de la couture augmente la durée de transmission d'un fichier de 9kB d'une manière non négligeable. Cependant, la section 4.3.2 a illustré que cet impact peut être négligé lorsque le fichier est de l'ordre de 15MB. Cette sous-section vise à déterminer la mesure dans laquelle le délai dû au *handover* de la couture a un impact sur le transfert d'un fichier, en fonction de sa taille.

Le Tableau 4.2 rassemble le temps de transfert moyen — Mean Transfer Time (MTT) pour un transfert de fichier entre un terminal situé à Los Angeles et un autre situé à Hawaï. Le $MTT_{no\ seam}$ signifie que le transfert de fichiers entre les terminaux n'est pas affecté par le délai dû au *handover* de la couture alors que c'est le cas pour le MTT_{seam}. Ce tableau confirme les conclusions qui ont été proposées au début de cette sous-section.

TABLE 4.2 MTT des fichiers

Paramètre \ Taille du fichier	fichier de 9kB	fichier de 15MB
$MTT_{no\ seam}$	90.35ms	89.67s
MTT_{seam}	179.43ms	91.02s
$MTT_{no\ seam}/MTT_{seam}$ (%)	50.35	98.52

Nous avons effectué plusieurs simulations avec des fichiers de tailles différentes envoyés d'un serveur situé près de la GW *Iridium NEXT* à Hawaï vers un client terminal placé dans 17 villes différentes. Le Tableau 4.3 présente les valeurs moyennes en pourcentage du rapport entre le $MTT_{no\ seam}$ et le MTT_{seam}.

TABLE 4.3 Impact de la couture en fonction de la taille des fichiers

Paramètre \ Tailles du fichier (kB)	25	50	100	500	1000	5000	10000
mean($MTT_{no\ seam}/MTT_{seam}$)(%)	59.38	59.35	69.79	90.71	96.78	99.97	98.55

En considérant les valeurs du Tableau 4.3, nous pouvons conclure qu'à partir de 100kB la couture a beaucoup moins d'effet sur le transfert de fichiers.

4.4 Impact de la variation de délai due à différentes sources autres que la couture

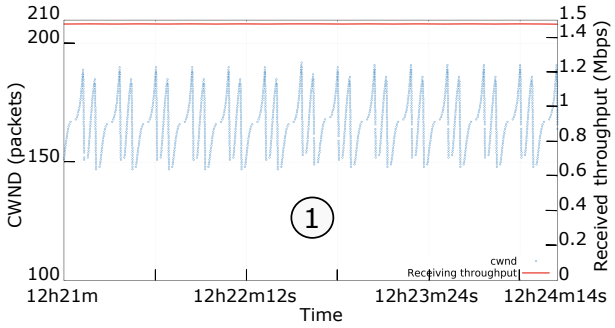
Dans cette section, nous allons illustrer l'impact des différentes sources de variations de délai d'*Iridium NEXT* autres que la couture sur une connexion TCP. Nous présenterons également les résultats pour les configurations à bas et haut débit afin d'illustrer l'impact des sources de variation de délai.

Nous avons vu que la variation de délai dans les constellations LEO fait intervenir quatre phénomènes inégalement fréquents. Après avoir étudié en détail l'effet de la couture, nous voulons

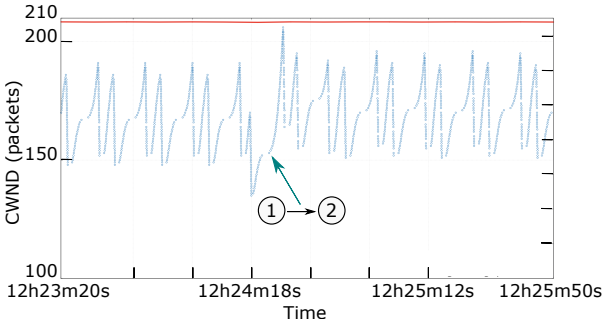
maintenant voir de près l'effet du comportement pendulaire des trois autres causes de variation de délai sur TCP que nous avons pu simuler : la variation d'élévation notée 1, le délai du *handover* intra-orbital noté 2 et le délai du *handover* inter-orbital noté 3 sur la Figure 3.1.

Nous avons effectué une simulation pour une source CUBIC qui émet en continu (*unlimited-bulk*) le terminal est sur un bateau dans l'océan Atlantique et le terminal de réception à Londres. Les résultats présentés ici correspondent à la phase de stabilité, c'est-à-dire après le *slow start* pendant 52 mn19s à partir de 12h21.

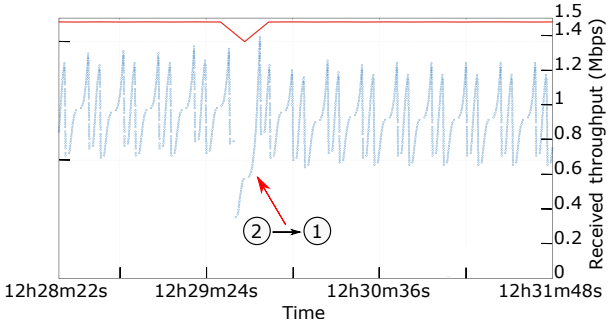
4.4.0.1 Résultats pour la configuration à bas débit



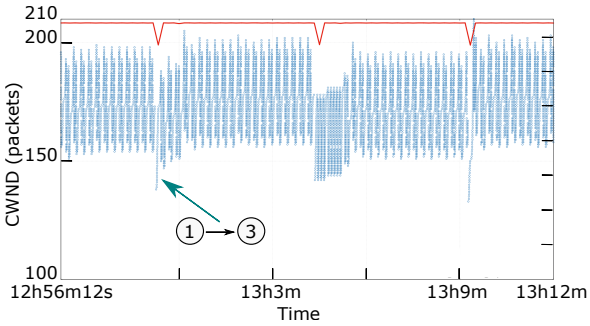
(a) Délai dû à la variation d'élévation



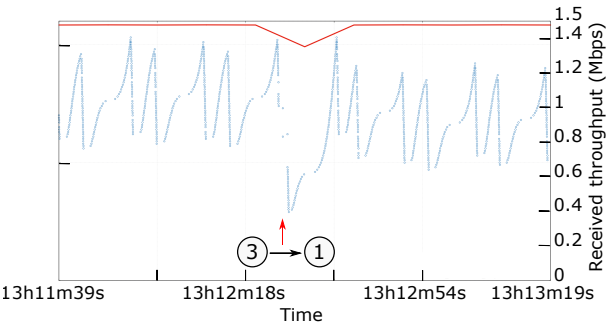
(b) Délai dû au *handover* de l'élévation à l'intra-orbital



(c) Délai dû au *handover* de l'intra-orbital à l'élévation



(d) Délai dû au *handover* de l'élévation à l'inter-orbital



(e) Délai dû au *handover* de l'inter-orbital à l'élévation

FIGURE 4.6 Impact de la variation de délai due à différentes sources autres que la couture sur la CWND et le débit en réception pour une configuration à bas débit

Sur la Figure 4.6, nous avons rassemblé les 3 différents cas d'utilisation de la variation de délai énoncés précédemment.

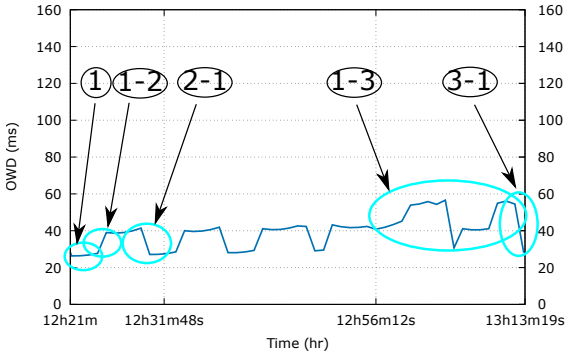


FIGURE 4.7 Zoom sur les transitions de l'OWD de la Figure 3.1

Dans la sous-figure 4.6a, nous pouvons voir que le délai de variation d'élévation n'affecte pas les mesures de performance de TCP *i.e.* la CWND et le débit instantané de réception, car le délai varie doucement dans [2.6ms ; 8.2ms] pour le même et seul lien satellite. Comme le montre la Figure 4.7. Comme nous pouvons le voir, les débuts de simulation sont tronqués et le *slow start* n'est pas visible. Nous pouvons voir une augmentation suivant la partie concave de la fonction cubique, puis le premier plateau qui correspond à W_{max} aux alentours de 167 paquets. Par la suite, la CWND augmente selon la partie convexe jusqu'à atteindre une valeur (un max absolu) vers 191 paquets où des pertes ont été détectées par l'émetteur.

Il est important de noter que, logiquement, la CWND se décortique de cette façon : CWND = BDP_*lien* + *reste_à_bufferiser* et dans le cas où *reste_à_bufferiser* est supérieur à la *capacité_buffer* les paquets seront de toute évidence rejetés, ce qui naturellement engendre des pertes.

Dans une configuration à bas débit, la *capacité_buffer* est à 167 comme expliqué dans 4.2, et le BDP correspondant (pour un OWD = 49ms) vaut sensiblement 17 paquets (pour des paquets de 1040B sous la configuration par défaut de NS-2). Le BDP reste quasi constant étant donné que l'OWD varie au plus à 8.2ms (environ 3 paquets). À partir d'une CWND = 17 + 167 = 184 paquets, des paquets sont rejetés (ce qui correspond au deuxième pic le plus important, qui se répète d'une manière périodique).

De ce fait, après la détection de pertes, CUBIC diminue sa CWND à $(1-\beta)$CWND.La nouvelle valeur de W_{max} prend la taille de la CWND qui est de 191 paquets.

Par la suite, la CWND reprend en *fast recovery* et CA sa croissance avec la partie convexe de la fonction cubique dans le but d'atteindre un plateau à la nouvelle valeur de W_{max} (= 191 paquets).

La perte a été détectée à environ 184 paquets donc la nouvelle valeur de $W_{max} = 184$ et $W\_last\_max = 191$. Du fait de l'application de la *fast convergence*, il y a une comparaison entre W_{max} et W_last_max. La condition $W_{max} < W\_last\_max$ est vérifiée, donc $W_{max} \leftarrow W_{max}(2-\beta)/2$ sont environ 166 paquets. CWND diminue et reprend de $(1-\beta)$ à environ 147 paquets.

Ainsi, la fonction cubique reprend encore une fois avec de la *fast recovery* et CA pour un plateau à la nouvelle valeur de $W_{max} = 166$ paquets. La perte a été détectée à 191 paquets, $W_{max} = 191 > W\_last\_max = 166$ donc $W\_last\_max \leftarrow W_{max}$ et CWND prend $(1-\beta)$CWND ≈ 153. Et ainsi de

suite.

Nous remarquons que cela n'a pas d'impact sur le débit en réception. Cela est probablement dû au fait que la CWND a été toujours supérieure au BDP qui est resté quasi-constant tout le long de la transmission. La bande est donc correctement utilisée.

Si nous nous concentrons sur le délai dû aux *handovers* intra-orbitaux (des cas d'utilisation 1 à 2 dans la sous-figure 4.6b), il n'y a pas beaucoup d'impact sur le débit même si cela a un impact sur l'évolution de la CWND. Avant la transition, nous maintenons la même interprétation que pour la Figure 4.6a étant donné que l'OWD varie peu autour de 26ms comme nous pouvons le voir sur la Figure 4.7. Le BDP vaut donc 9 paquets de 1040B. Par conséquent, au-delà du *buffer* à 167 paquets, la CWND pourra aller jusqu'à 176 paquets sans perte. Nous observons que la CWND dépasse cette limite où les pics restent inchangés à, respectivement, 191 et 184 paquets. Les pertes n'ont pas eu d'impact sur le débit en réception, même avec une diminution de la CWND au-delà de 150 paquets. Tout de même, le lien reste bien utilisé étant donné que le BDP devrait être à 9 paquets. À la transition, l'OWD passe de 26ms à 41ms, comme le montre la Figure 4.7. Le BDP passe donc de 9 à 14 paquets ce qui fait une différence de 5 paquets. Cela explique le fait que la CWND augmente de plus belle, puis les pics descendent aux alentours de 196 et 189 paquets ce qui correspond à la différence de BDP.

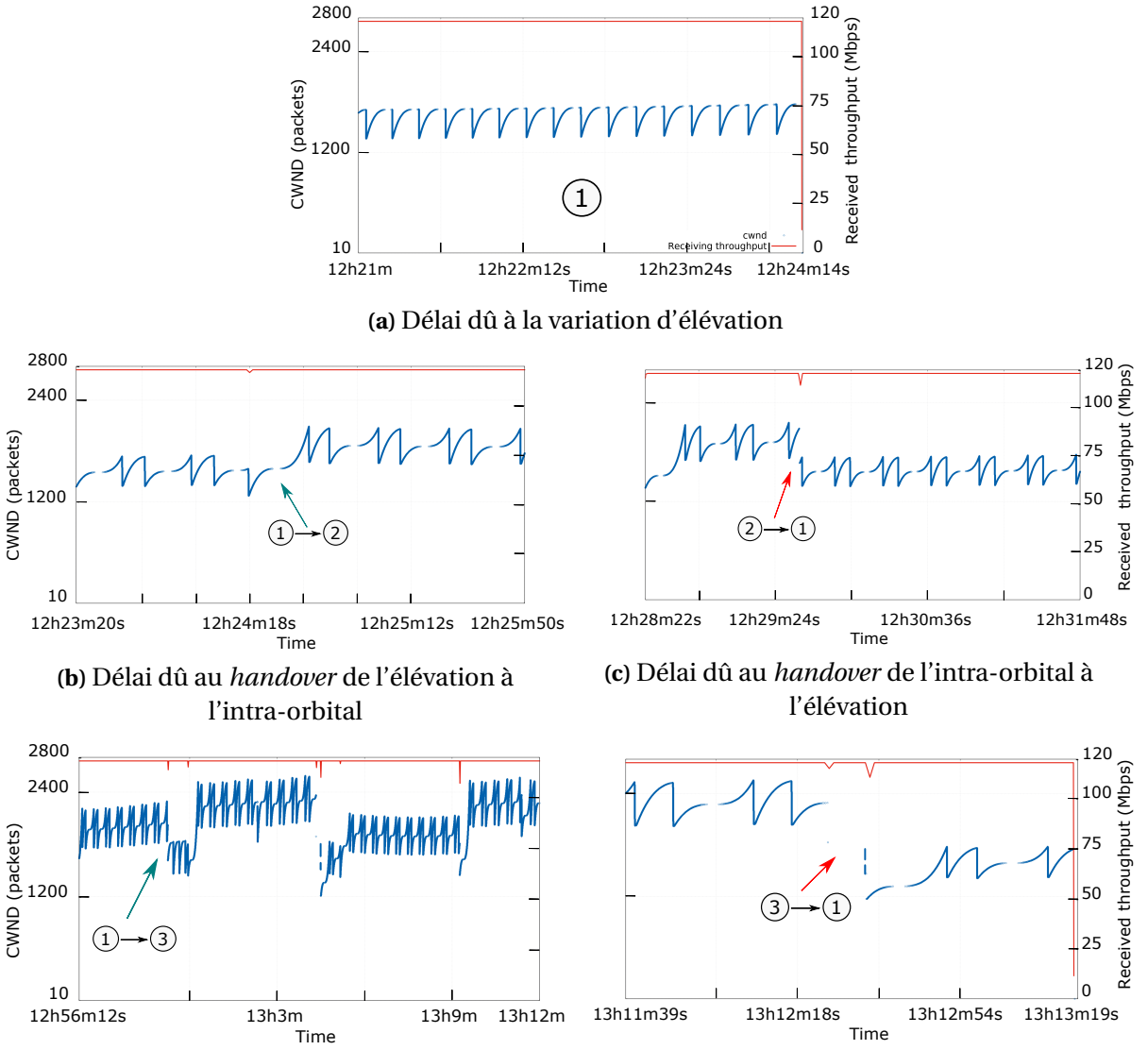
Cependant, dans la transition inverse (du 2 au 1 dans la sous-figure 4.6c), la transition quasi-inverse se produit, l'OWD passe de 41ms à 28ms. Comme le montre la Figure 4.7, le BDP passe de 14 paquets à 10 paquets, ce qui se traduit probablement par plus de pertes. La chute brusque dans l'OWD, donc le BDP, sera détectée par CUBIC à travers des pertes sur la CWND via des RTT plus courts. Cela pourrait expliquer la chute dans le débit. La CWND continue les oscillations périodiques avec des pics, et des plateaux de W_{max} moins importants passant de 182 à 175 paquets. Ainsi le débit chute de 1.47 Mbit/s (étant donné que le goulot d'étranglement est fixé à 1.5 Mbit/s) à 1,36 Mbit/s, soit une baisse de 7,48%.

