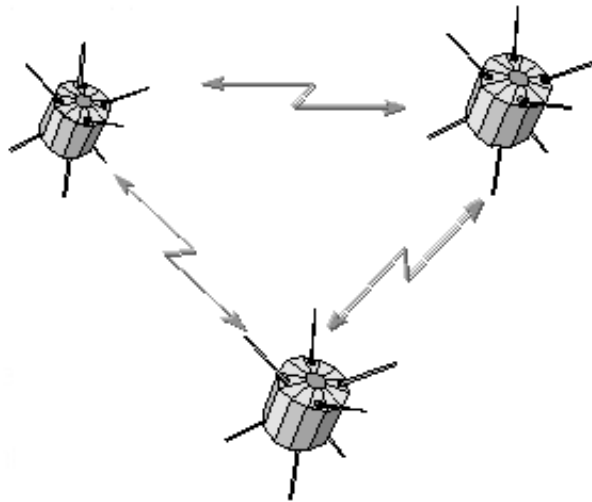


Stage de D.E.A Réseaux
Laboratoire PRiSM

Etude et Evaluation des techniques d'ordonnancement dans un réseau de constellations de satellites



Stagiaire : Erwan ERMEL
Encadrant : Khaled BOUSSETTA
Responsable de stage : Guy PUJOLLE
Septembre 2000

Table des matières

| | |
|-------------------------------------------------------------------------|-----------|
| Remerciements | 9 |
| Introduction | 11 |
| I Projet "Constellation de satellites pour le multimédia" | 13 |
| 1 Les conventions du projet | 15 |
| 1.1 Convention 1 : Aspect Physique | 15 |
| 1.2 Convention 2 : Aspect Réseaux | 15 |
| 1.3 Convention 3 : Les Services | 15 |
| 1.4 Convention 4 : Evaluation | 16 |
| 1.5 Convention 5 : Réalisation du simulateur | 16 |
| 1.6 Les partenaires du projet | 17 |
| 1.7 Notre contribution | 17 |
| II Constellation de Satellites | 19 |
| 2 Constellation de Satellites | 21 |
| 2.1 Description d'un satellite | 21 |
| 2.1.1 L'altitude, variable caractérisant tout un système | 21 |
| 2.1.2 Temps de propagation | 22 |
| 2.1.3 Limitation des traitements | 22 |
| 2.1.4 Partages des fréquences | 23 |
| 2.1.5 Conditions météorologiques | 23 |
| 2.1.6 Routage et Commutation sur satellites | 23 |
| 2.1.7 Les différents types de routages | 24 |
| 2.2 Caractéristiques propres aux constellations de satellites | 24 |
| 2.2.1 Configurations des constellations | 25 |
| 2.2.2 Trajectoires des satellites | 25 |
| 2.2.3 Répartition des satellites le long de ces trajectoires | 25 |
| 2.2.4 Liaisons Intersatellites | 25 |

| | | |
|------------|-------------------------------------------------------------|-----------|
| III | Etat de l'art des techniques d'ordonnancement | 27 |
| 3 | Objectifs des disciplines d'ordonnancement | 31 |
| 3.1 | La facilité d'implémentation | 31 |
| 3.2 | L'équité et la Protection | 32 |
| 3.3 | Les contraintes liées aux performances | 32 |
| 4 | Les choix fondamentaux | 33 |
| 4.1 | Les priorités | 33 |
| 4.2 | Travail conservatif ou non-conservatif | 33 |
| 4.3 | Degré d'agrégation | 33 |
| 4.4 | Agrégation, Ordre et Priorité des services | 34 |
| 5 | Les différentes techniques d'ordonnancement | 35 |
| 5.1 | Le First In First Out (FIFO) | 35 |
| 5.2 | Le Round Robin (RR) | 35 |
| 5.3 | Weighed Round Robin (WRR) | 35 |
| 5.4 | Deficit Round Robin (DRR) | 36 |
| 5.5 | Generalized Processor Sharing (GPS) | 36 |
| 5.6 | Weighed Fair Queuing (WFQ) | 36 |
| 5.7 | Horloge Virtuelle (VC) | 37 |
| 5.8 | "Earliest Due Date" | 38 |
| IV | Comparaison de plusieurs techniques d'ordonnancement | 39 |
| 6 | Network Simulator et modèles de simulation | 41 |
| 6.1 | Network Simulator | 41 |
| 6.2 | Modèle des simulations | 42 |
| 7 | Résultats et Analyses | 45 |
| 7.1 | Les sources | 45 |
| 7.2 | Résultats et Analyses | 46 |
| 7.2.1 | FIFO | 46 |
| 7.2.2 | RED | 47 |
| 7.2.3 | DRR | 48 |
| 7.2.4 | SFQ | 49 |
| | Conclusion | 51 |
| | Bibliographie | 52 |
| A | Trajectoires des satellites | 55 |
| B | Code pour Network Simulator : Terrestre | 57 |
| C | Code pour Network Simulator : Satellites | 61 |

Table des figures

| | | |
|------|------------------------------------------------------------------|----|
| 2.1 | Les types de satellites en fonction de leur altitude | 22 |
| 2.2 | Routage terrestre - Ground Base Routing | 24 |
| 2.3 | Routage via satellites - Satellites Base Routing | 24 |
| 2.4 | Routage "Bent Pipe" | 24 |
| 2.5 | Liens Inter-satellites pour la constellation Iridium | 26 |
| 2.6 | Liens Inter-satellites pour la constellation Télédésic | 26 |
| 6.1 | Scénario | 42 |
| 6.2 | Traffic On-Off | 42 |
| 7.1 | Les sources avec un temps de simulation de 100s | 45 |
| 7.2 | Les sources avec un temps de simulation de 600s | 45 |
| 7.3 | FIFO sur un réseau terrestre | 46 |
| 7.4 | FIFO sur un réseau de satellites | 46 |
| 7.5 | RED sur un réseau terrestre | 47 |
| 7.6 | RED sur un réseau de satellites | 47 |
| 7.7 | DRR sur un réseau terrestre | 48 |
| 7.8 | DRR sur un réseau de satellites | 48 |
| 7.9 | SFQ sur un réseau terrestre | 49 |
| 7.10 | SFQ sur un réseau de satellites | 49 |
| 7.11 | SFQ sur un réseau terrestre | 50 |
| 7.12 | SFQ sur un réseau de satellites | 50 |
| A.1 | Orbite polaire | 55 |
| A.2 | Orbite équatoriale | 55 |
| A.3 | Rossette | 56 |