Quant au délai dû aux *handovers* inter-orbitaux (pour la transition entre le cas d'utilisation 1 et 3 dans la sous-figure 4.6d), la variation de délai a un impact sur le débit qui passe de 1.47 Mbit/s à 1.35 Mbit/s, soit une diminution de 8.16%. De plus, le débit diminue puis reprend au niveau nominal. Comme cela correspond à une augmentation du délai nous maintenons la même explication que pour la transition entre les cas d'utilisation 1 et 2. Cependant, ici il s'agit bien d'un *handover* inter-orbital. Les données ont donc été remises à un satellite d'un autre plan orbital, ce qui pourrait probablement expliquer le fait que l'augmentation du délai soit accompagnée par plus de pertes qu'avec un *handover* intra-orbital donc une diminution du débit en réception. Ensuite, la CWND et le débit, diminuent encore deux fois. Pour la deuxième chute en débit, nous gardons la même explication que précédemment, la transition entre les cas d'utilisation 2 et 1, où l'on peut voir que l'OWD diminue de 56ms à 30.6ms, cf. la Figure 4.7. Pour la troisième chute en débit, l'OWD passe de 30.6ms à 41.17ms, cf. la Figure 4.7, en passant à un BDP plus important, cela pourrait être dû à des pertes supplémentaires (en comparaison avec la sous-figure 4.6b).

Pour le *handover* de l'inter-orbital à l'élévation dans la sous-figure 4.6e, l'OWD diminue de 56ms à 29ms. Comme le montre la Figure 4.7, nous gardons le même esprit d'explication que précédemment pour la transition du cas d'utilisation 2 à 1 de la sous-figure 4.6c. Le débit passe de 1.47 Mbit/s à

1.35 Mbit/s, soit une baisse de 8.16%. Cet impact est similaire à celui de l'intra-orbital (diminution de 7.48% de la valeur du débit).

4.4.0.2 Résultats pour la configuration à haut débit



(a) Délai dû à la variation d'élévation

(b) Délai dû au *handover* de l'élévation à l'intra-orbital

(c) Délai dû au *handover* de l'intra-orbital à l'élévation

(d) Délai dû au *handover* de l'élévation à l'inter-orbital

(e) Délai dû au *handover* de l'inter-orbital à l'élévation

FIGURE 4.8 Impact de la variation de délai due à différentes sources autres que la couture sur la CWND et le débit en réception pour une configuration à haut débit

Dans une configuration à haut débit, la *capacité_buffer* est à 1024 comme expliqué en 4.2, et le BDP correspondant (pour un OWD = 49ms) est à environ 1413 paquets (pour des paquets de 1040B sous la configuration par défaut de NS-2). Pour ce qui suit nous maintenons les mêmes analyses et interprétations que pour la sous-section 4.4.0.1.

Sur la Figure 4.8, nous avons également rassemblé les 3 différents cas d'utilisation de la variation de délai énoncés précédemment.

Par rapport à la contrepartie à bas débit, nous pouvons voir que dans la sous-figure 4.8a, le délai de variation de l'élévation n'affecte pas non plus les mesures de performance de TCP *i.e.* CWND ni le débit instantané de réception pour la configuration à haut débit. Dans ces cas, les conclusions des cas à bas débit s'appliquent.

En revanche, pour les *handovers* intra-orbitaux (transition des cas d'utilisation de 1 à 2, dans la sous-figure 4.8b), nous remarquons un impact à la fois sur le débit et sur la CWND. Le débit est passé de 118.19 Mbit/s (étant donné que le goulot d'étranglement est fixé à 120 Mbit/s) à 116.69 Mbit/s, soit une baisse de 1,27%. À partir de la transition entre les cas d'utilisation 1 et 2, dans la sous-figure 4.8c, le débit passe de 118.18 Mbit/s à 111.98 Mbit/s, soit une diminution de 5.25%.

Pour la transition inter-orbitale, (transition du cas d'utilisation 1 à 3 dans la sous-figure 4.8d), nous gardons la même explication que pour le *handover* intra-orbital (transition des cas d'utilisation 1 à 2). L'impact est légèrement plus perceptible ici, puisque le *handover* inter-orbital induit plus de délai que le *handover* intra-orbital. La transition de 1 à 3 a un impact sur le débit qui passe de 118.18 Mbit/s à 113.81 Mbit/s, soit une diminution de 3.7%. En outre, la CWND et le débit diminuent puis remontent plusieurs fois par la suite. Pour la transition inter-orbitale (transition du cas d'utilisation 3 à 1 dans la sous-figure 4.8e), nous pouvons remarquer que le premier accroc correspond à un *handover* d'élévation vers intra-orbital (la transition du cas d'utilisation 1 à 2, comme dans la sous-figure 4.8b). En ce qui concerne le second accroc, nous gardons la même explication que précédemment (la transition du cas d'utilisation 2 à 1). L'impact du *handover* inter-orbital induit plus de délai que le délai intra-orbital. Le débit passe de 118.19 Mbit/s à 110.88 Mbit/s, soit une baisse de 6.19 %.

4.5 Conclusion

Les quatre sources de variations de délai étudiées, à savoir la variation d'élévation, le délai du *handover* intra-orbital, le délai du *handover* inter-orbital et le délai du *handover* de la couture, ont des impacts différents sur une connexion CUBIC. Cet impact a pu être observé à travers la CWND et/ou le débit de réception instantané. Le *handover* en élévation se produit en continu, tandis que pour les terminaux fixes, les *handovers* intra-orbitaux environ toutes les 10 mn, les *handovers* inter-orbitaux toutes les 2h et les *handovers* dus à la couture se produisent au maximum trois fois et au minimum deux fois sur 24 heures, dans le cas étudié.

Nous avons vu que pendant le passage de la couture la durée de transfert des fichiers subit une augmentation significative pour les petits fichiers dont la taille est inférieure à 100kB, alors que pour les grandes tailles de fichiers, l'impact est marginal.

Comme indiqué dans les sous-sections 4.4.0.1 et 4.4.0.2, le délai de variation de l'élévation n'a pas d'impact sur la communication. Le débit utile disponible est pleinement exploité par TCP. L'évolution de la CWND de TCP est la même que celle attendue par CUBIC sur un lien à délai constant.

Néanmoins, dans notre cas de figure, le *handover* de l'élévation à l'intra-orbital entraîne une diminution de 1.27% du débit instantané pour une configuration à haut débit. Le transfert intra-

orbital vers l'élévation entraîne une diminution de 7.48% du débit instantané pour une configuration à bas débit et de 5.25% pour une configuration à haut débit.

En revanche, le *handover* de l'élévation à l'inter-orbital peut entraîner une baisse de 8.16% pour une configuration à bas débit. Quant à la configuration à haut débit, le *handover* élévation vers inter-orbital entraîne une baisse de 3.7% du débit instantané. Le *handover* inter-orbital vers l'élévation entraîne quant à lui une diminution de 6.19% du débit instantané.

La réduction du débit s'explique par l'évolution de la CWND de TCP qui présente des valeurs moyennes différentes pour divers points de fonctionnement. Cela peut être dû aux différents délais des chemins de bout-en-bout qui sont exploités.

Nous remarquons que pour la configuration à haut débit, l'impact des différentes variations de délai est généralement plus faible. Il peut y avoir une exception avec le *handover* de l'élévation à l'intra-orbital.

Pour finir, si l'impact sur la variation du débit peut être peu significative dans certains cas, son impact sera dépendant de l'application en jeu.

CHAPITRE 5

SIMULATION ET ÉMULATION DE L'IMPACT DE LA VARIATION DE DÉLAI SUR CUBIC ET BBRV1

5.1 Objectif

Dans ce chapitre, nous allons poursuivre l'étude de l'impact des différentes sources de variations de délai dans la constellation de satellites *Iridium NEXT* en nous focalisant sur les différentes variantes des protocoles de transport CUBIC et BBRv1.

5.2 Outils de simulation et d'émulation, et contexte de l'expérimentation

Ici le scénario vise à étudier l'impact des différentes sources de variations de délai sur CUBIC et BBRv1 via l'envoi en continu, en utilisant les paramètres définis ci-dessous. Cela est effectué sous NS-2, dans un premier temps. Des émulations sous OPENSAND ont été également entreprises permettant d'avoir une assimilation qui s'approche encore plus de la réalité. En environnement d'émulation, CUBIC est issu de l'implantation par défaut dans le kernel linux. Il en est de même pour BBRv1 pour un kernel version supérieure à 4.9.

Il est important de noter que les débits d'envoi ne sont limités que par le contrôle de congestion, et pas par l'application comme on peut le voir sur la Figure 2.8a.

La congestion et le routage sont deux éléments classiques qui auront un impact supplémentaire sur le délai. Nous ne présenterons pas ces phénomènes ici afin de rester concentrés sur les propriétés spécifiques aux constellations.

Pour étudier l'impact des différentes sources de variations de délai, nous nous sommes intéressés au transfert de données; cela nous permettra donc de percevoir concrètement cet impact. Les métriques par conséquent sont le débit instantané et la CWND. Le débit instantané permet d'avoir une idée détaillée et précise du comportement de TCP en ce qui concerne l'impact de ces sources de variation de délai. Nous nous concentrerons sur une mesure sensible au délai : le débit instantané reçu du point de vue de l'application, mesuré sur une fenêtre glissante de 25 secondes.

Nous définissons deux ensembles de débits (bas débit et haut débit, avec un rapport de 33, pour

des raisons de limitations systèmes (virtuels)) où bas débit correspond à des up- et down-links de 1.5 Mbit/s et 25 Mbit/s pour les ISLs. Pour le haut débit, les up- et down-links sont de 49.5 Mbit/s en raison de la limitation du débit de la plateforme de virtualisation de l'émulation. Par ailleurs, les constellations de satellites LEO doivent faire aussi bien que les constellations GEO [156], [157]. En outre, certains projets de constellations de satellites prévoient d'offrir au moins 50 Mbit/s en down-link et 10 Mbit/s en up-link [158]. Pour les ISLs, nous avons utilisé une valeur de 825 Mbit/s.

La durée de la simulation ou émulation est [9 :14]h, en raison de la périodicité des événements de variation de délai, à partir d'un profil de délai de 24 heures comme indiqué sur la figure 3.1. Nous avons zoomé sur l'intervalle [9 :14]h qui englobe tous les événements pouvant être représentés.

Le Tableau 5.1 regroupe les paramètres des scénarios de simulation et d'émulation.

TABLE 5.1 Paramètres des scénarios de simulation et d'émulation

Paramètres	Valeur
Métrique observée	Débit en réception (25s) et CWND
Durée	[9 :14]h
Capacité file d'attente = BDP (paquets de 1500B)	Bas débit : 116
	Haut débit : 2389
Émetteur/Récepteur	Londres / Atlantique
Protocole de transport	Émetteur : CUBIC ou BBRv1
	Récepteur : SACK
Débits Up/Down-links	Bas débit : 1.5 Mbit/s
	Haut débit : 49.5 Mbit/s
Débits ISLs	Bas débit : 25 Mbit/s
	Haut débit : 825 Mbit/s
Nombre de flux	1
Outil	NS-2 et OPENSAND

5.3 Impact des différentes sources de variations de délai sur CUBIC et BBRv1

Dans cette section, nous allons aborder l'impact des différentes sources de variations de délai sur CUBIC et BBRv1 durant l'envoi en continu. Le scénario considère les configurations à bas débit et à haut débit avec NS-2 et OPENSAND.

Parmi les raisons pour lesquelles nous nous intéressons à BBR, il y a sans doute le fait que BBR estime le RTT par des sondages. De plus, la constellation de satellites présente une variation de délai intrinsèque. Cela permet donc de voir le comportement de BBR vis-à-vis d'un tel environnement.

5.3.1 Impact sur CUBIC sous NS-2

La Figure 5.1 montre le débit reçu au niveau applicatif pour CUBIC dans un scénario à bas débit. Nous pouvons remarquer de petites variations autour d'un débit moyen de 1.4 Mbit/s. Ces variations

sont les conséquences d'un taux de perte de 3.5 % (dû à l'utilisation d'un TCP *loss-based*). Ces pertes en débit sont liées aux variations de l'OWD, comme nous l'avons vu dans la sous-section 4.4.0.1.

Nous voulions voir le comportement à gros grain sur une durée plus importante qui comprend les différentes sources de variation de délai. Dans l'ensemble, CUBIC s'en sort bien face à des changements de délai notables et minimes dans une configuration à bas débit. Pour une configuration à haut débit, nous observons le même comportement avec un débit moyen de 46.5 Mbit/s et un taux de perte de 2 %.

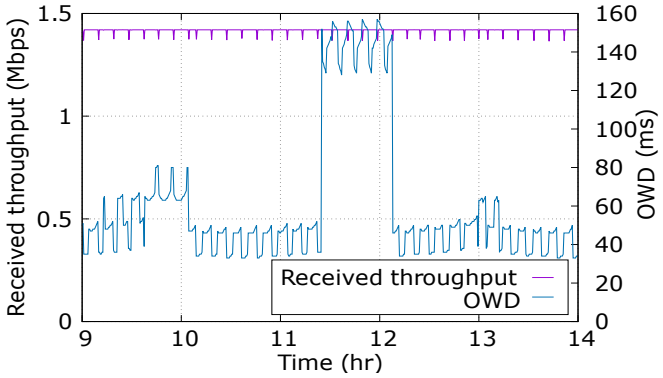


FIGURE 5.1 Débit en réception de CUBIC sous NS-2 pour une configuration à bas débit

Ces premiers résultats semblent montrer que CUBIC n'est pratiquement pas affecté par les variations de délai introduites par une constellation de type *Iridium NEXT*.

5.3.2 Impact sur CUBIC et BBRv1 sous OPENSAND

La plateforme OPENSAND permet de configurer les constellations de satellites via la trace des délais de paquets. Nous avons donc alimenté l'émulateur avec un fichier de trace implantant le comportement de la constellation *Iridium NEXT*. Les traces ont utilisées à la place de la configuration des ISLs. Du point de vue de l'émulation, nous avons utilisé iPerf 3.0 pour les deux extrémités de la connexion afin d'envoyer du trafic.

Ici nous cherchons à voir le comportement de CUBIC et de BBR par une émulation plus réaliste.

5.3.2.1 Configuration à bas débit

5.3.2.1.1 CUBIC

Dans cette sous-section, nous nous intéressons à CUBIC dans un scénario à bas débit et à l'impact des différentes sources de variation de délai.

La Figure 5.2 montre le débit reçu par CUBIC dans ce scénario. Ces résultats sont fondamentalement les mêmes que ceux décrits dans la Figure 5.1 puisque nous utilisons la même configuration mais avec un outil différent, plutôt plus réaliste, dans la mesure où l'outil d'émulation utilise directement l'implantation dans le *kernel* des algorithmes de contrôle de congestion.

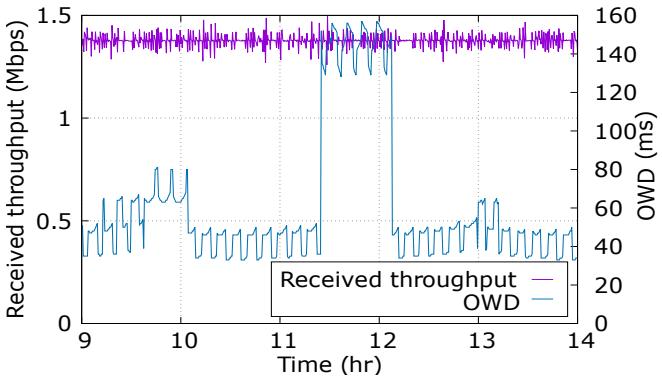


FIGURE 5.2 Débit en réception de CUBIC sous OPENSAND pour une configuration à bas débit

En tant qu'algorithme *loss-based*, CUBIC donne de bons résultats dans un contexte à bas débit et ne semble pas être affecté par les changements de délai, ce qui est cohérent avec les résultats de simulation.

Certaines pertes sont nécessaires pour que TCP fonctionne, même si CUBIC affiche un faible taux de perte. Comme CUBIC est un algorithme *loss-based*, l'étape de sondage pour plus de débit passe par l'inondation du lien et donc du *buffer*. Ainsi, les pertes sont un passage obligé pour déterminer le seuil du débit disponible à s'accaparer. Cela se traduit par les fluctuations en dessous de la valeur nominale. En raison de l'*overhead* de la pile de protocoles, le débit n'est pas optimal (1.5 Mbit/s) mais proche de celui-ci (94 %).

5.3.2.1.2 BBRv1

Dans cette sous-section, nous cherchons à voir le comportement gros grain de BBRv1 exposé aux différentes sources de variations de délai dans un environnement d'émulation. BBR est un algorithme hybride ralliant deux paramètres clés d'un lien réseau délai et capacité.

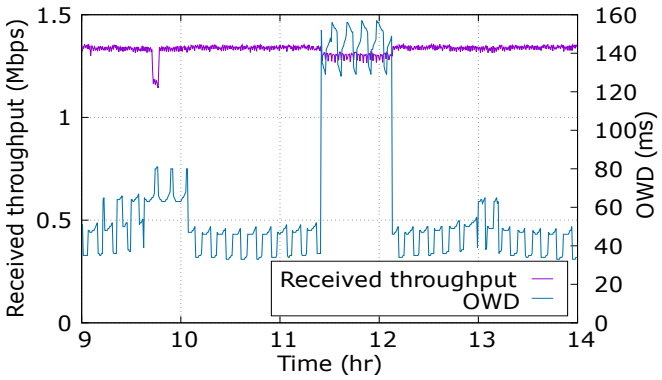


FIGURE 5.3 Débit en réception de BBRv1 sous OPENSAND pour une configuration à bas débit

La Figure 5.3 montre le débit obtenu par BBRv1 dans une configuration à bas débit. Nous pouvons remarquer que ce débit est un peu plus faible qu'avec CUBIC, et que certaines variations sont visibles.

Les variations de délai de type 3 (*handover* inter-orbital) et 4 (*handover* dû à la couture) entraînent une diminution du débit de 12.7 % et 3 %, respectivement.

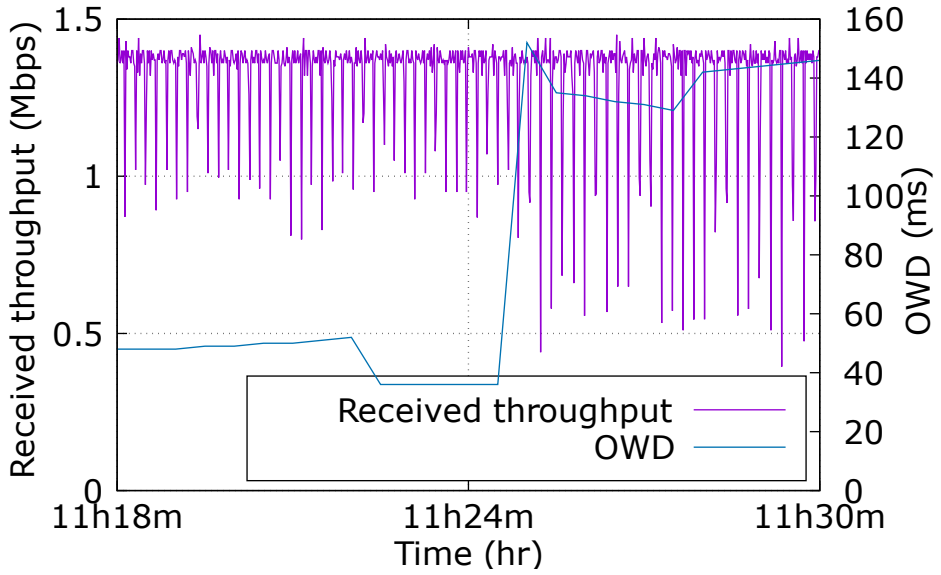


FIGURE 5.4 Zoom sur la transition par la couture du débit en réception (granularité 1s) de BBRv1 sous OPENSAND pour une configuration à bas débit

Comme on peut le voir sur la Figure 5.4 une explication à ces diminutions dans le débit serait le fait que l'augmentation de l'OWD donc le RTT et le passage périodique par la phase *Probe* RTT toutes les 10s où la CWND descend à 4 paquets pour une durée de 200ms + RTT. De ce fait, nous observons les chutes dans le débit avec plus d'ampleur qu'en dehors de la couture. Le débit obtenu est d'environ 1.33 Mbit/s. La couture semble avoir un impact faible sur le débit reçu, qui passe de 1.33 Mbit/s à 1.29 Mbit/s, soit une baisse de 3 %.

La même explication s'applique pour la transition du *handover* inter-orbital mais à moindre échelle. À ceci s'ajoute un phénomène pour lequel nous n'avons pas d'explications, la diminution aiguë du débit. Ce même type de phénomène se reproduit en répétant la même émulation à plusieurs reprises, mais sur d'autres intervalles.

5.3.2.1.3 Conclusion

En comparaison avec CUBIC, BBRv1 semble plus stable. Ceci serait dû, d'une part, au fait que BBR passe la plupart du temps dans la phase *Probe BW* où CWND_*gain* reste constant à 2, même si le *gain cycling* à 8 cycles $[1.25, 0.75, 1, 1, 1, 1, 1, 1]$ fait varier le *pacing_gain*. Ainsi cela permet d'obtenir plus de débit avec un *pacing_gain* à 1.25, puis de vider les files d'attente créées à 0.75 pour à la fin garder une file d'attente courte et une utilisation totale de la bande. D'autre part, cela pourrait être expliqué par le fait que CUBIC ait un facteur de diminution $\beta = 0.3$. Le débit fluctue encore plus, et cela affecte la convergence du débit vers une valeur nominale.

5.3.2.2 Configuration à haut débit

5.3.2.2.1 CUBIC

Dans cette sous-section, nous nous intéressons à CUBIC dans un scénario à haut débit que les nouvelles constellations de satellites permettent et aux conséquences qui en découlent.

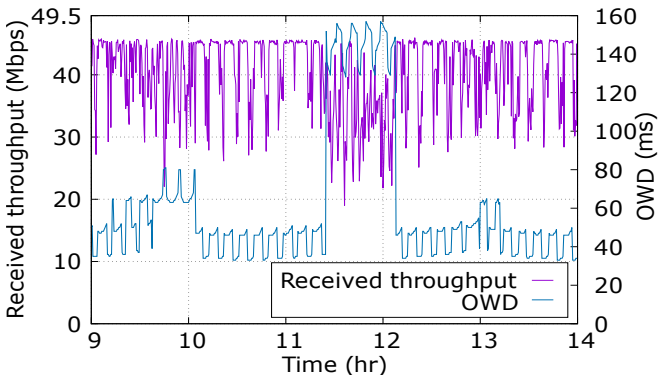


FIGURE 5.5 Débit en réception de CUBIC sous OPENSAND pour une configuration à haut débit

Les algorithmes de contrôle de la congestion *loss-based* augmentent constamment les données en vol afin de remplir le *buffer* du goulot d'étranglement. Ces algorithmes ont besoin de pertes de paquets pour fonctionner correctement et utilisent cela comme signe de congestion. Les algorithmes *loss-based* peuvent entraîner une dégradation des performances en raison du délai excessif de la mise en file d'attente et de la perte de paquets. Ces algorithmes sont très utilisés, ce qui a fonctionné relativement bien dans le passé avec un faible débit disponible et une petite taille de *buffer* du goulot d'étranglement.

La Figure 5.5 montre que CUBIC n'est pas aussi performant que dans une configuration à bas débit. La plateforme d'émulation OPENSAND (un peu plus réaliste) nous permet de constater de sérieuses dégradations du débit. Le débit moyen passe de 37 Mbits/s en dehors de la couture à 30 Mbit/s sous la couture. Ces pertes sont visibles dans les fluctuations du débit autour des variations de l'OWD dues aux *handovers* intra- et inter-orbitaux, tout particulièrement quand le délai et par conséquent le BDP diminuent. Comme nous l'avons expliqué précédemment dans la sous-section 4.4.0.1 pour la transition du 2 au 1 dans la sous-figure 4.6c, la diminution brusque du BDP entraîne la réduction de la CWND en raison des pertes.

Sur la Figure 5.6, nous avons zoomé sur la transition vers la couture pour voir l'évolution de la fenêtre de congestion CWND.

Nous pouvons remarquer que, après l'augmentation du délai (de 40ms à 145ms, le délai a plus que triplé), la CWND n'augmente pas de manière significative ; les plateaux ont été à la limite doublés passant de 0.5 MB à 1 MB. Comme ici il s'agit d'une transition entre les deux plans de couture, la communication est reroutée et passe par les satellites proches des pôles, ce qui explique que l'augmentation du délai soit accompagnée par plus de pertes que les autres *handovers*. De plus, la réactivité de TCP diminue en conséquence de cette augmentation du délai donc du RTT.

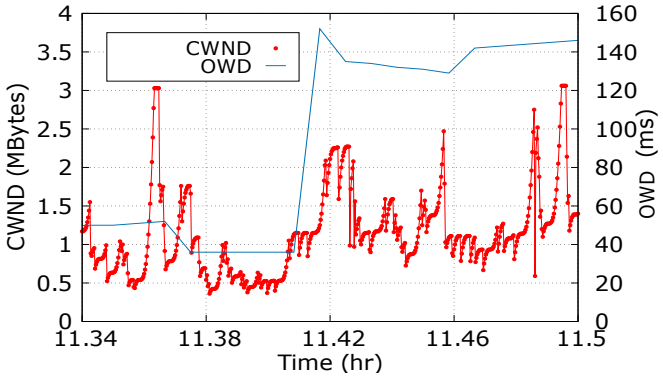


FIGURE 5.6 CWND de CUBIC sous OPENSAND pour une configuration à haut débit

La baisse du débit s'explique par le fait que l'augmentation du délai (ici le délai a triplé), ne se traduit pas par une augmentation de la CWND (qui n'a que doublé), le débit de bout-en-bout ne restera donc pas le même. Bien que CUBIC soit connu pour son agressivité, il ne s'empare pas de la capacité disponible.

5.3.2.2.2 BBRv1

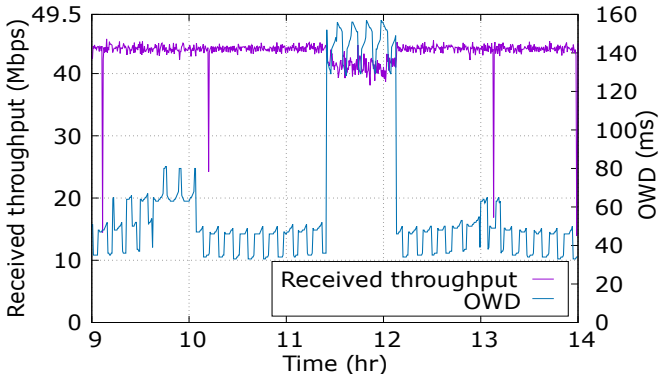


FIGURE 5.7 Débit en réception de BBRv1 sous OPENSAND pour une configuration à haut débit

Comme nous l'avons expliqué précédemment, comme BBRv1 est hybride, il a une meilleure compréhension de l'évolution du délai du réseau. Cela explique sa réactivité aux changements majeurs de délai comme on peut le remarquer sur la Figure 5.7.

Dans l'ensemble, BBRv1 fait mieux que CUBIC, puisqu'il atteint un débit moyen reçu de 44 Mbit/s.

BBR est un algorithme de contrôle de congestion hybride, il cherche un point de fonctionnement avec le débit le plus élevé possible et un délai faible : le point de fonctionnement optimal de Kleinrock [75], dans le but d'avoir des pertes moindres que les algorithmes *loss-based* dont le point de fonctionnement est axé sur les pertes. Sur la Figure 5.8, tout comme en bas débit (dans la sous-section 5.3.2.1.2) la couture avec une augmentation importante du délai entraîne des pertes en débit. Nous maintenons la même explication qu'à bas débit quant à l'effet de la couture.

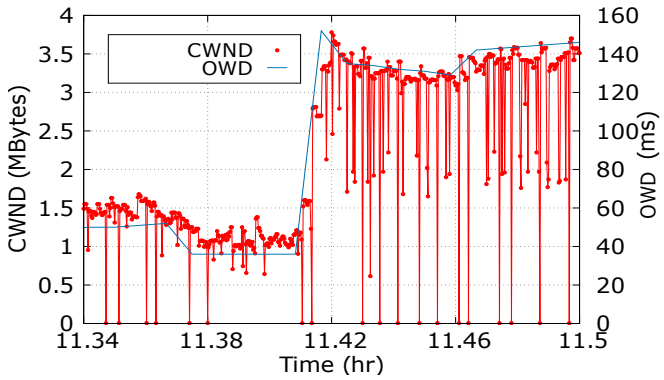


FIGURE 5.8 CWND de BBRv1 sous OPENSAND pour une configuration à haut débit

Sur la Figure 5.8, nous pouvons voir le passage par la couture à travers la CWND. La CWND de BBRv1 suit l'évolution de l'OWD. Quand l'OWD augmente, le BDP aussi, et comme BBR passe la plupart du temps (environ 98%) dans la phase *Probe Bandwidth* et qu'il maintient un cap des données en vol à 2× BDP, l'estimation de la bande accompagne l'OWD dans cette augmentation. Le débit est donc assez stable sur la Figure 5.7. La diminution du débit sous la couture pourrait se voir en effectuant une moyenne et quand la CWND descend à des valeurs très faibles ce qui est probablement dû au passage par la phase de *Probe* RTT.

Pour l'impact des autres sources de variations de délai telles que le *handover* inter-orbital sur la Figure 5.7, nous qualifierons ces chutes ponctuelles du débit d'épiphénomènes dues probablement à des pertes ponctuelles.

5.3.2.2.3 Conclusion

Par rapport à CUBIC, BBRv1 présente plus de stabilité et atteint des débits plus importants. BBRv1 atteint des débits aux alentours de 44 Mbit/s sous la couture, alors que pour CUBIC c'est de l'ordre de 37 Mbit/s. Sous la couture, pour BBRv1 le débit atteint est de l'ordre de 40 Mbit/s, tandis que CUBIC le débit est aux alentours de 30 Mbit/s.

5.4 Conclusion

Dans ce chapitre, nous nous sommes concentrés sur l'impact de la dynamique de la topologie de la constellation sur CUBIC et BBRv1 du point de vue de la variation de délai. Notre expérimentation a été menée par simulation sous NS-2 et par émulation sous OPENSAND. Deux scénarios ont été mis en place en utilisant une constellation de satellites *Iridium NEXT*.

Nous avons montré que CUBIC, fondé sur les pertes, peut souffrir de ces variations de délai, en particulier pour les scénarios à haut débit. D'autre part, BBRv1, qui utilise un algorithme hybride, semble être plus efficace. Ses performances ne sont pas profondément affectées par les variations de délai et il est assez réactif en dépit des délais et de leurs variations de bout-en-bout. Ce comportement est visible dans les deux configurations du scénario.

Nos résultats pourront être reproduits à l'aide des outils open source NS-2, OPENSAND et iPerf 3.0 avec les simulations et les paramètres d'émulation appropriés indiqués dans les sous-sections 3.3 et 5.2.

Ces résultats reposent sur la version v1 de BBR qui a été critiquée pour son manque d'équité. Les évaluations ultérieures vont prendre en compte la nouvelle version de BBR qui est censée être plus équitable que BBRv1. Nous aborderons la question de l'équité de manière plus générale, car nous avons observé que différentes variantes de TCP, qui sont encore largement déployées, peuvent présenter des performances différentes.

Comme l'outil d'émulation OPENSAND nous permet d'avoir une représentation plus réaliste et que NS-2 ne comprend pas toutes les saveurs de TCP présentes dans le kernel (linux), nous allons donc pour la suite de la thèse nous concentrer sur l'émulation.

CHAPITRE 6

ÉQUITÉ INTRA- ET INTER-PROTOCOLAIRES POUR CUBIC ET BBRV2

6.1 Objectif

Dans ce chapitre, nous allons nous intéresser plus spécifiquement à l'étude de l'équité et à l'impact des flux concurrents. Pour ce faire, nous allons regarder l'équité intra- et inter-protocolaires pour CUBIC et BBRv2. L'objectif est donc d'observer les performances de ces protocoles de transport.

6.2 Outil d'émulation et contexte de l'expérimentation

Les expérimentations dans ce chapitre ont été effectuées avec OPENSAND.

La version BBRv1 est désormais considérée comme obsolète. Des résultats d'émulation l'ont confirmé. Ces limitations s'étendent également à un contexte de constellations de satellites. Nous nous sommes donc intéressés à la nouvelle version de BBR la v2 qui tente de remédier aux problèmes de la première version, comme discuté dans la Section 2.3.2.5.2.

Le *repository* Linux de Google[1] contenant l'implantation de BBRv2 a été utilisé dans cette évaluation. L'implantation par défaut de CUBIC dans Linux a été retenue pour étudier l'équité inter-protocole de BBRv2.