Liste des tableaux

| | | |
|-----|-------------------------------------------------------|----|
| 2.1 | Types d'orbites | 21 |
| 2.2 | Temps de propagation d'un signal End-To-End | 22 |
| 2.3 | Couverture globale | 22 |
| 2.4 | Nombres d'opérations pouvant être traités | 23 |
| 2.5 | Fréquences utilisées par les satellites | 23 |
| 2.6 | Temps de routage et de commutation | 24 |

Remerciements

Je souhaite remercier les personnes suivantes qui m'ont prodigué leur aide et soutien durant ce stage :

- Khaled BOUSSETTA, mon responsable de stage, qui a su orienter mes recherches vers des objectifs pas toujours simples à mettre en oeuvre mais dont l'étude était intéressante.
- Lila BOUKHATEM, Nadjib ACHIR, Andre-Luc BEYLOT, Marcello DIAS DE AMORIN, Yacine GHAMRI DOUDANE, Mourad GUEROUI membres de l'équipe réseaux du PRiSM, qui m'ont aidé durant ce stage.
- Guy PUJOLLE, pour m'avoir permis d'effectuer ce stage dans le domaine satellitaire au sein de son équipe.

Introduction

Il existe actuellement trois grands types de systèmes de communications globales : le premier est le réseau terrestre à base de câblages, comme par exemple les fibres optiques, les câbles coaxiaux. Il est destiné à fournir des services à des utilisateurs fixes. Le deuxième est le réseau cellulaire terrestre s'appuyant sur une partie du spectre radio. Il est voué à des utilisateurs mobiles. Enfin le troisième système repose sur l'utilisation des satellites, dont le principal intérêt est de donner une liberté et une mobilité accrue aux utilisateurs d'un tel système grâce à une couverture globale de la Terre.

On peut considérer les réseaux satellitaires comme une extension des réseaux cellulaires terrestres mais les contraintes des réseaux satellitaires sont bien plus fortes que celles des réseaux cellulaires terrestres. Le principal problème de ces deux types de réseaux est le spectre radio, car celui-ci est limité. Cette ressource est donc rare. Il faut alors l'utiliser au mieux pour optimiser son utilisation et donc le nombre de communications simultanées.

Les études précédentes sur les réseaux cellulaires terrestres ne peuvent pas être appliquées aux réseaux satellites sans prendre en compte des contraintes liées à l'environnement spatial, comme le temps de propagation, le grand nombre d'utilisateurs potentiels par cellule, les *handover*. Nous possédons plusieurs outils pour gérer au mieux cette ressource rare qu'est le spectre radio : l'acceptation des nouveaux appels est gérée, l'allocation dynamique des canaux, on peut garantir des qualités de services en utilisant des techniques d'ordonnancement au niveau des paquets (MAC).

Il nous a semblé très intéressant de se pencher sur les techniques d'ordonnancement dans un contexte satellitaire, afin de savoir si de telles méthodes sont envisageables dans de tels systèmes. Dans une première partie nous allons vous présenter le projet "Constellation", projet d'études sur les constellations de satellites auquel mon sujet de stage est rattaché. Ensuite je vais faire un rappel sur les principales caractéristiques des réseaux de satellites.. Suivra un état de l'art des techniques d'ordonnancement. Enfin dans la quatrième et dernière partie, les résultats d'une étude comparative de plusieurs techniques d'ordonnancement seront donnés.

Première partie

Projet "Constellation de satellites pour le multimédia"

Chapitre 1

Les conventions du projet

Le projet "Constellation de satellites pour le multimédia" est un projet R.N.R.T. (R N R T) dont le but est de réaliser une plateforme d'évaluation d'une constellation de satellites.

Afin de remplir cet objectif, le projet a été divisé en cinq conventions.

1.1 Convention 1 : Aspect Physique

Cette première convention concerne les problèmes posés par les constellations de satellites du point de vue de la qualité de la transmission sur l'interface radio. En effet, la qualité et les performances qui seront obtenues sur cette interface seront primordiales pour la mise en place des protocoles de communication pour le multimédia. Les connaissances dans ce domaine sont très fragmentaires et des études plus poussées sont absolument nécessaires même si le but direct du projet global est plutôt dirigé vers l'utilisation des constellations de satellites pour le transport d'applications multimédias.

1.2 Convention 2 : Aspect Réseaux

On se préoccupe ici de la gestion des ressources disponibles (fréquence, mémoire, capacité), la gestion des handover, le routage en tenant compte de la dynamique des satellites. Cette gestion des ressources demande une signalisation importante qui sera également étudiée dans cette convention. En effet, les constellations de satellites n'ont qu'une quantité de ressources limitée et l'optimisation en est obligatoire pour atteindre des coûts compétitifs. L'optimisation peut s'effectuer à travers la gestion des handovers et plus globalement la gestion des ressources de communication et de commutation des matériels embarqués.

1.3 Convention 3 : Les Services

Pour effectuer des communications multimédias des protocoles adaptés doivent être mis en place. C'est le but de la convention 3 qui permet d'étudier l'utilisation des protocoles ATM et IP dans le cas de communications avec un système satellitaire, toujours en regardant le

problème de l' utilisation efficace des ressources (taux d'erreur, codage) et dans le but d'obtenir un contrôle de la qualité de service (maintien de la qualité de transport de bout en bout...).

1.4 Convention 4 : Evaluation

La quatrième convention se propose de concevoir et de réaliser un nouvel environnement capable de contrôler la complexité et de maîtriser les paramètres d'une constellation de satellites. Les architectures réseaux deviennent de plus en plus complexes malgré la volonté affichée et répétée de la communauté industrielle et académique d'aller vers une simplification. La maîtrise globale pour un environnement aussi complexe qu'une constellation de satellites est capitale. Le développement d'architectures à base d'agents intelligents semblent être l'une des voies possibles pour arriver à maîtriser cette complexité. Les problèmes de contrôle et de gestion de ces réseaux passeront par ce biais. Nous souhaitons pouvoir allier à ces contrôles des propriétés d'intelligence qui seront capables de prendre en charge la complexité inhérente à ces architectures (contrôle, routage, gestion, garantie de qualité de service, sécurité).