L'une des principales améliorations de BBRv2 est l'utilisation d'une variante d'ECN pour la notification de congestion : DCTCP/L4S. DCTCP/L4S est une variante qui utilise les longueurs instantanées des files d'attente sur les nœuds de transit (routeurs et commutateurs) pour un marquage ECN, et utilise une loi de contrôle différente sur les points terminaux pour ajuster la fenêtre de congestion par rapport à TCP standard avec ECN. BBRv2 adopte cette variante d'ECN. Donc, quand les connexions subissent une congestion cela se matérialisera par un marquage ECN de type DCTCP/L4S, plutôt qu'à un marquage ECN RFC3168 [159]. Comme BBRv2 et CUBIC utilisent deux types de marquage différents, nous sommes dans l'impossibilité d'avoir deux types de marquage hétérogènes et nous n'allons donc pas utiliser le marquage par ECN.

1. https://github.com/google/bbr/tree/v2alpha-experimental-pacing

Dans la même veine, nous nous sommes intéressés au transfert de données pour avoir un aperçu du partage de la bande entre les différents flux. La métrique observée est donc le débit en réception avec une granularité de 25 s.

Pour ce scénario, deux flux concurrents sont considérés.

Il est important de noter que les débits d'envoi ne sont limités que par le contrôle de congestion, et non par l'application comme on peut le voir sur la Figure 2.8a.

La modélisation de la capacité de la file d'attente est fonction du BDP : $f(\text{BDP}) = \alpha \times$ BDP où BDP = 2×OWD×Débit

TABLE 6.1 Valeurs du facteur α

α	0.05	0.1	0.2	0.33	0.5	1	2

Pour les modélisations de la capacité des *buffers* en fonction du BDP, elles sont fondées sur l'OWD (en prenant la valeur maximale, donc de l'OWD de l'entrée sous la couture de la Figure 6.1) et pas le RTT. En effet, le RTT, à son tour, est influencé par la valeur de la taille des *buffers*. Cela dit, nous avons essayé de voir si une modélisation des *buffers* fondée sur le RTT[2] donnerait de résultats différents, pour un scénario à OWD constant = 143ms (valeur maximale de l'OWD, sous la couture). Le moins que l'on puisse dire, c'est que nous avons le même comportement que pour des modélisations de la capacité des *buffers* en fonction de l'OWD.

Concernant l'émetteur et le récepteur situés, respectivement, à Londres et l'océan Atlantique cela est pris en compte dans l'OWD de bout-en-bout. Nous avons deux configurations possibles à OWD constant ou OWD variable. L'OWD constant peut prendre deux valeurs : 25ms = valeur minimale hors-couture, 143ms = valeur maximale sous la couture. Le profil variable est à la transition lors du passage par la couture comme le montre la Figure 6.1.

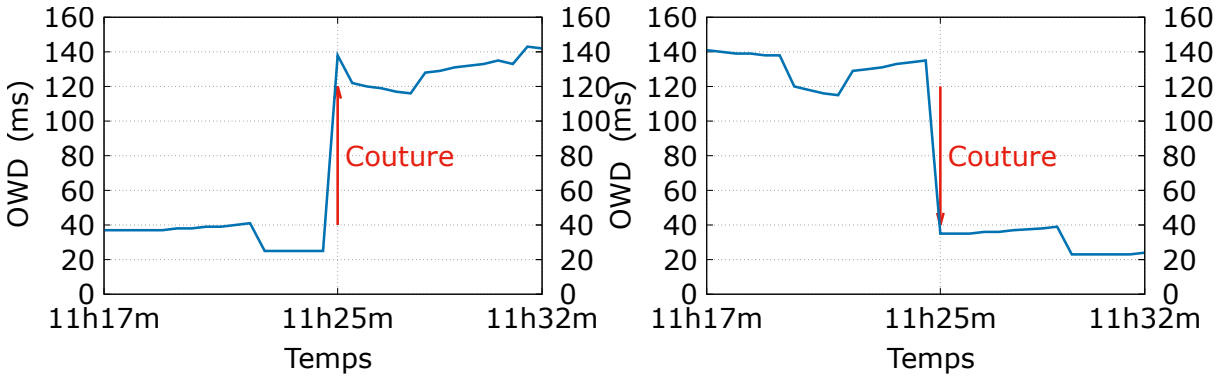


FIGURE 6.1 Profil de l'OWD variable pour 15 mn d'émulation pour les deux transitions passage par la couture et sortie de celle-ci

La durée émulée est de 15 mn, autour de la transition de la couture, de façon à avoir la couture au milieu. (7 mn30s avant et 7 mn30s sous la couture).

2. RTT du *ping* à une capacité du buffer = 1 BDP

Selon [160] la modélisation de la capacité des *buffers* (pour nous les *buffers* de la couche 2 au niveau du Satellite Terminal (ST) et de la GW) joue un rôle important dans l'équité dans un contexte multi-flux mettant en jeu un flux BBR. Dans notre cas, nous nous sommes limités à deux flux. Nous avons essayé de reproduire leurs résultats, c'est-à-dire en reprenant les valeurs du délai et donc des capacités des *buffers*. Mais en raison d'une limitation de la plateforme, nous ne pouvions pas configurer un débit à 1 Gbit/s (la valeur prise dans le papier [160]). Par ailleurs, nous avons remarqué que les capacités des *buffers* étaient si faibles que les pertes engendrées reflétaient des sommes de débits atteints par les deux flux en concurrence non nominales, c'est-à-dire < 49.5 Mbit/s (valeur prise pour les débits de nos liens montants et descendants). De ce fait, nous nous sommes intéressés qu'aux scénarios mettant en jeu les délais relatifs à la constellation en question qui sont bien supérieurs à ceux de [160], et donc forcément des capacités de *buffers* plus importantes.

Cela reflète l'importance de la configuration du paramètre capacité des *buffers*; nous le retraçons dans le tableau suivant :

TABLE 6.2 Valeurs des capacités de *buffers* en paquets en fonction de l'OWD et de α

α	0.05	0.1	0.2	0.33	0.5	1	2
Pour OWD = 143ms	59	118	236	400	590	1180	2360
Pour OWD = 25ms	11	21	42	-	104	207	414

Pour un profil de OWD constant, les configurations de capacité de *buffers*, énoncées dans le Tableau 6.2, ont été utilisées. Pour le profil d'OWD variable en entrée sous la couture, nous avons testé avec des valeurs de capacités de *buffers* correspondant aux délais que nous pouvons trouver dans le Tableau 6.2.

Nous avons aussi testé des cas de figure où nous avons un OWD constant = 25ms et avec les capacités de *buffers* correspondant à un OWD de 143ms. De façon symétrique, à OWD constant = 143ms, nous avons mené les expérimentations avec les capacités de *buffers* correspondant à un OWD de 25ms.

Pour un profil de l'OWD variable en sortie et en entrée de la couture, nous avons pris des capacités de *buffers* dans les intervalles du Tableau 6.2.

Le Tableau 6.3 regroupe les paramètres des scénarios d'émulation.

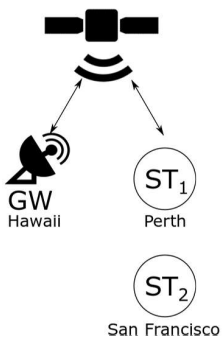


FIGURE 6.2 Schéma expérimental

TABLE 6.3 Paramètres des scénarios d'émulation

Paramètres	Valeur
Métrique observée	Débit en réception (25s)
Durée	15 mn
Capacité file d'attente	f(BDP)
Émetteur/Récepteur	Londres / Atlantique
Protocole de transport	Émetteur : CUBIC ou BBRv2
	Récepteur : SACK
Débits Up/Down-links	49.5 Mbit/s
Débits ISLs	825 Mbit/s
Nombre de flux	2
Outil	OPENSAND

6.3 Partage inter-protocolaire

Dans cette sous-section, nous allons observer le partage de la bande entre deux flux utilisant des protocoles différents. Les scénarios mettent en jeu des flux à RTT égaux et/ou différents avec des OWD constants ou variables.

6.3.1 À RTT égaux

6.3.1.1 CUBIC *vs* BBRv2

6.3.1.1.1 À OWD constant

6.3.1.1.1.1 À OWD constant = 25ms

TABLE 6.4 Valeurs des débits de CUBIC *vs* BBRv2 en fonction des capacités de *buffers* en paquets pour un OWD constant = 25ms, à RTT égaux

α	0.05	0.1	0.2	0.5	1	2
Capacité des *Buffers* (paquets)	11	21	42	104	207	414
Débit CUBIC (Mbit/s)	4.1	8.9	18.8	32.5	32.5	34.7
Débit BBRv2 (Mbit/s)	2.3	4.8	8.4	10.2	10.2	8.9
Débit total (Mbit/s)	6.4	13.7	27.2	42.6	42.6	43.57

Pour un OWD = 25ms, nous obtenons un débit total moyen inférieur à 27 Mbit/s pour des capacités de *buffers* inférieures à 104 paquets. Dans tous les cas, CUBIC domine la bande. Les débits moyens globaux se détériorent (inférieurs à 28 Mbit/s) pour des tailles de *buffers* inférieures à 42 paquets.

Nous avons aussi testé des cas de figures où l'on a un OWD constant = 25ms et des capacités de *buffers* correspondant à un OWD de 143ms. Quelle que soit la capacité des *buffers*, CUBIC domine également la bande.

La v2 de BBR rompt avec le cap de données en vol à 2× BDP de la v1 pour inclure les paramètres *inflight_hi* et *inflight_lo*. Le cap *inflight* explique que BBRv1 ne puisse avoir sa juste part de la bande avec un flux concurrent ayant un algorithme de congestion *loss-based*.

Pour le scénario pour lequel les capacités de *buffers* correspondent à un OWD de 25 ms, les capacités sont faibles (facteur $\alpha \in \{0.05, 0.1, 0.2\}$). Par ailleurs, l'OWD constant à 25ms est faible ; la réactivité est accentuée pour les deux flux TCP. Les pertes par saturation du *buffer* sont récurrentes et par conséquent les reprises sur erreur sont enclenchées.

La reprise sur erreur pour CUBIC consiste en une réduction multiplicative par $(1-\beta)$ (avec $\beta = 0.3$), tandis que pour BBRv2 ceci repose principalement sur la mise à jour des bornes des données en vol permises *inflight_hi* et *inflight_lo*. Cela explique la limitation en débit pour les deux types de TCP.

Nous observons que CUBIC domine la bande même avec des capacités de *buffers* limitées. Parmi les principales limitations connues de BBRv1 il y a son agressivité durant la phase de *startup* avec l'augmentation exponentielle (un *pacing_gain* et CWND_*gain* $= 2/ln(2) \approx 2.89$) qui crée une file d'attente et ne laisse plus la place aux flux concurrents.

BBRv2 reprend cette agressivité durant la phase de *startup* en limitant les données en vol à un plafond *inflight_hi* comme le volume maximal estimé sûr en vol. Le plafond *inflight_hi* est déterminé, si le taux de perte est trop élevé (> 2%) ; ainsi BBRv2 se comporte comme un algorithme *loss-based*.

Ainsi, BBRv2 sort de la phase *startup* si *inflight_hi* a été déterminé ou quand la valeur maximale du débit estimé n'augmente pas plus de 25% sur 3 RTT consécutifs.

Par la suite, BBRv2 entre dans la phase *drain* (tout comme BBRv1) pour vider la file d'attente en maintenant un faible *pacing_rate* pour évacuer rapidement l'excès en vol (*pacing_gain* $= ln(2)/2 \approx 0.35$ et CWND_*gain* est inchangé $= 2/ln(2)$), jusqu'à ce que *inflight* $\leq min$(BDP estimé, *headroom*× *inflight_hi*) (*headroom*= 0.85). Cette dernière condition fait que BBRv2 n'a plus d'avantage compétitif avec un algorithme *loss-based* tel que CUBIC, que cela soit avec des capacités de *buffers* fondés sur un OWD de 25ms ou de 143ms.

De plus, comme les algorithmes *loss-based* opèrent à partir du moment où le *buffer* est saturé et que les paquets sont perdus, cela ne laisse pas la possibilité à BBRv2 d'avoir une occupation équitable et donc d'avoir une bonne valeur de l'estimation du RTT du lien. RTT_*min* ne peut donc pas être mesuré correctement dans *probe* RTT, si un autre flux crée un délai supplémentaire dans la file.

Cela pourrait expliquer le fait que CUBIC domine la bande.

6.3.1.1.1.2 À OWD constant = 143ms

TABLE 6.5 Valeurs des débits de CUBIC *vs* BBRv2 en fonction des capacités de *buffers* en paquets pour un OWD constant = 143ms, à RTT égaux

α	0.05	0.1	0.2	0.34	0.5	1	2
Capacité des *buffers* (paquets)	59	118	236	400	590	1180	2360
Débit CUBIC (Mbit/s)	6.4	9.1	14.4	20.0	26.1	31.8	28.5
Débit BBRv2 (Mbit/s)	26.2	26.4	23.8	21.0	15.5	10.6	14.3
Débit total (Mbit/s)	32.6	35.5	38.2	41.0	41.5	42.4	42.8

Nous observons une domination de BBRv2 au départ (facteur $\alpha \leq 0.2$, et une capacité du *buffer* ≤ 236). Par la suite, on a une équité aux alentours de $\approx$ BDP/3 (correspondant à une capacité du *buffer* = 400). Enfin, CUBIC prend le dessus. On remarque aussi que la bande moyenne occupée par les deux flux conjointement (dernière ligne du Tableau 6.5) croît en partant de 32.6 Mbit/s pour une capacité de *buffer* à 59 paquets jusqu'à 42.79 Mbit/s à 2360 paquets de capacité de *buffer*.

Nous avons aussi testé des cas de figure dans lesquels nous avons un OWD constant = 143ms et des capacités de *buffers* $\in \{11, 21, 42, 104, 207, 414\}$. Pour les capacités ≤ 207, BBRv2 domine la bande. Pour une capacité > 207, c'est CUBIC qui domine la bande. Mais, dans les deux cas de figure, le débit total atteint est sous-optimal.

Pour les capacités de *buffers* où CUBIC domine nous conservons la même explication que précédemment. Quand BBRv2 emporte la majorité de la bande sur CUBIC, cela se joue probablement dans la phase *probe up* durant le sondage du débit où l'*inflight_probe* croît exponentiellement, pour BBRv2. BBRv2 réduit par la suite la file d'attente créée en sortant de cette phase si *inflight_hi* a été déterminé ou bien si la file d'attente estimée est assez élevée (*inflight* $> 1.25\times$ *estimated_bdp i.e.* un *pacing_gain* à 1.25). Cela s'observe dans le cas où les pertes ont dépassé la limite de 2%, avec un *pacing_gain* à 0.75 jusqu'à ce que *inflight* $\leq min$(BDP estimé, *headroom* $\times$ *inflight_hi*). A contrario, avec CUBIC, en cas de perte, la CWND est réduite de 30%. Cela est fait ponctuellement dès que les pertes ont été détectées sans différé contrairement à BBRv2 qui attend que les pertes soient inférieures à 2%. Dans le cas où la condition de sortie de la phase *probe up* est le fait que la file d'attente estimée soit assez élevée (*inflight* $> 1.25\times$ *estimated_bdp*), cela pourrait impliquer que BBRv2 prenne plus que sa juste part. En conséquence, CUBIC subit soit des pertes auquel cas il est dans l'incapacité de relancer sa CWND, soit il est dans l'impossibilité d'accaparer plus de bande étant donné que BBRv2 en saisit la plupart. La raison possible derrière la limite séparatrice de la capacité du *buffer* à $1/3\times$ BDP est le fait que la file d'attente créée durant la phase *probe up* à $1.25\times$ BDP soit de l'ordre de $0.25\times$ BDP. Au-delà d'une capacité de *buffer* $0.25\times$ BDP, BBRv2 ne peut pas augmenter *inflight* plus que $1.25\times$ de la bande sondée ; CUBIC domine alors.

6.3.1.1.2 À OWD variable : passage par la couture

- Modélisation pour des capacités de *buffers* avec un OWD = 143ms :
  Pour des capacités de *buffers* supérieures à 118 paquets, CUBIC monoplise la bande, tandis que pour des capacités testées $\in \{59,118\}$, hors couture CUBIC monopolise la bande et sous la couture c'est plutôt BBRv2. Atteindre l'équité à $\approx$ BDP/3 n'est pas possible car pour des capacités de 236 paquets et 590 paquets ($236 <$ BDP/3 < 590) CUBIC domine la bande. Le débit total atteint est plus ou moins proportionnel aux capacités des *buffers*.
  Pour un OWD variable à la sortie de la couture, nous avons remarqué la même tendance à un cas de figure près. La domination de CUBIC commençait plutôt à partir d'une capacité de 590 paquets, et pas 236 paquets comme présenté ici dans le Tableau 6.6.
- Modélisation pour des capacités de *buffers* avec un OWD = 25ms :
  Pour cette configuration nous avons des capacités de *buffers* moins importantes, on constate

que hors-couture (OWD d'environ 25ms) CUBIC monopolise la bande. Sous la couture BBRv2 prend le dessus.
Pour des capacités de *buffers* moins importantes (inférieures ou égales à 21) les débits totaux atteints sont sous-optimaux.
Nous voyons que pour cette configuration, avec des capacités de *buffers* moins importantes comme dans la sous-section 6.3.1.1.1.1, nous ne pourrons pas atteindre une équité à BDP/3.
Nous avons une concordance entre les résultats avec des délais constants et les différentes configurations de capacités de *buffers* et ceux en délais variables par tranche de délai en palier.
Il y a une exception pour une capacité de 236 paquets pour un OWD constant à 143ms où BBRv2 domine, tandis que pour un scénario à OWD variable et à capacité de *buffer* égale (236 paquets) sous la couture (donc un OWD de 143ms) c'est CUBIC qui domine. Cela est probablement dû à l'effet de la transition par la couture, ce qui fait que la limite d'équité à 1/3 du BDP n'est plus respectée.
Pour un OWD variable à la sortie de la couture, nous avons remarqué globalement la même tendance à deux cas de figure près. La domination de CUBIC n'a pas eu lieu pour la capacité à 414 paquets, contrairement à l'OWD variable à l'entrée à la couture Tableau 6.7, mais une continuité de la même tendance. Pour une capacité de 11 paquets, bien que les débits atteints se détériorent à faible capacité (et ce entre $[\![11;104]\!]$), BBRv2 domine la bande.

6.3.1.1.3 Conclusion

Pour des scénarios impliquant des OWD faibles *e.g.* 5ms (scénario de [160]), ou 25ms (scénario Océan Atlantique - Londres, valeur minimale de l'OWD hors-couture), donc des capacités de *buffers* en facteurs de BDP faibles, les débits totaux atteints sont faibles. On ne peut donc pas trancher quant à l'équité. Pour un délai à 25ms et donc des capacités de *buffers* supérieures ou égales à 21 paquets CUBIC domine la bande.

Pour des OWD constants moins faibles *e.g.* à 143ms (scénario Océan Atlantique - Londres, valeur maximale de l'OWD sous la couture), nous avons une domination de la bande par BBRv2 pour des capacité de *buffers* inférieures ou égales à $<$ BDP/3 puis équité $\sim$ BDP/3 puis domination de la bande par CUBIC.

La somme des débits atteints est croissante.

Pour des OWD variables :

- Modélisation des *buffers* avec le max OWD (= 143ms) : Pour des capacités de *buffers* supérieures à 118 paquets CUBIC monoplise la bande, tandis que pour des capacités testées $\in \{59,118\}$, hors couture CUBIC monopolise la bande et sous la couture c'est plutôt BBRv2. Le débit total atteint est plus ou moins proportionnel aux capacités des *buffers*.
- Modélisation des *buffers* avec le min OWD (=25ms) : Pour cette configuration nous avons des capacités de *buffers* moins importantes. On constate que hors-couture (OWD $\sim$ 25ms) CUBIC domine. En revanche sous la couture, BBRv2 prend le dessus.
  Pour des valeurs de capacités de *buffers* moins importantes (inférieures ou égales à 21 paquets)

les débits totaux atteints sont sous-optimaux.

Les conclusions précédentes peuvent se résumer dans le schéma présent sur la sous-figure 6.3a.

6.3.2 À RTT différents

6.3.2.1 CUBIC *vs* BBRv2

Considérons des RTTs différents, c'est-à-dire, dans le cas où l'OWD est constant (à 25ms ou à 143ms), ou variable (passage par la couture en entrée ou sortie) et les deux flux des RTT différents. Nous avons constaté que les conclusions sur les dominations restent inchangées par rapport des RTTs égaux. Elles sont résumées dans le schéma de la sous-figure 6.3b.

Nous avons étendu les scénarios pour des instants de début d'émulation décalés pour l'un des deux flux pour en voir l'impact sur la répartition du partage. Nous avons remarqué qu'à partir de la phase stable, le partage dans des conditions normales (c'est-à-dire avec des instants de début d'émulation égaux) est rétabli donc selon le schéma précédent de la Figure 6.3.

6.3.3 Conclusion

Pour deux flux hétérogènes, l'un *loss-based* (CUBIC) et l'autre *hybride* (BBRv2), le schéma de la Figure 6.3 résume le partage de la bande dans les différents scénarios étudiés. Nous observons un monopole assez prononcé pour CUBIC dans la majorité des cas de figure. Les caractéristiques inflexibles de BBRv1 ont fait de BBRv2 un peu trop flexible cédant ainsi sa part.

(a) À RTT égaux **(b)** À RTT différents

FIGURE 6.3 Partage de la bande entre CUBIC *vs* BBRv2 en fonction des capacités de *buffers* et du délai

6.4 Partage intra-protocolaires

Dans cette sous-section, nous allons voir le partage de la bande entre deux flux du même protocole à RTT identiques ou non et des OWD constants ou variables.

6.4.1 BBRv2 *vs* BBRv2

À RTT identiques, une équité existe entre les deux flux.

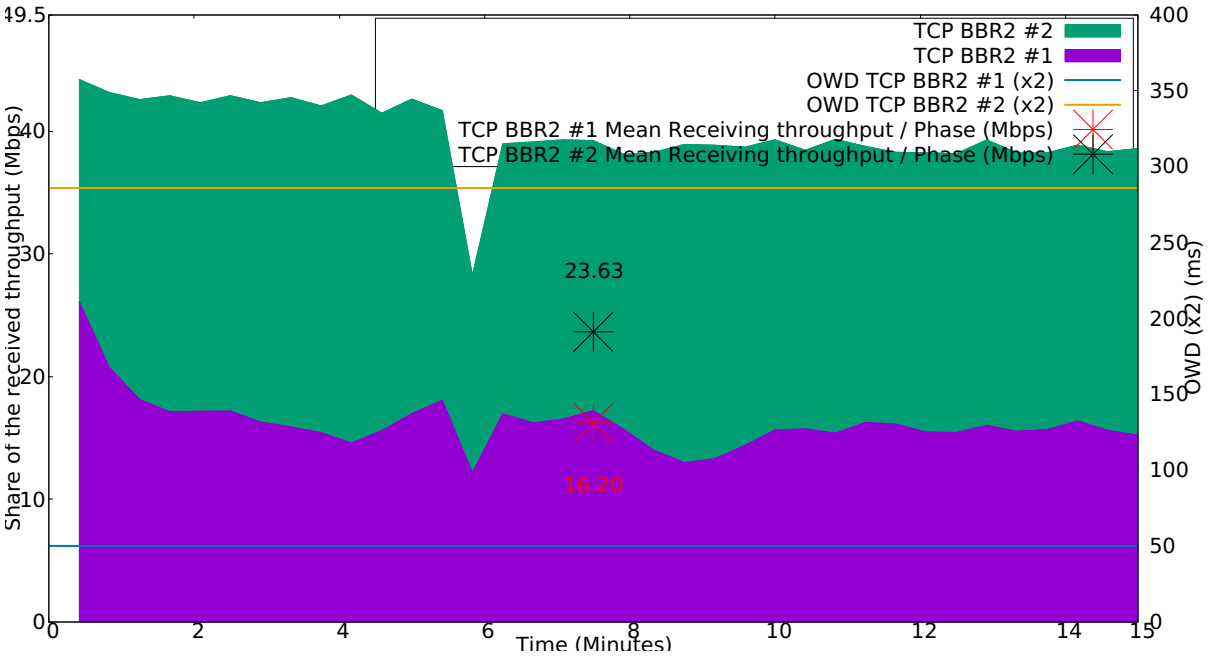


FIGURE 6.4 Partage du débit en réception entre deux flux BBRv2 à RTT constants différents en configuration haut débit

Par ailleurs à RTT différents, le flux dont le RTT est le plus important l'emporte et ce quelle que soit la capacité du *buffer* et à OWD constant ou variable. Cela s'explique par le fait que le flux dont le RTT est le plus important estime un BDP plus grand. Il est alors susceptible d'envoyer plus de données en vol durant les phases de *startup* et de *probe up*, et donc d'occuper plus la file d'attente provoquant ainsi des pertes pour le flux concurrent avec un RTT moins important. C'est ce que nous pouvons voir sur la Figure 6.4.

6.4.2 CUBIC *vs* CUBIC

Pour CUBIC, à RTT égaux, une équité existe entre les deux flux.

Sinon, le flux dont le RTT est le plus petit l'emporte et ce quelle que soit la capacité du *buffer* et à OWD constant ou variable. L'explication réside dans l'équation de CUBIC $W(t) = C(t-K)^3 + W_{max}$ (2.1) et son élément clé le paramètre K qui représente le temps que la fonction de la CWND met pour atteindre W_{max}, lorsqu'il n'y a pas eu de pertes. Le flux dont le RTT est le plus grand, mettra alors plus de temps avant d'atteindre la W_{max} alors que le flux avec un RTT moins important aura déjà augmenté sa fenêtre de congestion et se situera dans la partie convexe [161]. Nous pouvons le voir sur la Figure 6.5.

6.5 Conclusion

Dans ce chapitre, nous nous sommes focalisés sur l'équité intra- et inter-protocolaires.

Pour le partage inter-protocolaire, CUBIC VS BBRv2, à RTT égaux pour les différents cas de figure

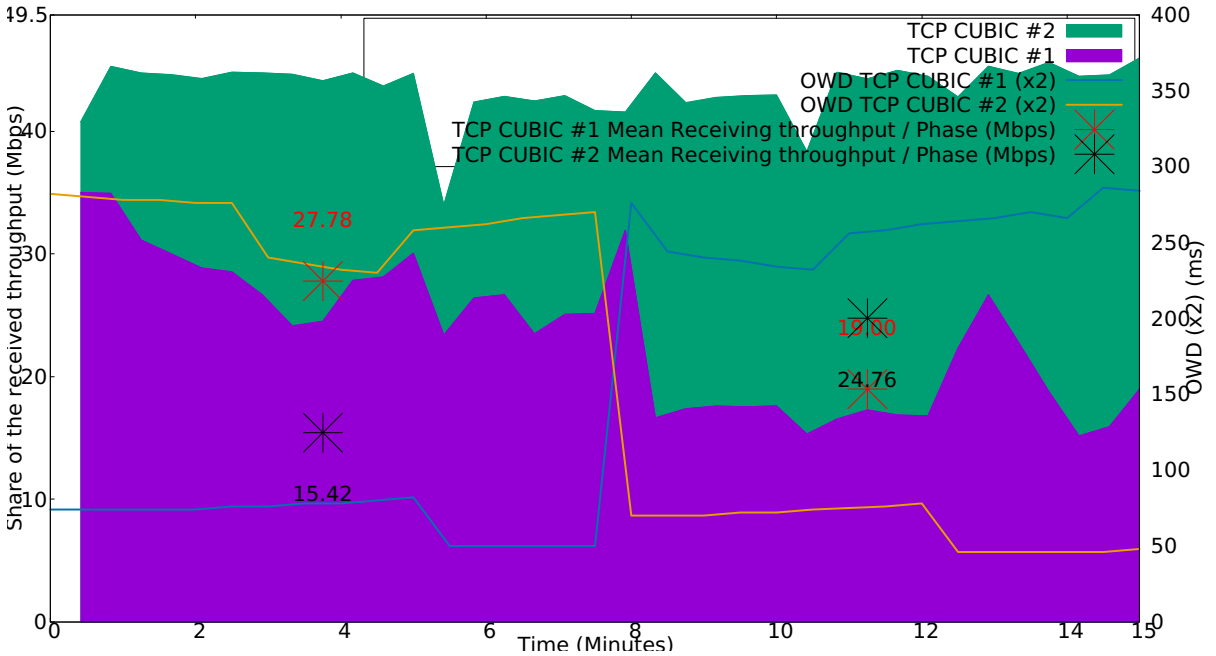


FIGURE 6.5 Partage du débit en réception entre deux flux CUBIC à RTT variables différents en configuration haut débit

CUBIC domine la bande sauf pour le cas où l'OWD est important et la capacité du *buffer* assez petite ($\sim < 0.25$ BDP). À RTT différents, la même tendance persiste avec une domination de CUBIC sauf dans le cas où l'OWD de BBRv2 est plus important que celui de CUBIC pour des capacités de *buffers* faibles. Cela soulève des questions quant aux améliorations dont BBRv2 a fait l'objet.

Pour le partage intra-protocolaire, à RTT identiques, une équité existe entre deux flux BBRv2 et pour CUBIC. Sinon, le flux dont le RTT est le plus important l'emporte et ce quelle que soit la capacité du *buffer*, à OWD constant ou variable. A contrario, pour CUBIC, à RTT différents, le flux dont le RTT est le plus petit l'emporte et ce quelle que soit la capacité du *buffer*, à OWD constant ou variable.

TABLE 6.6 Valeurs des débits de CUBIC *vs* BBRv2 en fonction des capacités de *buffers* en paquets pour un OWD variable et un BDP modélisé à un OWD = 143ms, à RTT égaux

α	0.05		0.1		0.2		0.5		1		2	
Capacité des *Buffers*	59		118		236		590		1180		2360	
	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam
Débit CUBIC (Mbit/s)	17.4	9.3	22.2	14.4	26.8	18.8	30.2	33.7	33.9	33.3	36.2	31.0
Débit BBRv2 (Mbit/s)	16.7	22.0	16.5	20.1	15.2	17.9	12.7	7.1	10.4	9.0	7.5	11.6
Débit total (Mbit/s)	24.1	31.3	38.7	34.5	42.0	36.7	43.0	40.8	44.29	42.3	43.7	42.7

TABLE 6.7 Valeurs des débits de CUBIC *vs* BBRv2 en fonction des capacités de *buffers* en paquets pour un OWD variable et un BDP modélisé à un OWD = 25ms, à RTT égaux

α	0.05		0.1		0.2		0.5		1		2	
Capacité des *Buffers*	11		21		42		104		207		414	
	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam	No seam	Seam
Débit CUBIC (Mbit/s)	3.57	1.81	8.59	4.06	16.55	8.09	24.61	12.06	29.39	16.47	30.89	23.7
Débit BBRv2 (Mbit/s)	3.2	5.41	5.12	10.35	11.08	19.56	14.68	24.16	12.84	21.05	11.59	18.24
Débit total (Mbit/s)	6.77	7.22	13.71	14.41	27.63	27.65	39.29	36.22	42.23	37.52	42.48	41.94

CHAPITRE 7

CONCLUSIONS ET PERSPECTIVES

Conclusions

Les constellations de satellites ont regagné un intérêt significatif ces dernières années avec un engouement notable et de nouveaux acteurs qui ont eu pour ambition d'améliorer la connectivité à l'Internet en particulier dans les régions difficiles d'accès.

Dans cette thèse, nous avons abordé les différentes caractéristiques du contexte satellite partant de l'orbitographie en passant par les applications et enfin l'impact des caractéristiques inhérentes des constellations de satellites sur la variation de délais engendrée. Nous nous sommes penchés sur les problèmes généraux puis spécifiques de TCP. Nous avons retenu CUBIC et BBR en tant qu'algorithmes de contrôle de congestion à étudier plus spécifiquement car, à eux deux ils représentent la moitié du trafic Internet.

Nous avons également choisi une constellation de satellites de référence qui présente l'avantage d'être représentative et qui a été déployée en vraie grandeur ce qui permet d'en connaître les principaux paramètres.

Nous avons alors défini notre environnement système et ses spécificités, et ce que cela peut engendrer en termes de variations de délai. Par ailleurs, nous avons choisi les outils d'évaluation par simulation (NS-2) et émulation (OPENSAND).

Quatre sources de variations de délai ont été étudiées : la variation d'élévation, le délai du *handover* intra-orbital, le délai du *handover* inter-orbital et le délai dû au *seam handover*. Ils ont des impacts différents sur une connexion CUBIC. Le *handover* en élévation se produit en continu, tandis que pour les terminaux fixes, les *handovers* intra-orbitaux se produisent environ toutes les 10 mn, les *handovers* inter-orbitaux toutes les 2h et les *handovers* dus à la couture se produisent au maximum trois fois et au minimum deux fois sur 24 heures, dans le cas étudié. La couture a engendré une augmentation significative dans les durées de transfert des fichiers de petites inférieures à 100kB, alors que pour les grandes tailles de fichiers, l'impact est marginal. Le délai de variation de l'élévation n'a pas d'impact sur la communication. Néanmoins, dans notre cas de figure, le *handover* de l'élévation à l'intra-orbital entraîne une diminution du débit instantané pour une configuration à haut débit. Le transfert intra-orbital vers l'élévation entraîne, également, une diminution du débit instantané

pour une configuration à bas ou haut débit. Le *handover* de l'élévation à l'inter-orbital peut également s'accompagner d'une chute des performances. Le *handover* de l'inter-orbital à l'élévation se traduit enfin par une diminution du débit instantané. À haut débit, l'impact des différentes variations de délai est généralement plus faible. Néanmoins, l'impact sur la variation du débit n'est peut-être pas colossal. Cela dépendra en particulier de la sensibilité de l'application vis-à-vis de ces variations de débit et de la durée de la communication.