1.5 Convention 5 : Réalisation du simulateur

La cinquième convention concerne le développement d'un atelier contenant un environnement de simulation pour dimensionner, tester et évaluer les performances de systèmes (ou sous-systèmes) de télécommunications basés sur des constellations de satellites. Cet atelier sera mis à disposition des promoteurs de systèmes, industriels et opérateurs. La simulation est une technique largement répandue pour l'optimisation des systèmes de télécommunication. De nombreux outils existent déjà, ils ont été développés afin de répondre aux exigences de projets spécifiques. Ils sont en général partiels, peu évolutifs, et propriété des promoteurs de systèmes. Cet atelier de simulation sera donc :

- puissant afin d'étudier les futures applications multimédias dont les performances visées sont très élevées et les techniques à développer nombreuses et complexes,
- évolutif et adaptatif afin de pouvoir répondre aux besoins de nombreux projets dans différents domaines d'application,
- fédérateur et normalisateur, afin de recevoir des modules développés par les spécialistes des différentes techniques de télécommunication et de leur proposer un environnement global,
- disponible pour les promoteurs de systèmes spatiaux en leur assurant le niveau requis de confidentialité.

1.6 Les partenaires du projet

- PRiSM
- ENST
- LM2S - UTT
- INT
- LIRMM
- INRIA
- LIP6
- CETP
- SUPELEC
- CNES
- ALCATEL ESPACE
- FRANCE TELECOM - CNET
- MATRA MARCONI SPACE

1.7 Notre contribution

Notre contribution est une contribution transversale. Bien que nos travaux soient principalement basés sur l'étude de différentes techniques d'ordonnancement afin de garantir une qualité de service (Convention 3), une partie est aussi utilisable par la convention 2 avec les problèmes liés aux *handover* et par la convention 5 avec l'évaluation de *Network Simulator*.

Deuxième partie

Constellation de Satellites

Chapitre 2

Constellation de Satellites

2.1 Description d'un satellite

Cette première partie a pour but d'étudier en détails les différents types de satellites et de constellations.

Un satellite est caractérisé par :

- son orbite ou altitude
- sa trajectoire
- ses méthodes de communications ainsi que leur traitements
- ...

2.1.1 L'altitude, variable caractérisant tout un système

Un satellite est caractérisé par son altitude par rapport à la surface de la Terre. On distingue trois types de satellites :

- LEO : Orbite basse ou Low Elevation Orbit
- MEO : Orbite moyenne ou Medium Elevation Orbit
- GEO : Orbite Géostationnaire

Le Schéma 2.1 représente graphiquement les différentes orbites.

Le Tableau 2.1 nous montre les différentes altitudes possibles définissant une constellation de satellites.

| Type | LEO | MEO | GEO |
|----------|-------------|---------------|----------|
| Altitude | 700-1500 km | 8000-15000 km | 36000 km |

TAB. 2.1 – Types d'orbites

L'orbite d'un satellites joue sur les facteurs suivants :

- Le diamètre de l'empreinte du faisceau radio du satellite sur la Terre
- Temps de propagation du signal
- ...

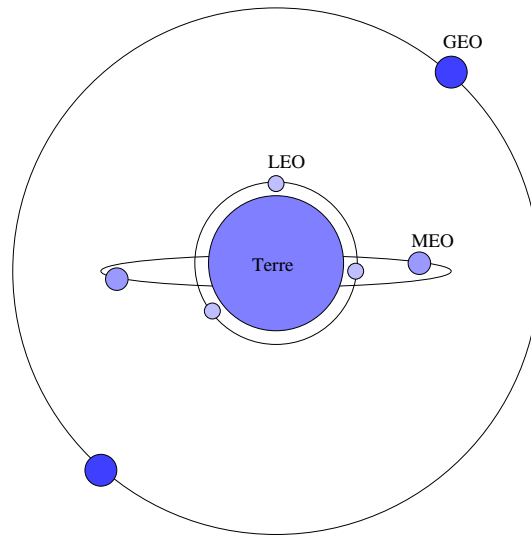


FIG. 2.1 – Les types de satellites en fonction de leur altitude

2.1.2 Temps de propagation

Le temps de propagation du signal de bout en bout ou " End-to-End " dans un réseau est primordial pour les performances de celui-ci. En fonction des différents temps obtenus pour chaque type de satellites, des applications et services sont envisageables.

| | LEO | MEO | GEO |
|----------------------------|----------|--------|-------------|
| Tps End-To-End | 10-15 ms | 100 ms | 250 ms |
| Dif Tps max/Tps moy | grande | faible | negligeable |

TAB. 2.2 – Temps de propagation d'un signal End-To-End

2.1.3 Limitation des traitements

La limitation du nombre de traitement est liée aux empreintes des satellites et des ressources énergétiques dont les satellites disposent. Plus l'orbite est grande, plus le footprint est grand et donc plus le nombre de communications possibles est important.

| | LEO | MEO | GEO |
|--------------------------------------------------------------------------------------|-----|-------|-----|
| Nombre de satellites nécessaire pour une couverture totale du globe terrestre | 70 | 15-16 | 4 |

TAB. 2.3 – Couverture globale

Le rapport " nombres de clients pouvant émettre ou recevoir / nombres de clients total dans l'empreinte " décroît avec la distance. Ce rapport détermine un seuil au delà duquel le

satellite ne peut accepter de nouvelles communications ou clients.

| | LEO | MEO | GEO |
|---------------------------------------------------------------------|-------------|-----|-------------|
| Nombre de communications pouvant être traitées simultanément | en millions | - | en milliers |

TAB. 2.4 – Nombres d'opérations pouvant être traités

2.1.4 Partages des fréquences

Pour qu'il n'y ait pas d'interférences lors des communications, les satellites dont le footprint est voisin, doivent utiliser des fréquences d'émissions différentes. Par conséquent, plus le footprint est petit, plus les fréquences sont réutilisables.

| | Up-link (Ghz) | Down-link (Ghz) |
|------------|------------------|------------------|
| C | 4(3.7-4.2) | 6(5.925-6.425) |
| Ku | 11(11.7-12.2) | 14(14.0-14.5) |
| Ka | 20(17.7-21.7) | 30(27.5-30.5) |
| L/S | 1.6(1.610-1.625) | 2.4(2.483-2.500) |

TAB. 2.5 – Fréquences utilisées par les satellites

2.1.5 Conditions météorologiques

Si les conditions climatiques ne sont pas favorables sur une zone géographique, les communications via GEO sont pénalisées. Etant donné que plusieurs des LEO et des MEO couvrent cette même zone, si un satellite est sujet à des conditions météorologiques désastreuses, les autres satellites prennent le relais.