Ensuite, nous nous sommes concentrés sur l'impact de la dynamique de la topologie de la constellation sur CUBIC et BBRv1 du point de vue de la variation de délai. Nous avons montré que CUBIC, fondé sur les pertes, peut souffrir de ces variations de délai, en particulier pour les scénarios à haut débit. D'autre part, BBRv1, qui utilise un algorithme hybride, semble être plus efficace. Ses performances ne sont pas profondément affectées par les variations de délai et il est assez réactif en dépit des délais et de leurs variations de bout-en-bout.

Par ailleurs, ces résultats reposent sur la version v1 de BBR qui a été critiquée pour son manque d'équité. Les évaluations suivantes prennent en compte la nouvelle version de BBR qui est censée être plus équitable que BBRv1. Nous avons abordé la question de l'équité de manière plus générale, car nous avons observé que différentes variantes de TCP, qui sont encore largement déployées, peuvent présenter des performances dissemblables.

Nous nous sommes donc focalisés sur l'équité intra- et inter-protocolaires entre CUBIC et BBRv2.

Pour le partage inter-protocolaire, CUBIC *vs* BBRv2, CUBIC (*loss-based*) domine la bande dans la plupart des cas. À RTT égaux, l'OWD est important et la capacité du *buffer* assez petite, c'est BBRv2 qui domine. Cela soulève des questions quant aux améliorations dont BBRv2 a fait l'objet, dans la mesure où les problèmes persistent. Pour le partage intra-protocolaire, à RTT identiques, une équité existe entre flux BBRv2 ou CUBIC. Si les RTT diffèrent avec des protocoles différents ou BBRv2 tout seul, le flux dont le RTT est le plus important l'emporte et ce quelque soit le cas de figure. A contrario pour CUBIC seul, à RTT différents, le flux dont le RTT est le plus petit l'emporte.

Perspectives

Parmi les pistes que nous envisageons, il pourrait être intéressant d'effectuer des expérimentations avec un trafic plus représentatif, incluant notamment des flux courts et un nombre de flux plus important. C'est particulièrement important d'avoir une étude d'équité avec plus de flux. Nous nous sommes limités à 2 flux, ce qui nous permet une analyse fine et une interprétation du comportement. À plus large échelle, cela permettrait d'obtenir une meilleure quantification des phénomènes observés.

En outre, cela serait pertinent de passer en vraie grandeur les expérimentations sur des liens satellites et des constellations de satellites, par exemple un vrai accès *Iridium NEXT*. Ainsi, nous pourrons confirmer les tendances observées dans un contexte de simulation et d'émulation. Il peut être également intéressant d'ajouter des éléments de considération d'une gestion des classes de service plus réaliste. Si certains flux se voient accorder l'accès en priorité parce qu'appartenant à une classe de service prioritaire, cela aura un impact sur l'équité.

De surcroît, la nouvelle version BBRv2 ouvre par ailleurs de nombreuses perspectives de recherche.

En effet, BBRv2 peut utiliser le marquage par ECN du type DCTCP/L4S [102], [107], [112], tandis que par exemple CUBIC utilise lui plutôt un marquage ECN standard RFC3168 [159]. Il peut être intéressant d'évaluer l'impact du marquage ECN sur les analyses d'équité entre BBRv2 et CUBIC, en particulier leur réaction à un marquage ECN différent. La communauté IETF s'intéresse activement à l'étude d'un cadre appelé DualQ Coupled AQM [162] qui permet la coexistence d'algorithmes nécessitant un marquage de type DCTCP et ceux nécessitant un marquage ECN de type RFC3168 [107]. Cela permettrait d'avoir une étude d'équité plus aboutie et de fournir un délai de file d'attente faible pour les trafics compatibles ECN tout en étant équitables en débit avec les trafics non-ECN.

Par ailleurs, il serait également pertinent d'utiliser une politique AQM plus complexe que le *Drop Tail*, et de voir l'impact que cela peut avoir sur l'équité intra- et inter-protocolaires entre les différents types d'algorithmes.

D'autre part, nous pouvons envisager de mettre en jeu d'autres algorithmes dans un contexte satellite, notamment QUIC. Ce dernier a pris de l'ampleur durant cette thèse. Il serait intéressant de mener des études similaires de performance quand CUBIC ou BBR sont exploités par QUIC. QUIC a fait ses preuves depuis sa conception (annoncé publiquement) en 2013 par *Google*, son implantation sur plusieurs services en 2016 et en 2019 en tant que RFC [163] (parution en 2021). Il sous-tend la majorité des applications actuelles, la philosophie de QUIC est au diapason avec les limitations de TCP relevées par la communauté, allant du multiplexage des connexions, au contrôle de congestion aux extrémités, au 0-RTT-*like*, à l'accusé de réception optimisé, au chiffrement niveau paquet, jusqu'au *network-switch* et à la migration des connexions. Les principales innovations de QUIC résident en l'exploitation d'un mécanisme en espace utilisateur qui permet de déploiements rapides de nouvelles fonctionnalités et d'une agrégation des fonctions avant partagées entre la couche applicative et la couche transport (notamment la sécurité). La communauté s'est penchée sur l'évaluation des performances de QUIC dans les satellites. Dans [164] ils ont évalué plutôt un lien GEO et remarqué que l'établissement rapide de la connexion par Google QUIC (GQUIC) ne compense pas un contrôle de congestion inapproprié. Plusieurs projets ont été entrepris par l'European Space Agency (ESA) [165] en 2020 pour évaluer les performances de QUIC, proposer des axes d'amélioration et contribuer à la normalisation de QUIC. Comme la normalisation de QUIC n'est pas finie, plusieurs *drafts* de l'IETF sont en cours pour l'étude de ses performances dans un contexte satellite avec des BDP élevés [166]. Cependant, l'utilisation et l'intégration de QUIC dans les constellations de satellites ne sont pas discutées à notre connaissance. Même dans le tout dernier *draft* [167], la question reste en suspens.

D'autre part, les méga-constellations de satellites commencent à faire leur apparition. Il serait intéressant d'étudier leurs différentes caractéristiques et l'impact qu'elles pourraient avoir sur le comportement des protocoles considérés en raison du grand nombre de satellites et de la plus forte variabilité potentielle de délai.

PUBLICATIONS

Conférence internationale avec comité de relecture

A. BOUBAKER, E. CHAPUT, N. KUHN et al., « On the Impact of Intrinsic Delay Variation Sources on Iridium LEO Constellation », in *International Conference on Wireless and Satellite Systems (WiSATS)*, Springer, sept. 2020, p. 206-226. DOI : 10.1007/978-3-030-69069-4_18

BIBLIOGRAPHIE

[1] T. R. HENDERSON et R. H. KATZ, « Transport Protocols for Internet-Compatible Satellite Networks », *IEEE Journal on Selected Areas in Communications*, t. 17, n° 2, p. 326-344, fév. 1999. DOI : 10.1109/49.748815.

[2] I. F. AKYILDIZ, G. MORABITO et S. PALAZZO, « TCP-Peach: a New Congestion Control Scheme for Satellite IP Networks », *IEEE/ACM Transactions on Networking*, t. 9, n° 3, p. 307-321, juin 2001. DOI : 10.1109/90.929853.

[3] D. KATABI, M. HANDLEY et C. ROHRS, « Internet Congestion Control for Future High Bandwidth-Delay Product Environments », in *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, ACM, t. 2, oct. 2002, p. 89-102. DOI : 10.1145/964725.633035.

[4] S. FU, M. ATIQUZZAMAN et W. IVANCIC, « SCTP over Satellite Networks », in *International Conference on Ion Implantation Technology (IIT)*, IEEE, oct. 2003, p. 112-116. DOI : 10.1109/CCW.2003.1240798.

[5] K. ZHOU, K. L. YEUNG et V. O. LI, « P-XCP: a Transport Layer Protocol for Satellite IP Networks », in *IEEE Global Telecommunications Conference (GLOBECOM)*, IEEE, t. 5, déc. 2004, p. 2707-2711. DOI : 10.1109/GLOCOM.2004.1378847.

[6] ITU, *Sir Arthur C. Clarke — Un Visionnaire de l'Ère Spatiale*, https://www.itu.int/itunews/manager/display.asp?lang=fr&year=2008&issue=03&ipage=Arthur-Clarke&ext=html, 2021.

[7] D. BHATTACHERJEE, W. AQEEL, I. N. BOZKURT et al., « Gearing up for the 21st Century Space Race », in *ACM Workshop on Hot Topics in Networks (HotNets)*, IEEE, nov. 2018, p. 113-119. DOI : 10.1145/3286062.3286079.

[8] J. GRINER, D. D. GLOVER, S. DAWKINS et al., *Ongoing TCP Research Related to Satellites*, RFC 2760, fév. 2000. DOI : 10.17487/RFC2760. adresse : https://www.rfc-editor.org/info/rfc2760.

[9] I. F. AKYILDIZ, G. MORABITO et S. PALAZZO, « Research Issues for Transport Protocols in Satellite IP Networks », *IEEE Wireless Communications*, t. 8, n° 3, p. 44-48, juin 2001. DOI : 10.1109/98.930096.

[10] L. WOOD, « Internetworking with Satellite Constellations », thèse de doct., University of Surrey, Guildford, juin 2001.

[11] L. WOOD, G. PAVLOU et B. EVANS, « Effects on TCP of Routing Strategies in Satellite Constellations », *IEEE Communications Magazine*, t. 39, n° 3, p. 172-181, mars 2001. DOI : 10.1109/35.910605.

[12] Y. CHOTIKAPONG, H. CRUICKSHANK et Z. SUN, « Evaluation of TCP and Internet Traffic via Low Earth Orbit Satellites », *IEEE Wireless Communications*, t. 8, n° 3, p. 28-34, juin 2001. DOI : 10.1109/98.930094.

[13] R. C. DURST, G. J. MILLER et E. J. TRAVIS, « TCP Extensions for Space Communications », *Wireless Networks*, t. 3, n° 5, p. 389-403, nov. 1997. DOI : 10.1145/236387.236398.

[14] L. A. SANCHEZ, M. ALLMAN et D. D. GLOVER, *Enhancing TCP Over Satellite Channels using Standard Mechanisms*, RFC 2488, jan. 1999. DOI : 10.17487/RFC2488. adresse : https://www.rfc-editor.org/info/rfc2488.

[15] C. CAINI, R. FIRRINCIELI, M. MARCHESE et al., « Transport Layer Protocols and Architectures for Satellite Networks », *International Journal of Satellite Communications and Networking*, t. 25, n° 1, p. 1-26, jan. 2007. DOI : 10.1002/sat.855.

[16] S. SUBRAMANIAN, S. SIVAKUMAR, W. J. PHILLIPS et W. ROBERTSON, « Investigating TCP Performance Issues in Satellite Networks », in *Communication Networks and Services Research Conference (CNSR)*, IEEE, mai 2005, p. 327-332. DOI : 10.1109/CNSR.2005.37.

[17] G. NEGLIA, V. MANCUSO, F. SAITTA et I. TINNIRELLO, « A Simulation Study of TCP Performance Over Satellite Channels », *Committee on Space Research (COSPAR) Scientific Assembly*, oct. 2002.

[18] M. ZHANG, M. DUSI, W. JOHN et C. CHEN, « Analysis of UDP Traffic Usage on Internet Backbone Links », in *International Symposium on Applications and the Internet (SAINT)*, IEEE, juill. 2009, p. 280-281. DOI : 10.1109/SAINT.2009.65.

[19] D. LEE, B. E. CARPENTER et N. BROWNLEE, « Observations of UDP to TCP Ratio and Port Numbers », in *International Conference on Internet Monitoring and Protection (ICIMP)*, IEEE, jan. 2010, p. 99-104. DOI : 10.1109/ICIMP.2010.20.

[20] SANDVINE, *Rapport sur l'Internet durant le COVID-19*, https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2020/Phenomena/COVID%20Internet%20Phenomena%20Report%2020200507.pdf, 2021.

[21] WIKIPÉDIA, *Protocole HTTP*, https://fr.wikipedia.org/wiki/Transmission_Control_Protocol, 2021.

[22] CHROMIUM, *QUIC : Protocole de Transport de HTTP*, https://www.chromium.org/quic, 2021.

[23] A. MISHRA, X. SUN, A. JAIN, S. PANDE, R. JOSHI et B. LEONG, « The Great Internet TCP Congestion Control Census », *the ACM on Measurement and Analysis of Computing Systems*, t. 3, n° 3, p. 1-24, déc. 2019. DOI : doi.org/10.1145/3366693.

[24] GOOGLE, *Le Protocole de Transport de Youtube*, https://groups.google.com/g/bbr-dev/c/EipEH0Bq8J0?pli=1, 2021.

[25] SANDVINE, *Rapport sur l'Internet Mobile*, https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2021/Phenomena/MIPR%20Q1%202021%2020210510.pdf, 2021.

[26] FACEBOOK, *Les Protocoles de Transport de Facebook, Instagram, Chrome et Safari*, https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/, 2021.

[27] G. MARAL, M. BOUSQUET et Z. SUN, *Satellite Communications Systems: Systems, Techniques and Technology*. Wiley Online Library, fév. 2020. DOI : 10.1002/9781119673811.

[28] L. WOOD, « Appendix A: Satellite Constellation Design for Network Interconnection Using Non-Geo Satellites », in *Service Efficient Network Interconnection via Satellite*, Wiley Online Library, jan. 2002, p. 215-237. DOI : 10.1002/0470845929.

[29] Y. P. COUBLE, « Optimisation de la Gestion des Ressources Voie Retour », thèse de doct., INPT, Toulouse, sept. 2018.

[30] WIKIPÉDIA, *Orbite HEO*, https://en.wikipedia.org/wiki/Highly_elliptical_orbit, 2022.

[31] ESA, *Orbites*, https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits, 2021.

[32] WIKIPÉDIA, *Constellation de Satellites*, https://fr.wikipedia.org/wiki/Constellation_de_satellites, 2021.

[33] M. LUGLIO et W. PIETRONI, « Optimization of Double-Link Transmission in Case of Hybrid Orbit Satellite Constellations », *Journal of Spacecraft and Rockets (JSR)*, t. 39, n° 5, p. 761-770, mai 2002. DOI : 10.2514/2.3876.

[34] M. ASVIAL, R. TAFAZOLLI et B. G. EVANS, « Genetic Hybrid Satellite Constellation Design », in *International Communications Satellite Systems Conference and Exhibit (ICSSC)*, AIAA, juin 2003, p. 2283. DOI : 10.2514/6.2003-2283.

[35] WIKIPÉDIA, *Constellation de Satellites avec des Orbites différentes*, https://o3bmpower.ses.com/sites/o3bmpower/files/2021-02/SES_O3bmPOWER_LEO_MEO_GEO_guide.pdf, 2021.

[36] A. FERREIRA, J. GALTIER et P. PENNA, « Topological Design, Routing and Handover in Satellite Networks », *Handbook of Wireless Networks and Mobile Computing*, t. 473, p. 493, fév. 2002. DOI : 10.1002/0471224561.ch22.

[37] F. RINALDI, H.-L. MAATTANEN, J. TORSNER et al., « Non-Terrestrial Networks in 5G & Beyond: A Survey », *IEEE Access*, t. 8, p. 165 178-165 200, sept. 2020. DOI : 10.1109/ACCESS.2020.3022981.

[38] M. BREILING, W. ZIA, Y. S. de la FUENTE et al., « LTE Backhauling over MEO-satellites », in *Advanced Satellite Multimedia Systems Conference and the Signal Processing for Space Communications Workshop (ASMS/SPSC)*, IEEE, oct. 2014, p. 174-181. DOI : 10.1109/ASMS-SPSC.2014.6934541.

[39] A. GUIDOTTI, A. VANELLI-CORALLI, M. CAUS et al., « Satellite-Enabled LTE Systems in LEO Constellations », in *IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, juill. 2017, p. 876-881. DOI : 10.1109/ICCW.2017.7962769.