Les mauvaises conditions météorologiques entraînent bien d'autres perturbations aux niveaux des communications *uplink* ou *downlink*. Un canal peut devenir mauvais. On interrompt l'émission sur ce canal. Il faut alors redistribuer le flux de données sur les autres canaux. Comment faire cette nouvelle allocation ? Une fois le canal rétabli, quels sont les choix à faire pour que la connexion récupère le déficit qu'elle a subi pendant les perturbations ? Une des solutions envisageable pour résoudre ces types de problèmes est de faire appel à une technique d'ordonnancement, pour réguler les flux.

2.1.6 Routage et Commutation sur satellites

Pour effectuer le routage et la commutation, il faut au moins 2 à 3 ms au processeur de bord. Pour un GEO, cela représente seulement 2% d'un " End-to-End ", valeur qui peut être négligée. Il en va autrement pour un LEO ou MEO, où les temps de propagation sont du même ordre que celui de la commutation. C'est un point capital à optimiser pour qu'un système à base de LEO puisse fournir des QoS aux utilisateurs.

2.1.7 Les différents types de routages

Trois types de routage existent :

- Le routage Terrestre (Fig 2.2)

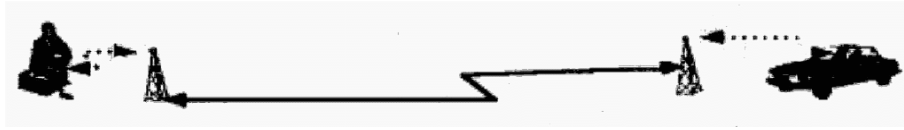


FIG. 2.2 – Routage terrestre - Ground Base Routing

- Le routage via Satellites (Fig 2.3)

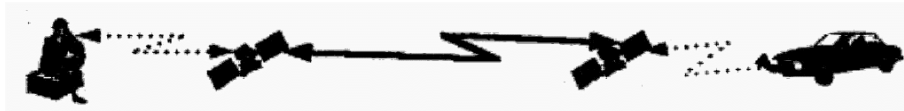


FIG. 2.3 – Routage via satellites - Satellites Base Routing

- Routage composite (composition des deux premiers routages)



FIG. 2.4 – Routage "Bent Pipe"

Le temps de routage et de commutation est fixe. C'est une constante liée au hardware du satellite. Le Tableau 2.6 nous montre l'importance du temps de routage ou de commutation par rapport au temps de propagation.

| | LEO | GEO |
|----------------------------------|-----|----------|
| Temps de commutation (ms) | 5 | 5 |
| Temps de propagation (ms) | 3-8 | 200 et + |

TAB. 2.6 – Temps de routage et de commutation

Il faut faire attention à ce temps de commutation dans des systèmes composés de LEOS : le temps de traitement dégrade de façon considérable la qualité de service sur un tel réseau en ayant un temps de traitement trop proche du temps de propagation. Ainsi des techniques d'ordonnancement compliquées, qui requièrent des temps de traitements élevés ne sont pas envisageables sur des systèmes à bases de LEO et MEO.

2.2 Caractéristiques propres aux constellations de satellites

Une multitude de satellites identiques forme une constellation de satellites. Une constellation est donc un réseau de satellites. Ce réseau doit obéir à certaines règles qui seront détaillées

dans les paragraphes suivants. Ces règles sont les suivantes :

- Configuration spatiale et temporelle des satellites
- Répartition géographique des stations terrestres gérant la constellation
- Propriétés propres aux constellations

2.2.1 Configurations des constellations

La configuration d'une constellation dépend de deux facteurs :

- Les trajectoires des satellites
- Le répartition des satellites sur ces trajectoires

2.2.2 Trajectoires des satellites

Il existe trois types de trajectoires (voir Annexe A pour les figures) :

- trajectoire polaire
- trajectoire équatoriale
- trajectoire en rosette

2.2.3 Répartition des satellites le long de ces trajectoires

A chaque type de trajectoires correspond un modèle de répartition des satellites. Ce modèle doit prendre en compte le nombre de satellites, leur vitesse. D'un point de vue géométrique, les satellites sont les sommets d'une géodésique (ex : dômes radars). Pour des LEOS et MEOS, cette géodésique est mobile autour de la Terre car les satellites se déplacent sur leur trajectoires respectives. De plus pour les LEOS et les MEOS ont distingué deux cycles :

- cycle de rotation autour de la Terre (longitude - longitude) : pour un LEOS ou un MEOS ce cycle est effectué en quelques heures. Ce cycle est lié à la vitesse des satellites par rapport à un repère Galiléen au centre de la Terre. Un GEOS n'est pas considéré par ces mesures car il est stationnaire.
- cycle de rotation autour de la Terre (latitude - latitude) : La rotation de la Terre induit une dérive suivant la latitude. Un satellite basse orbite ne sera à la verticale d'un même point fixe sur Terre qu'au bout de quelques jours. On dit que le satellite balaye la surface terrestre.

La géodésique d'une constellation de satellites est donc soumise à un mouvement complexe autour de la Terre et doit être judicieusement choisie pour que l'utilisation de cette constellation soit optimale.

2.2.4 Liaisons Intersatellites

Une constellation de satellites n'a d'intérêt que si ceux ci communiquent entre eux. Suivant la géodésique de la constellation, on détermine le nombre de liens inter-satellites nécessaires pour le bon acheminement des informations transitant par le réseau de satellites.

Dans ce domaine, toutes les options sont envisageables, comme on peut le voir sur les schémas suivants Fig 2.5 et Fig 2.6.

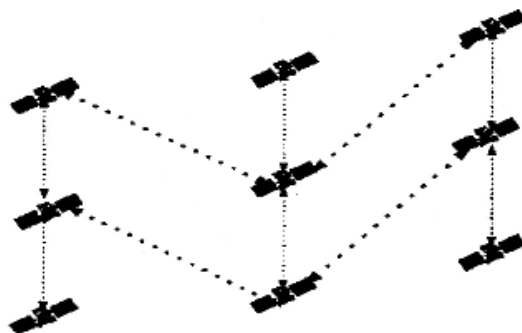


FIG. 2.5 – Liens Inter-satellites pour la constellation Iridium

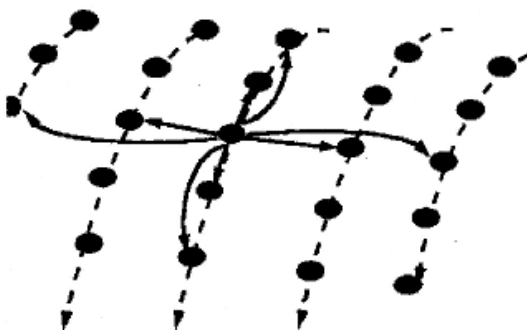


FIG. 2.6 – Liens Inter-satellites pour la constellation Télédesic

Troisième partie

Etat de l'art des techniques d'ordonnancement

Les réseaux informatiques autorisent les utilisateurs à partager les ressources du réseau, comme les imprimantes, les fichiers, les liens de connections. Dans un premier temps, on s'intéresse aux problèmes des acceptations des nouveaux appels des clients dans un réseau. On se base sur des algorithmes de réservation de la bande passante pour résoudre ces problèmes. Une fois la demande de connexion du nouvel appel acceptée, il faut traiter les données de ce nouveau client sans trop perturber les autres connexions déjà existantes. Cela revient à essayer de respecter la qualité de service (QoS) de chaque connexion. Pour respecter ces différentes QoS, on peut utiliser des contrôles de flux comme le *leaky bucket* (shaping de flux) et utiliser des disciplines d'ordonnancement.

Si on considère un ensemble des paquets dans la file d'attente d'un serveur, quelle sera la politique de service de ce serveur pour déterminer le prochain paquet à être traité? Ces politiques de services utilisent des algorithmes de contrôle et d'ordonnancement pour atteindre leur but.

Pourquoi a-t-on besoin de disciplines d'ordonnancement? Tout simplement pour faire cohabiter sur les réseaux plusieurs types de trafics totalement différents, par exemple le *Best Effort*, dont performances sont dites élastiques car elles s'adaptent aux ressources existantes. Les clients *Best Effort* n'ont aucune QoS (aucune garantie en termes de délais, de gigue ...). Les applications qui ont besoin de QoS forment un deuxième type de trafic. Face à ces deux trafics concurrentiels aux ressources, il faut proposer des solutions facilitant la cohabitation des flux dans les réseaux actuels et futurs. On peut trouver des solutions à ce problème en utilisant des disciplines d'ordonnancement.

Dans cette troisième partie, nous allons décrire les propriétés nécessaires pour définir correctement une discipline d'ordonnancement. Ensuite un état de l'art des différentes techniques d'ordonnancement sera effectué.

Chapitre 3

Objectifs des disciplines d'ordonnancement

Une discipline d'ordonnancement doit satisfaire les quatres propriétés suivantes, parfois contradictoires :

- optimiser l'utilisation des ressources afin d'augmenter la capacité du système).
- des performances bornées pour les connexions à QoS
- une équité et une protection des connexions
- une implémentation aisée

En combinant ces quatres propriétés, on obtient une discipline d'ordonnancement. Toutefois, cette discipline est soumise aux mêmes contraintes que celles de ses propriétés. Ainsi si on favorise une propriété, c'est au détriment des autres. Aucune discipline d'ordonnancement peut satisfaire pleinement ces quatres propriétés.

3.1 La facilité d'implémentation

Plus le débit du réseau est important, plus le temps de service doit être faible, surtout au niveau de la couche radio. De ce fait, le temps de décision doit être très court. Or le temps de calcul lié cette décision dépend de la complexité de l'algorithme d'ordonnancement. Plus il est compliqué, plus le nombre de micro-instructions sera grand, donc plus le temps de calcul sera élevé (et inversement). On peut aussi considérer le fait que la discipline est plus ou moins *scalable* i.e. le temps de calcul de décision doit être, si possible, indépendant du nombre de connexions régies par un algorithme d'ordonnancement.

De plus certains algorithmes nécessitent l'organisation des paquets à même la mémoire, ce qui est matériellement très difficile à implémenter. En utilisant des réseaux actifs pour programmer les noeuds du réseau avec des disciplines d'ordonnancement, on peut obtenir des implémentations rapides mais là encore, tout dépend de la complexité des algorithmes et du matériels.

3.2 L'équité et la Protection

La répartition équitable des ressources entre les différentes connexions est une propriété souhaitée pour une discipline d'ordonnancement.

La protection des connexions consiste à réguler tout comportement déviant d'une ou de plusieurs de ces connexions, enfreignant leur droits, comme le dépassement du débit maximal autorisé. Cela arrive même quand les connexions sont soumises à du *shapping* et du *policing*.

La relation entre équité et protection est très proche car en utilisant un algorithme d'ordonnancement équitable pour gérer les connexions, le nombre de comportement hors norme baisse, mais n'est jamais nul. Le système est ainsi mieux géré.

3.3 Les contraintes liées aux performances

La troisième propriété attendue d'une discipline d'ordonnancement est celle qui autorise l'opérateur du réseau à réserver des ressources pour garantir la QoS demandée par un client. On ne peut garantir une QoS à des clients que si on réserve des ressources du réseau pour ces mêmes clients. On observe donc une relation étroite entre le client et le réseau : si le client respecte les limitations fixées par le réseau alors le réseau sera capable de le desservir selon ses exigences. Il reste à fixer les bornes de qualité de services acceptables par les clients et le réseau.

Comment déterminer ces bornes ? Selon les méthodes de calcul utilisées, on distingue deux types de bornes : les bornes déterministes et les bornes statistiques.

On sait calculer les premières si on réserve au débit crête et qu'on connaît le comportement de toutes les connexions. Les bornes statistiques sont quand à elles calculées de manière probabiliste (multiplexage statistique des connexions pour optimiser l'utilisation de la bande passante).

Quatre paramètres entrent en jeu pour calculer ces bornes :

- La bande passante
- Le délai
- La gigue
- Le taux de pertes de paquets au sein d'une même classe de service

Chapitre 4

Les choix fondamentaux

Les quatre principaux degrés de liberté dans une discipline d'ordonnancement sont :

- Le nombre de niveaux de priorité.
- Travail conservatif ou non-conservatif.
- Le degré d'agrégation des connexions d'un même niveau de priorité.
- Le service au sein d'une même classe de service.

Nous allons les examiner dans cette partie.

4.1 Les priorités

Chaque connexion est associée à un niveau de priorité. Dans un réseau à intégration de services, on peut s'attendre à au moins trois niveaux priorités : " haute priorité " pour les messages importants (contrôle du réseau), " priorité moyenne " pour les connexions à garantie de services et " faible priorité " pour le trafic *Best Effort*.

4.2 Travail conservatif ou non-conservatif

Une discipline " conservatrice " est une discipline où l'ordonnanceur n'est au repos que s'il ne reste aucun paquet dans la file en attente d'un service. Au contraire, une discipline non-conservatrice est une discipline où l'ordonnanceur peut être au repos même si des paquets sont en attente d'un service.