[40] X. LIN, S. ROMMER, S. EULER, E. A. YAVUZ et R. S. KARLSSON, « 5G from Space: An Overview of 3GPP Non-Terrestrial Networks », *IEEE Communications Standards Magazine*, t. 5, n° 4, p. 147-153, oct. 2021. DOI : 10.1109/MCOMSTD.011.2100038.

[41] M. GIORDANI et M. ZORZI, « Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities », *IEEE Network*, t. 35, n° 2, p. 244-251, déc. 2020. DOI : 10.1109/MNET.011.2000493.

[42] K. SAMDANIS et T. TALEB, « The Road beyond 5G: A Vision and Insight of the Key Technologies », *IEEE Network*, t. 34, n° 2, p. 135-141, fév. 2020. DOI : 10.1109/MNET.001.1900228.

[43] S. LIU, Z. GAO, Y. WU et al., « LEO Satellite Constellations for 5G and Beyond: How Will They Reshape Vertical Domains? », *IEEE Communications Magazine*, t. 59, n° 7, p. 30-36, juill. 2021. DOI : 10.1109/MCOM.001.2001081.

[44] M. GIORDANI, M. POLESE, M. MEZZAVILLA, S. RANGAN et M. ZORZI, « Toward 6G Networks: Use Cases and Technologies », *IEEE Communications Magazine*, t. 58, n° 3, p. 55-61, mars 2020. DOI : 10.1109/MCOM.001.1900411.

[45] Z. QU, G. ZHANG, H. CAO et J. XIE, « LEO Satellite Constellation for Internet of Things », *IEEE Access*, t. 5, p. 18 391-18 401, août 2017. DOI : 10.1109/ACCESS.2017.2735988.

[46] V. DESLANDES et D. FERNANDEZ, « MUSTANG: The Ultimate Integrated System for IoT & M2M Communications », in *Advanced Satellite Multimedia Systems Conference and Signal Processing for Space Communications Workshop (ASMS/SPSC)*, IEEE, sept. 2016, p. 1-6.

[47] S. CLUZEL, « Système M2M/IoT par Satellite pour l'Hybridation d'un Réseau NB-IoT via une Constellation LEO », thèse de doct., ISAE, Toulouse, mars 2019.

[48] Z. ZHANG, W. ZHANG et F.-H. TSENG, « Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques », *IEEE Network*, t. 33, n° 1, p. 70-76, jan. 2019. DOI : 10.1109/MNET.2018.1800172.

[49] M. ILCHENKO, T. NARYTNIK, V. PRISYAZHNY, S. KAPSHTYK et S. MATVIENKO, « The Solution of the Problem of the Delay Determination in the Information Transmission and Processing in the LEO Satellite Internet of Things System », in *IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, IEEE, avr. 2019, p. 419-425. DOI : 10.1109/PICST47496.2019.9061350.

[50] MICROSOFT, *Microsoft Azure Space*, https://news.microsoft.com/transform/azure-space-partners-bring-deep-expertise-to-new-venture/, 2021.

[51] PWC, *GSaaS*, https://www.pwc.fr/fr/assets/files/pdf/2020/11/en-france-pwc-space-practice-research-paper-gsaas.pdf, 2021.

[52] AWS, *AWS Ground Station*, https://docs.aws.amazon.com/whitepapers/latest/aws-overview/satellite.html, 2021.

[53] SPACENEWS, *LEOcloud Cloud Computing*, https://spacenews.com/introducing-leocloud/, 2021.

[54] SPACEBELT, *Constellation Cloud de Spacebelt*, https://spacebelt.com/, 2021.

[55] M. COMPANY, *Rapport sur les Constellations de Satellites LEO 2020*, https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/large-leo-satellite-constellations-will-it-be-different-this-time, 2021.

[56] T. BUTASH, P. GARLAND et B. EVANS, « Non-Geostationary Satellite Orbit Communications Satellite Constellations History », *International Journal of Satellite Communications and Networking*, t. 39, n° 1, p. 1-5, août 2020. DOI : 10.1002/sat.1375.

[57] T. R. HENDERSON et L. WOOD, *NS-2 Satellite Plot Scripts*, https://savi.sourceforge.io/sat-plot-scripts/, 2000.

[58] FCC, *Constellations NGSO*, https://www.fcc.gov/document/cut-established-additional-ngso-satellite-applications, 2017.

[59] ——, *Iridium NEXT*, http://licensing.fcc.gov/myibfs/download.do?attachment_key=1031348, 2021.

[60] SES, *O3B*, https://www.ses.com/our-coverage/o3b-meo, 2021.

[61] SLIDESHARE, *Globalstar 2ème Génération*, https://fr.slideshare.net/SambitShreeman/iridium-globalstar-ico-satellite-system, 2018.

[62] WIKIPÉDIA, *Globalstar 2ème Génération*, https://en.wikipedia.org/wiki/Globalstar, 2018.

[63] SKYROCKET, *Globalstar 1ère Génération*, http://space.skyrocket.de/doc_sdat/globalstar-2.htm, 2018.

[64] SES, *O3Bm*, https://www.ses.com/networks/o3b-mpower, 2021.

[65] FCC, *OneWeb*, https://licensing.fcc.gov/myibfs/download.do?attachment_key=1134939, 2021.

[66] ——, *OneWeb*, https://transition.fcc.gov/Daily_Releases/Daily_Business/2017/db0601/DOC-345159A1.pdf, 2021.

[67] ——, *OneWeb*, https://docs.fcc.gov/public/attachments/FCC-20-117A1.pdf, 2021.

[68] ——, *Kuiper*, https://www.fcc.gov/document/fcc-authorizes-kuiper-satellite-constellation, 2021.

[69] ——, *Starlink*, https://licensing.fcc.gov/myibfs/download.do?attachment_key=1190019, 2021.

[70] ——, *LeoSat*, http://licensing.fcc.gov/myibfs/download.do?attachment_key=1158225, 2021.

[71] ——, *Theia*, https://www.fcc.gov/document/fcc-authorizes-theias-satellite-constellation-0, 2021.

[72] ——, *Telesat*, https://www.telesat.com/leo-satellites/, 2021.

[73] H. YI, W. LEI, F. WENJU et al., « LEO Navigation Augmentation Constellation Design with the Multi-Objective Optimization Approaches », *Chinese Journal of Aeronautics*, t. 34, n° 4, p. 265-278, avr. 2021. DOI : 10.1016/j.cja.2020.09.005.

[74] R. GOYAL, S. KOTA, R. JAIN, S. FAHMY, B. VANDALORE et J. KALLAUS, « Analysis and Simulation of Delay and Buffer Requirements of Satellite-ATM Networks for TCP/IP Traffic », *arXiv preprint cs/9809052*, sept. 1998.

[75] L. KLEINROCK, « Power and Deterministic Rules of Thumb for Probabilistic Problems in Computer Communications », in *International Conference on Communications (ICC)*, t. 3, jan. 1979, p. 43.1.1-43.1.10.

[76] R. SALLANTIN, « Optimisation de Bout-en-Bout du Démarrage des Connexions TCP », thèse de doct., INPT, Toulouse, sept. 2014.

[77] D. C. PARTRIDGE, M. ALLMAN et S. FLOYD, *Increasing TCP's Initial Window*, RFC 3390, nov. 2002. DOI : 10.17487/RFC3390. adresse : https://www.rfc-editor.org/info/rfc3390.

[78] J. CHU, N. DUKKIPATI, Y. CHENG et M. MATHIS, *Increasing TCP's Initial Window*, RFC 6928, avr. 2013. DOI : 10.17487/RFC6928. adresse : https://www.rfc-editor.org/info/rfc6928.

[79] D. V. PAXSON et M. ALLMAN, *Computing TCP's Retransmission Timer*, RFC 2988, nov. 2000. DOI : 10.17487/RFC2988. adresse : https://www.rfc-editor.org/info/rfc2988.

[80] *Transmission Control Protocol*, RFC 793, sept. 1981. DOI : 10.17487/RFC0793. adresse : https://rfc-editor.org/rfc/rfc793.txt.

[81] E. BLANTON, D. V. PAXSON et M. ALLMAN, *TCP Congestion Control*, RFC 5681, sept. 2009. DOI : 10.17487/RFC5681. adresse : https://rfc-editor.org/rfc/rfc5681.txt.

[82] A. GURTOV, T. HENDERSON, S. FLOYD et Y. NISHIDA, *The NewReno Modification to TCP's Fast Recovery Algorithm*, RFC 6582, avr. 2012. DOI : 10.17487/RFC6582. adresse : https://rfc-editor.org/rfc/rfc6582.txt.

[83] M. POLESE, F. CHIARIOTTI, E. BONETTO, F. RIGOTTO, A. ZANELLA et M. ZORZI, « A Survey on Recent Advances in Transport Layer Protocols », *IEEE Communications Surveys & Tutorials*, t. 21, n° 4, p. 3584-3608, nov. 2019. DOI : 10.1109/COMST.2019.2932905.

[84] WIKIPÉDIA, *Reno et New Reno*, https://fr.wikipedia.org/wiki/Algorithme_TCP, 2021.

[85] S.-u. LAR et X. LIAO, « An Initiative for a Classified Bibliography on TCP/IP Congestion Control », *Journal of Network and Computer Applications*, t. 36, n° 1, p. 126-133, jan. 2013. DOI : 10.1016/j.jnca.2012.04.003.

[86] L. XU, K. HARFOUSH et I. RHEE, « Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks », in *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, t. 4, nov. 2004, p. 2514-2524. DOI : 10.1109/INFCOM.2004.1354672.

[87] S. HA, I. RHEE et L. XU, « CUBIC: a New TCP-Friendly High-Speed TCP Variant », *ACM SIGOPS Operating Systems Review*, t. 42, n° 5, p. 64-74, juill. 2008. DOI : 10.1145/1400097.1400105.

[88] J. SING et B. SOH, « TCP New Vegas: Performance Evaluation and Validation », in *IEEE Symposium on Computers and Communications (ISCC)*, IEEE, juin 2006, p. 541-546. DOI : 10.1109/ISCC.2006.159.

[89] S. SHALUNOV, G. HAZEL, J. IYENGAR et M. KÜHLEWIND, *Low Extra Delay Background Transport (LEDBAT)*, RFC 6817, déc. 2012. DOI : 10.17487/RFC6817. adresse : https://www.rfc-editor.org/info/rfc6817.

[90] S. MASCOLO, C. CASETTI, M. GERLA, M. Y. SANADIDI et R. WANG, « TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links », in *International Conference On Mobile Computing And Networking (MobiCom)*, ACM, juill. 2001, p. 287-297. DOI : 10.1145/381677.381704.

[91] K. WINSTEIN, A. SIVARAMAN et H. BALAKRISHNAN, « Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks », in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, USENIX Association, avr. 2013, p. 459-471. DOI : 10.5555/2482626.2482670.

[92] K. TAN, J. SONG, Q. ZHANG et M. SRIDHARAN, « A Compound TCP Approach for High-Speed and Long Distance Networks », in *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, avr. 2006. DOI : 10.1109/INFOCOM.2006.188.

[93] N. CARDWELL, Y. CHENG, C. S. GUNN, S. H. YEGANEH et V. JACOBSON, « BBR: Congestion-Based Congestion Control », *Communications of ACM*, t. 60, n° 2, p. 58-66, fév. 2017. DOI : 10.1145/3009824.

[94] K. WINSTEIN et H. BALAKRISHNAN, « TCP ex Machina: Computer-Generated Congestion Control », *SIGCOMM Computer Communication Review*, t. 43, n° 4, p. 123-134, oct. 2013. DOI : 10.1145/2534169.2486020.

[95] M. ŠOŠIĆ et V. STOJANOVIĆ, « Resolving Poor TCP performance on High-Speed Long Distance Links—Overview and Comparison of BIC, CUBIC and Hybla », in *IEEE International Symposium on Intelligent Systems and Informatics (SISY)*, IEEE, sept. 2013, p. 325-330. DOI : 10.1109/SISY.2013.6662595.

[96] M. AHMAD, A. B. NGADI, A. NAWAZ, U. AHMAD, T. MUSTAFA et A. RAZA, « A Survey on TCP CUBIC Variant Regarding Performance », in *International Multitopic Conference (INMIC)*, IEEE, déc. 2012, p. 409-412. DOI : 10.1109/INMIC.2012.6511454.

[97] I. RHEE, L. XU, S. HA, A. ZIMMERMANN, L. EGGERT et R. SCHEFFENEGGER, *CUBIC for Fast Long-Distance Networks*, RFC 8312, fév. 2018. DOI : 10.17487/RFC8312. adresse : https://rfc-editor.org/rfc/rfc8312.txt.

[98] L. S. BRAKMO, S. W. O'MALLEY et L. L. PETERSON, « TCP Vegas: New techniques for Congestion Detection and Avoidance », in *Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, ACM, oct. 1994, p. 24-35. DOI : 10.1145/190809.190317.

[99] L. S. BRAKMO et L. L. PETERSON, « TCP Vegas: End-to-End Congestion Avoidance on a Global Internet », *IEEE Journal on Selected Areas in Communications*, t. 13, n° 8, p. 1465-1480, oct. 1995. DOI : 10.1109/49.464716.

[100] S. MASCOLO, L. A. GRIECO, R. FERORELLI, P. CAMARDA et G. PISCITELLI, « Performance Evaluation of Westwood+ TCP Congestion Control », *Performance Evaluation*, t. 55, n° 1-2, p. 93-111, jan. 2004. DOI : 10.1016/S0166-5316(03)00098-1.

[101] N. CARDWELL, Y. CHENG, S. H. YEGANEH et V. JACOBSON, « BBR Congestion Control », IETF, Internet-Draft draft-cardwell-iccrg-bbr-congestion-control-00, juill. 2017, Work in Progress, 34 p. adresse : https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-00.

[102] N. CARDWELL, Y. CHENG, S. H. YEGANEH et al., « BBRv2: A Model-Based Congestion Control », in *Presentation in Internet Congestion Control (ICCRG) at IETF 104th meeting*, IETF Meeting Proceedings, nov. 2019.

[103] N. CARDWELL, Y. CHENG, C. S. GUNN, S. H. YEGANEH et V. JACOBSON, « Phases de Fonctionnement de BBR », in *Presentation in Internet Congestion Control (ICCRG) at IETF 97th meeting*, IETF Meeting Proceedings, juill. 2016, p. 1-36. adresse : https://www.ietf.org/proceedings/97/slides/slides-97-iccrg-bbr-congestion-control-02.pdf.

[104] D. SCHOLZ, B. JAEGER, L. SCHWAIGHOFER, D. RAUMER, F. GEYER et G. CARLE, « Towards a Deeper Understanding of TCP BBR Congestion Control », in *IFIP Networking Conference and Workshops*, IEEE, mai 2018, p. 1-9. DOI : 10.23919/IFIPNetworking.2018.8696830.

[105] M. HOCK, R. BLESS et M. ZITTERBART, « Experimental evaluation of BBR congestion control », in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, IEEE, 2017, p. 1-10.

[106] K. MIYAZAWA, K. SASAKI, N. ODA et S. YAMAGUCHI, « Cyclic performance fluctuation of TCP BBR », in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, t. 1, 2018, p. 811-812.

[107] A. NANDAGIRI, M. P. TAHILIANI, V. MISRA et K. RAMAKRISHNAN, « BBRvl vs BBRv2: Examining Performance Differences through Experimental Evaluation », in *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*, IEEE, juill. 2020, p. 1-6. DOI : 10.1109/LANMAN49260.2020.9153268.

[108] P. FARROW, « Performance analysis of heterogeneous TCP congestion control environments », in *2017 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, IEEE, 2017, p. 1-6.

[109] N. CARDWELL, Y. CHENG, S. H. YEGANEH, I. SWETT et V. JACOBSON, « BBR Congestion Control », Internet Engineering Task Force, Internet-Draft draft-cardwell-iccrg-bbr-congestion-control-01, nov. 2021, Work in Progress, 66 p. adresse : https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-01.

[110] J. GOMEZ, E. KFOURY, J. CRICHIGNO, E. BOU-HARB et G. SRIVASTAVA, « A Performance Evaluation of TCP BBRv2 Alpha », in *International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, août 2020, p. 309-312. DOI : 10.1109/TSP49548.2020.9163512.

[111] WIKIPÉDIA, *TCP BBRv2 : Évolution de BBRv1*, https://en.wikipedia.org/wiki/TCP_congestion_control, 2021.

[112] N. CARDWELL, Y. CHENG, S. H. YEGANEH et al., « BBR v2: A Model-based Congestion Control », in *Presentation in Internet Congestion Control (ICCRG) at IETF 105th meeting*, IETF Meeting Proceedings, juill. 2019, p. 1-21. adresse : https://datatracker.ietf.org/meeting/105/materials/slides-105-iccrg-bbr-v2-a-model-based-congestion-control-00.