4.3 Degré d'agrégation

Le troisième degré de liberté est celui avec lequel les connexions individuelles sont agrégées pour décider de leur ordre de service. Dans un cas extrême, le *scheduler* utilise une unique variable d'état pour caractériser toutes les connexions, qui doivent avoir la priorité ou la même qualité de services. A l'autre extrême, chaque connexion possède une variable d'état unique. Tandis que dans un cas intermédiaire, on a autant de variables d'état que de priorités ou de

classes de services dans le réseau (le *scheduler* traite tous les paquets provenant de connexions ayant la même classe de service de la même manière).

4.4 Agrégation, Ordre et Priorité des services

Comment choisir la bonne discipline d'ordonnancement pour servir les paquets d'une même classe de trafic. On a le choix entre servir les paquets dans leur ordre d'arrivée ou bien les servir dans le désordre en utilisant un *tag* de service. Les propriétés du *scheduler* dépendront énormément de la façon dont les tags seront calculés.

Chapitre 5

Les différentes techniques d'ordonnancement

5.1 Le First In First Out (FIFO)

Le *First In First Out* est la discipline de service la plus simple (algorithmique comme pour l'implémentation). Le serveur sert les clients dans l'ordre de leur arrivée. Il ne sait pas faire la différence entre des paquets de plusieurs connexions. Il n'y a aucune QoS possible avec le FIFO.

5.2 Le Round Robin (RR)

Une évolution du FIFO a été de prendre en compte le nombre de connexions. Ainsi l'ordonnanceur RR sert tour à tour un paquet de chaque connexion. C'est un progrès mais la nature des connexions n'est pas prise en compte : les connexions à QoS sont servies comme du *Best-Effort*.

5.3 Weighed Round Robin (WRR)

Le WRR est une version améliorée du Round Robin. Il gère les différentes priorités des connexions en les répercutant sur le quantum de temps. Le quantum de temps est le temps mis par le serveur pour servir un paquet d'une taille d'une connexion. La somme de tous ces quantum donne le temps d'un cycle mis par le serveur pour servir au moins un paquet de chaque connexion. Dans le cas du Round Robin, tous les quantum sont égaux mais pour le WRR, les quantum sont pondérés par les priorités des connexions. Plus une connexion a une priorité élevée, plus son quantum de service sera grand. Enfin si les paquets de différentes connexions ont des tailles différentes, alors WRR divise chaque poids par la taille moyenne des paquets d'une connexion donnée. On a ainsi normalisé les poids de toutes les connexions. Le seul problème pour effectuer cette normalisation est que le serveur WRR doit connaître la taille moyenne des paquets d'une connexion à l'avance c'est à dire à l'établissement de la connexion. Si tel n'est pas le cas, le serveur WRR ne pourra pas jouer son rôle d'ordonnanceur équitable.

5.4 Deficit Round Robin (DRR)

Le DRR est une version améliorée du WRR. Il n'a plus besoin de connaître la taille moyenne des paquets de chaque connexion. Il utilise un compteur de " déficit " pour chaque connexion. Ce compteur est initialisé à 0 lors de l'établissement de la connexion. Le *scheduler* visite chaque connexion, les une après les autres, et essaye de servir un quantum de bits fixé par le *scheduler*. On distingue trois cas (on note D_c le Compteur de déficit, Q le quantum, et t la taille du paquet) :

- Si $D_c + Q \geq t$ alors le paquet est servi et $D_c = D_c - t$
- si $D_c + Q < t$ alors le paquet n'est servi et $D_c = D_c + Q$
- Si $D_c \neq 0$ et aucun paquet présent dans le file alors $D_c = 0$

5.5 Generalized Processor Sharing (GPS)

Le *General Processor Sharing* (GPS) sert les paquets comme s'ils appartenait à différentes files d'attente virtuelles, servant chacune d'entre elles non vide, un quantum infinitésimal de données (on parle ici en bit(s)). Donc, dans un intervalle de temps fini, chaque file sera servie au moins une fois. On peut aussi associé à chaque connexion un poids. Ce poids servira à calculer le quantum de données à servir pour chaque connexion. Comme on peut le voir, GPS est idéal mais il n'en reste pas moins qu'il n'est pas implémentable à l'heure actuelle. Idéal mais utopique. On peut toutefois utiliser le schéma principal de GPS tout en le simplifiant.

Une amélioration possible du GPS est de ne plus considérer un quantum de données en bits mais en paquets. On obtient alors le Paquets General Processor Sharing (PGPS).

5.6 Weighed Fair Queuing (WFQ)

WFQ est un GPS paquet par paquet. Les quantum de temps ou de données et la taille moyenne des paquets sont devenus de paramètres inutiles pour cette discipline. L'idée générale du WFQ est de calculer la date de sortie de service d'un paquet. Le *scheduler* sert les paquets de plus petites dates de fin de service noté F_d (Finish date).

Date de sortie :

Pour calculer cette date de sortie, on a besoin de connaître une autre variable comptant le nombre de cycles effectués par un *scheduler* bit-à-bit pendant un quantum de temps. On note cette variable R_n (Round number). Quand R_n n'est pas entier, cela signifie qu'un cycle est en cours. Il faut noter que chaque cycle est effectué en un temps donné et par ce fait, plus il y a des connexions à servir, plus grand sera le temps pour effectuer un cycle. Par la suite, on va appeler une connexion *active* une connexion dont la plus grande date de sortie des paquets présents dans la file d'attente est plus grand que le R_n courant. Ainsi, le temps mis pour effectuer un cycle est proportionnel au nombre de connexions actives.

Pour une connexion inactive, le F_d d'un paquet arrivant est la somme du R_n courant plus la taille du paquet. Si un paquet arrive sur une connexion active, son F_d est la somme du

plus grand Fd des paquets de la file plus la taille du paquet. Au lieu de voir R_n comme le nombre de cycles courants, on peut se dire que cette variable est en faite une variable réelle qui augmente de façon inversement proportionnelle aux nombres de connexions actives. Avec cette modification, WFQ s'approche encore plus de GPS.

WFQ peut aussi choisir quels sont les paquets à éliminer quand sa file d'attente est pleine. Tant qu'il n'y a pas assez de place pour le nouveau paquet, WFQ supprime les paquets de plus grand Fd de façon décroissante. On obtient ainsi une allocation *min-max* des connexions.

Le problème pour calculer le Fd est de connaître le Rd courant. Or Rd est soumis au problème de *suppression itérative* ("*iterated deletion*" [Keshav 91]). C'est là la principale difficulté d'implémentation du WFQ.