[113] Y.-J. SONG, G.-H. KIM, I. MAHMUD, W.-K. SEO et Y.-Z. CHO, « Understanding of BBRv2: Evaluation and Comparison With BBRv1 Congestion Control Algorithm », *IEEE Access*, t. 9, p. 37 131-37 145, fév. 2021. DOI : 10.1109/ACCESS.2021.3061696.

[114] M. WANG, Y. CUI, X. WANG, S. XIAO et J. JIANG, « Machine Learning for Networking: Workflow, Advances and Opportunities », *IEEE Network*, t. 32, n° 2, p. 92-99, nov. 2017. DOI : 10.1109/MNET.2017.1700200.

[115] M. SCHAPIRA et K. WINSTEIN, « Congestion-Control Throwdown », in *ACM Workshop on Hot Topics in Networks (HotNets)*, IEEE, nov. 2017, p. 122-128. DOI : 10.1145/3152434.3152446.

[116] H. JIANG, Q. LI, Y. JIANG et al., « When Machine Learning meets Congestion Control: A Survey and Comparison », *Computer Networks*, t. 192, p. 1-28, juin 2021. DOI : 10.1016/j.comnet.2021.108033.

[117] M. GERLA, M. LUGLIO, R. KAPOOR, J. STEPANEK, F. VATALARO et M. VÁZQUEZ-CASTRO, « TCP via Satellite Constellations », in *European Workshop on Mobile/Personal Satcoms (EMPS)*, IEEE, sept. 2000, p. 82-91.

[118] C. ROSETI et E. KRISTIANSEN, « TCP Noordwijk: TCP-based Transport Optimized for Web Traffic in Satellite Networks », in *International Communications Satellite Systems Conference (ICSSC)*, juin 2008. DOI : 10.2514/6.2008-5524.

[119] C. CAINI et R. FIRRINCIELI, « TCP Hybla: a TCP Enhancement for Heterogeneous Networks », *International Journal of Satellite Communications and Networking*, t. 22, n° 5, p. 547-566, sept. 2004. DOI : 10.1002/sat.799.

[120] A. ABDELSALAM, M. LUGLIO, C. ROSETI et F. ZAMPOGNARO, « TCP Wave: a New Reliable Transport Approach for Future Internet », *Computer Networks*, t. 112, p. 122-143, jan. 2017. DOI : 10.1016/j.comnet.2016.11.002.

[121] C. CAINI et R. FIRRINCIELI, « End-to-end TCP enhancements performance on satellite links », in *IEEE Symposium on Computers and Communications (ISCC)*, IEEE, juin 2006, p. 1031-1036. DOI : 10.1109/ISCC.2006.66.

[122] B. TAURAN, « On the Interaction between Transport Protocols and Link-Layer Reliability Schemes for Satellite Mobile Services », thèse de doct., ISAE, Toulouse, déc. 2018.

[123] J.-Y. LE BOUDEC, « Rate Adaptation, Congestion Control and Fairness: A Tutorial », *Web page*, t. 4, nov. 2005. adresse : https://moodle.epfl.ch/file.php/523/CC_Tutorial/cc.pdf.

[124] E. ALTMAN, K. AVRACHENKOV et B. PRABHU, « Fairness in MIMD Congestion Control Algorithms », *Telecommunication Systems*, t. 30, n° 4, p. 387-415, août 2005. DOI : 10.1109/INFCOM.2005.1498360.

[125] S. MA, J. JIANG, W. WANG et B. LI, « Fairness of Congestion-Based Congestion Control: Experimental Evaluation and Analysis », *arXiv preprint arXiv:1706.09115*, juill. 2017.

[126] G. APPENZELLER, I. KESLASSY et N. MCKEOWN, « Sizing Router Buffers », *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, t. 34, n° 4, p. 281-292, août 2004. DOI : 10.1145/1015467.1015499.

[127] M. ENACHESCU, Y. GANJALI, A. GOEL, N. MCKEOWN et T. ROUGHGARDEN, « Routers with Very Small Buffers », in *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, avr. 2006, p. 1-11. DOI : 10.1109/INFOCOM.2006.240.

[128] B. SPANG, S. ARSLAN et N. MCKEOWN, « Updating the Theory of Buffer Sizing », *Performance Evaluation*, p. 102 232, nov. 2021. DOI : 10.1016/j.peva.2021.102232.

[129] F. FEJES, G. GOMBOS, S. LAKI et S. NÁDAS, « Who will Save the Internet from the Congestion Control Revolution? », in *the Workshop on Buffer Sizing*, ACM, déc. 2019, p. 1-6. DOI : 10.1145/3375235.3375242.

[130] S. RADHAKRISHNAN, Y. CHENG, J. CHU, A. JAIN et B. RAGHAVAN, « TCP Fast Open », in *the Seventh COnference on emerging Networking EXperiments and Technologies (Co-NEXT)*, ACM, déc. 2011, p. 1-12. DOI : 10.1145/2079296.2079317.

[131] A. FORD, C. RAICIU, M. J. HANDLEY, O. BONAVENTURE et C. PAASCH, *TCP Extensions for Multipath Operation with Multiple Addresses*, RFC 8684, mars 2020. DOI : 10.17487/RFC8684. adresse : https://rfc-editor.org/rfc/rfc8684.txt.

[132] D. A. BORMAN, R. T. BRADEN et V. JACOBSON, *TCP Extensions for High Performance*, RFC 1323, mai 1992. DOI : 10.17487/RFC1323. adresse : https://www.rfc-editor.org/info/rfc1323.

[133] S. IREN, P. D. AMER et P. T. CONRAD, « The Transport Layer: Tutorial and Survey », *ACM Computing Surveys*, t. 31, n° 4, p. 360-404, déc. 1999. DOI : 10.1145/344588.344609.

[134] S. KUMAR et S. RAI, « Survey on Transport Layer Protocols: TCP & UDP », *International Journal of Computer Applications*, t. 46, n° 7, p. 20-25, mai 2012.

[135] M. OULMAHDI, C. CHASSOT et N. VAN WAMBEKE, « Transport Protocols: Limitations, Evolution Ostacles and Solutions for an Actual Deployment in the Internet », *International Journal of Parallel, Emergent and Distributed Systems*, t. 30, n° 6, p. 515-535, déc. 2015. DOI : 10.1080/17445760.2015.1053807.

[136] S. FLOYD, J. MAHDAVI, M. MATHIS et D. A. ROMANOW, *TCP Selective Acknowledgment Options*, RFC 2018, oct. 1996. DOI : 10.17487/RFC2018. adresse : https://www.rfc-editor.org/info/rfc2018.

[137] K. LI, B. LIN et J. MA, « Bit-Error Eate Investigation of Satellite-To-Ground Downlink Optical Communication Employing Spatial Diversity and Modulation Techniques », *Optics Communications*, t. 442, p. 123-131, mars 2019. DOI : 10.1016/j.optcom.2019.03.012.

[138] M. GREGORY, F. HEINE, H. KÄMPFNER, R. LANGE, M. LUTZER et R. MEYER, « Coherent Inter-Satellite and Satellite-Ground Laser Links », in *Free-Space Laser Communication Technologies*, International Society for Optics et Photonics, t. 7923, fév. 2011, p. 13-20. DOI : 10.1117/12.873532.

[139] ITU, *TEB Recommandé par l'ITU*, https://www.itu.int/dms_pubrec/itu-r/rec/s/R-REC-S.2099-0-201612-I!!PDF-E.pdf, 2021.

[140] B. NIEHOEFER, S. ŠUBIK et C. WIETFELD, « The CNI Open Source Satellite Simulator based on OMNeT++ », in *International Conference on Simulation Tools and Techniques (ICST/SimuTools)*, ICST/ACM, jan. 2013, p. 314-321. DOI : 10.4108/simutools.2013.251580.

[141] J. PUTTONEN, B. HERMAN, S. RANTANEN, F. LAAKSO et J. KURJENNIEMI, « Satellite Network Simulator 3 », in *Workshop on Simulation for European Space Programmes (SESP)*, t. 24, mars 2015, p. 26. adresse : https://www.sns3.org/content/references.php.

[142] NSNAM, *Network Simulator 3*, https://www.nsnam.org/, 2021.

[143] S. SIRAJ, A. GUPTA et R. BADGUJAR, « Network Simulation Tools Survey », *International Journal of Advanced Research in Computer and Communication Engineering*, t. 1, n° 4, p. 199-206, juin 2012. DOI : 10.21095/ajmr/2017/v0/i0/122468.

[144] J. PUTTONEN, S. RANTANEN, F. LAAKSO, J. KURJENNIEMI, K. AHO et G. ACAR, « Satellite Model for Network Simulator 3 », in *International Conference on Simulation Tools and Techniques (ICST/SimuTools)*, ICST/ACM, mars 2014, p. 86-91. DOI : 10.4108/icst.simutools.2014.254631.

[145] M. LIU, Y. GUI, J. LI et H. LU, « Large-Scale Small Satellite Network Simulator: Design and Evaluation », in *IEEE International Conference on Hot Information-Centric Networking (HotICN)*, IEEE, déc. 2020, p. 194-199. DOI : 10.1109/HotICN50779.2020.9350838.

[146] B. KEMPTON et A. RIEDL, « Network Simulator for Large Low Earth Orbit Satellite Networks », in *IEEE International Conference on Communications (ICC)*, IEEE, juin 2021, p. 1-6. DOI : 10.1109/ICC42927.2021.9500439.

[147] A. KAYSSI et A. EL-HAJ-MAHMOUD, « EmuNET: a Real-Time Network Emulator », in *ACM Symposium on Applied Computing (SAC)*, ACM, jan. 2004, p. 357-362. DOI : 10.1145/967900.967977.

[148] E. DUBOIS, N. KUHN, J.-B. DUPÉ et al., « OpenSAND, an Open Source SATCOM Emulator », *Ka and Broadband Communications Conference (Ka)*, oct. 2017. adresse : https://opensand.org/content/references.php.

[149] D. F. PIÑAS et C. MORLET, « AINE: An IP Network Emulator », in *International Workshop on Signal Processing for Space Communications (SPSC)*, IEEE, oct. 2008, p. 1-7. DOI : 10.1109/SPSC.2008.4686711.

[150] S. ERL et T. de COLA, « DVB-RCS2/S2 Testbed: A Distributed Testbed for Next-Generation Satellite System Design and Validation », in *Advanced Satellite Multimedia Systems Conference and the Signal Processing for Space Communications Workshop (ASMS/SPSC)*, IEEE, sept. 2014, p. 382-389. DOI : 10.1109/ASMS-SPSC.2014.6934571.

[151] ITRINEGY, *NE-ONE Émulateur d'iTrinegy*, https://itrinegy.com/solution-by-task/satellite-link-simulation/, 2021.

[152] ESA, *SCNE - Satellite Constellation Network Emulator*, https://artes.esa.int/projects/scne, 2021.

[153] ——, *SCNE - Satellite Constellation Network Emulator*, https://artes.esa.int/funding/realtime-satellite-network-emulator-artes-3a082, 2021.

[154] E. PETERSEN, J. LÓPEZ, N. KUSHIK, C. POLETTI et D. ZEGHLACHE, « Satellite Communication Digital Twin for Evaluating Novel Solutions: Dynamic Link Emulation Architecture », *arXiv preprint arXiv:2107.07217*, juill. 2021.

[155] IRIDIUM, *Valeurs des Débits Up-links et Down-links d'Iridium*, https://www.iridium.com/services/iridium-certus/, 2020.

[156] ARIASE, *Valeurs des Débits Up-links et Down-links d'Iridium*, https://blog.ariase.com/box/actualite/internet-satellite-offres-europasat-data-illimitee-aide-etat-150-euros, 2021.

[157] IETF, *Valeurs des Débits pour les SATCOMS*, https://tools.ietf.org/html/draft-kuhn-quic-4-sat-04, 2021.

[158] CLUBIC, *Valeurs des Débits de Telesat*, https://www.clubic.com/connexion-internet/actualite-865352-deserts-numeriques-canada-investira-600-10-ans-internet-satellite.html, 2021.

[159] S. FLOYD, D. K. K. RAMAKRISHNAN et D. L. BLACK, *The Addition of Explicit Congestion Notification (ECN) to IP*, RFC 3168, sept. 2001. DOI : 10.17487/RFC3168. adresse : https://rfc-editor.org/rfc/rfc3168.txt.

[160] S. NÁDAS, B. VARGA, F. FEJES, G. GOMBOS et S. LAKI, *On Congestion Control (un)fairness*, IETF Meeting Proceedings, nov. 2020. adresse : https://www.ietf.org/proceedings/109/slides/slides-109-iccrg-on-congestion-control-compatibility-00.pdf.

[161] T. KOZU, Y. AKIYAMA et S. YAMAGUCHI, « Improving RTT Fairness on CUBIC TCP », in *International Symposium on Computing and Networking (CANDAR)*, IEEE, déc. 2013, p. 162-167. DOI : 10.1109/CANDAR.2013.30.

[162] K. D. SCHEPPER, B. BRISCOE et G. WHITE, « DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S) », IETF, Internet-Draft draft-ietf-tsvwg-aqm-dualq-coupled-20, déc. 2021, Work in Progress, 61 p. adresse : https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-aqm-dualq-coupled-20.

[163] M. THOMSON, *Version-Independent Properties of QUIC*, RFC 8999, mai 2021. DOI : 10.17487/RFC8999. adresse : https://rfc-editor.org/rfc/rfc8999.txt.

[164] L. THOMAS, E. DUBOIS, N. KUHN et E. LOCHIN, « Google QUIC Performance over a Public SATCOM Access », *International Journal of Satellite Communications and Networking*, t. 37, n° 6, p. 601-611, fév. 2019. DOI : 10.1002/sat.1301.

[165] ESA, *Projet ESA : QUIC via Satellite*, https://artes.esa.int/projects/assessment-quic-over-satellite-links, 2021.

[166] N. KUHN, G. FAIRHURST, J. BORDER et S. EMILE, « QUIC for SATCOM », IETF, Internet-Draft draft-kuhn-quic-4-sat-06, oct. 2020, Work in Progress, 15 p. adresse : https://datatracker.ietf.org/doc/html/draft-kuhn-quic-4-sat-06.

[167] T. JONES, G. FAIRHURST, N. KUHN, J. BORDER et S. EMILE, « Enhancing Transport Protocols over Satellite Networks », IETF, Internet-Draft draft-jones-tsvwg-transport-for-satellite-00, fév. 2021, Work in Progress, 27 p. adresse : https://datatracker.ietf.org/doc/html/draft-jones-tsvwg-transport-for-satellite-00.

[168] A. BOUBAKER, E. CHAPUT, N. KUHN et al., « On the Impact of Intrinsic Delay Variation Sources on Iridium LEO Constellation », in *International Conference on Wireless and Satellite Systems (WiSATS)*, Springer, sept. 2020, p. 206-226. DOI : 10.1007/978-3-030-69069-4_18.

Résumé

Les constellations de satellites ont pris un nouvel essor ces dernières années avec des projets particulièrement ambitieux qui ont suffi à raviver la flamme de la communauté de recherche.

Un des problèmes cruciaux est alors l'étude de l'adéquation des protocoles principalement conçus et utilisés dans un contexte terrestre vis-à-vis de ces communications par satellite.

Nous avons étudié les différences entre les piles TCP du passé et celles qui sont actuellement déployées. Nous nous sommes alors intéressés à l'adéquation des algorithmes récents tels que CUBIC et BBR, conçus pour un contexte terrestre, aux réseaux satellite. Nous avons montré que les variations de délais induites par les constellations de satellites pouvaient être globalement bien absorbées par ces nouvelles variantes.

Nous nous sommes enfin intéressés à l'équité entre les flux. Nous avons montré que des iniquités existent dans le cas de flux ayant des caractéristiques de délai et/ou utilisant différentes variantes de TCP y compris avec BBRv2 qui était censé y remédier.

Mots clés : Constellation de Satellites, TCP, Contrôle de Congestion

Abstract

Satellite constellations have taken a new impetus in recent years with even more ambitious future projects. The momentum has lasted long enough to raise the interest of the research community.

One of the crucial issue is addressing the adequacy of the protocols mainly designed and used in a terrestrial context with respect to these satellite communications.

We evaluated the differences between past and currently deployed TCP stacks. We then focused on the suitability of recent algorithms such as CUBIC and BBR, designed for a terrestrial context, to satellite networks. We have shown that the delay variations induced by satellite constellations could be globally well absorbed by these new variants.

Finally, we looked at the fairness between flows. We have shown that unfairness exists in the case of flows having delay characteristics and/or using different variants of TCP including with BBRv2 which was supposed to address it.

Keywords : Satellite Constellations, TCP, Congestion Control