Les propriétés du WFQ :

- Comme WFQ est un modèle approximant GPS, les connexions sont donc protégées des mauvais comportements des autres.
- Dans certaines conditions, une connexion peut obtenir un délais de bout en bout borné même dans le pire des cas possible. Ainsi les ordonnanceurs WFQ peuvent garantir une qualité de services aux connexions temps-réel.

Les inconvénients du WFQ :

- Complexité du calcul pour obtenir R_n .
- Doit connaître l'état de chaque connexion à chaque instant.
- Gestion de la file d'attente complexe et couteuse en temps de calcul.

5.7 Horloge Virtuelle (VC)

L'horloge Virtuelle est une discipline proche du WFQ. Un ordonnanceur à horloge virtuelle marque chaque paquet avec un *tag*. Puis le serveur sert les paquets dans l'ordre respectif des *tags*. Ce *tag* n'est pas calculé pour émuler un GPS mais pour émuler un multiplexage temporel. Le calcul de ce tag ressemble aux calculs pour obtenir le temps de fin, Fd, du WFQ. Seulement ici, le R_n a été remplacé par la date en temps réel. Le VC est donc plus simple à implémenté que le WFQ, car le calcul de R_n n'est plus utile.

Pour les connexions à QoS, le VC est équivalent au WFQ car il fournit la même QoS, les mêmes temps de bout-en-bout. Pour le Best-Effort, le VC est moins équitable que le WFQ car une connexion Best-Effort sera toujours privilégiée par rapport aux autres, par exemple, une connexion aura un débit bien plus important que les autres même si toutes les connexions ont des caractéristiques identiques.

Si on peut implémenter un VC dans un noeud si la majeure partie des connexions sont des connexions à QoS. Mais dès que le trafic Best-Effort est important, le WFQ est plus équitable que le VC.

5.8 "Earliest Due Date"

La *Earliest Due Date* (EDD) est la date estimée d'arrivée prévue au plus tôt d'un paquet. L'ordonnanceur EDD sers les paquets dans l'ordre des date de fin. Si la date réelle (celle de l'ordonnanceur) est proche de la date d'arrivée prévue d'un paquet, alors ce paquet sera plus prioritaire au service que les autres.

Le *Delay Earliest Due Date* (DEDD) est une amélioration du EDD en y ajoutant un date de fin de service, calculée en fonction des paramètres contractés par le client lors de la connexion. Cette date de fin service est généralement la date prévu de sortie de l'ordonnanceur. L'ordonnanceur sers ensuite les paquets en fonction des leur date prévu de fin de service.

Quatrième partie

Comparaison de plusieurs techniques d'ordonnancement

Chapitre 6

Network Simulator et modèles de simulation

Comme il est impossible d'effectuer une évaluation analytique des différentes disciplines d'ordonnancement sur constellation de satellites, vous avons donc simulé différents scénarios avec *Network Simulator* (NS). Nous allons dans cette partie vous décrire NS ainsi que nos modèles.

6.1 Network Simulator

NS [13] est un simulateur à événements discrets dont le but est, comme son nom l'indique, de simuler des réseaux. NS permet de simuler des réseaux terrestres, sans fils ou satellitaires et sait gérer TCP, le routage, le multicast. C'est un simulator très orienté Internet.

Avantages :

- Sources disponibles libres de droits
- Conception Orienté Objet : C++ interfacé avec un TCL
- Les objets existants sont facilement utilisables
- Création des objets en C++, Création de la topologie et des scénario en TCL
- Utilisé dans le monde de la recherche
- Le forum des utilisateurs
- Des scripts décrivant des constellations de satellites utilisables.

Faiblesses :

- Pas de hotline.
- Le manuel n'est pas un modèle de simplicité.
- Pas optimisé.
- Difficultés d'installation suivant le type et la version des systèmes d'exploitation.
- Peu de codes sont commentés.
- Trop grande importance donnée au monde Internet.
- L'interface satellite est toujours en cours de développement.

6.2 Modèle des simulations

Pour évaluer les différents algorithmes d'ordonnancement, nous allons simuler le modèle suivant :

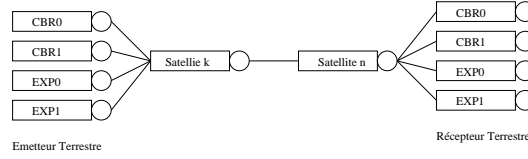


FIG. 6.1 – Scénario

Les connexions ont les paramètres suivant :

CBR 0 Générateur CBR

Taille des paquets : 200

Débit : 448 kbit/s

CBR 1 Générateur CBR

Taille des paquets : 200

Débit : 320 kbit/s

EXP 0 Générateur Exponentiel

Taille des paquets : 200

Débit : 768 kbit/s

Burst Time : 5 s

Idle Time : 10 s

EXP 1 Générateur Exponentiel

Taille des paquets : 200

Débit : 1024 kbit/s

Burst Time : 10 s

Idle Time : 10 s

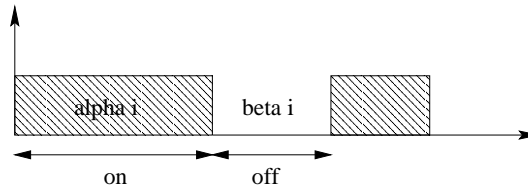


FIG. 6.2 – Traffic On-Off

La formule du débit moyen d'un flux exponentiel est :

$$\tilde{C}_i = \frac{C_i T_{on}}{T_{on} + T_{off}} = \frac{C_i \frac{1}{\alpha_i}}{\frac{1}{\alpha_i} + \frac{1}{\beta_i}}$$

avec :

- \tilde{C}_i le débit moyen de la source exponentielle.
- C_i le debit crête.
- T_{on} la durée d'un *burst*.
- T_{off} la durée d'un repos.

On obtient donc un débit cumulé moyen :

$$\tilde{C} = C_{cbr0} + C_{cbr1} + \tilde{C}_{exp0} + \tilde{C}_{exp1} = 448 + 320 + 256 + 512 = 1.5 Mbit/s$$

Nous avons choisi un débit cumulé moyen de 1.5 Mbit/s pour être en adéquation avec les modèles de constellations de satellites existants dont le lien descendant a une bande passante de 1.5 Mbit/s.

Deplus nous avons utilisé une topologie de satellites LEO à empreintes défilantes pour être là encore en adéquation avec les choix des futures constellations de satellites où les empreintes défilantes ont été préféré aux empreintes fixes. Pour ces raisons, nous avons utilisé le script NS de la constellation Iridium [12].

Pour avoir des courbes de références, nous avons effectué les simulations sur un réseau terrestre filaire. Ensuite nous avons effectué les mêmes simulations sur la constellation d'Iridium. Nous allons de plus utiliser les algorithmes d'ordonnancement déjà existant sous NS : le FIFO, le DRR, le RED, le SFQ (Stochastic Fair Queuing) qui est une variante du WFQ.

Chapitre 7

Résultats et Analyses

7.1 Les sources

Voici les courbes représentant les flux issus des sources, avec deux durées de simulation différentes :

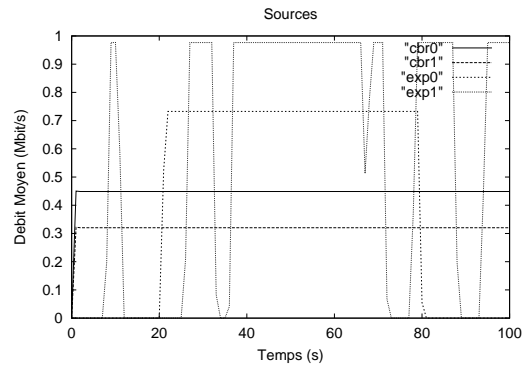


FIG. 7.1 – Les sources avec un temps de simulation de 100s

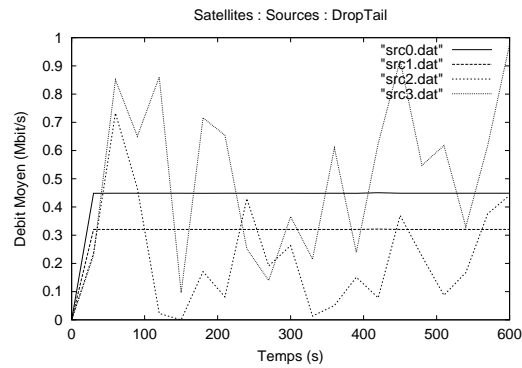


FIG. 7.2 – Les sources avec un temps de simulation de 600s

Nous avons bien des flux CBR et exponentiels, verifiant leur débit moyen.

7.2 Résultats et Analyses

7.2.1 FIFO

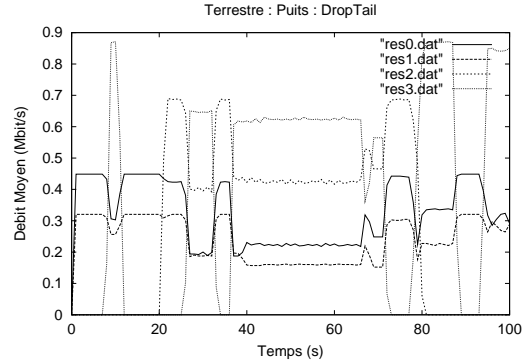


FIG. 7.3 – FIFO sur un réseau terrestre

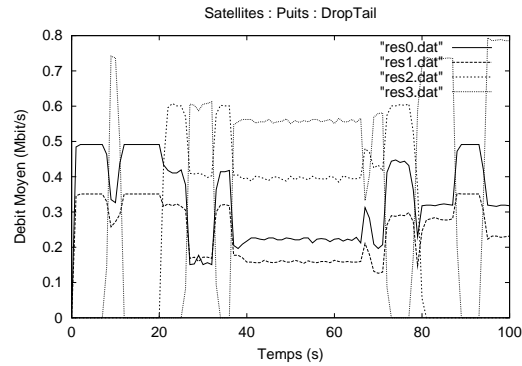


FIG. 7.4 – FIFO sur un réseau de satellites

Quelques soient les classes de trafic, les débits moyens baissent de manière proportionnelle à partir d'un certain moment. Le FIFO n'est donc pas équitable car il traite le CBR comme du Best-Effort. Deplus, les sources On-Off ont une trop grande influence sur le CBR.

Nous allons par la suite utiliser les courbes FIFO comme courbes de références pour nos comparaisons entre les autres algorithmes d'ordonnancement.

7.2.2 RED

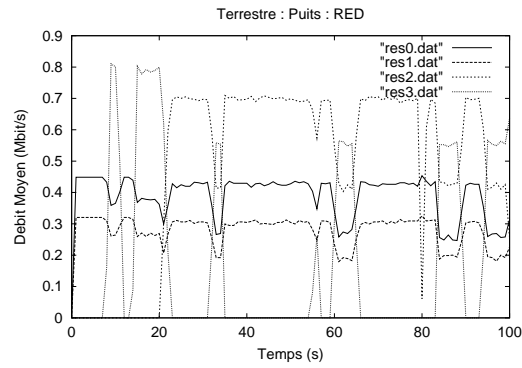


FIG. 7.5 – RED sur un réseau terrestre

Toutes les connexions perdent des paquets mais moins que dans le cas d'un FIFO. Il n'y a qu'une isolation faible des traffics.

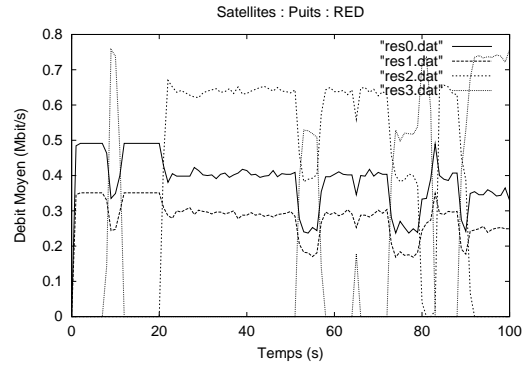


FIG. 7.6 – RED sur un réseau de satellites

Nous obtenons ici presque les mêmes résultats que dans le cas d'un FIFO. Autant utiliser le FIFO pour obtenir les mêmes résultats dans ce cas.

7.2.3 DRR

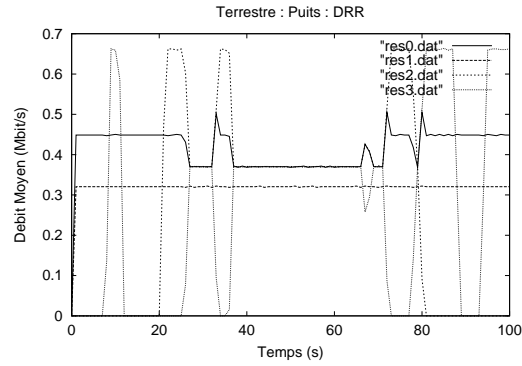


FIG. 7.7 – DRR sur un réseau terrestre

Lors des grands burst des flux Best-Effort, il y a beaucoup de pertes de paquets pour les traffics CBR.. Mais les flux CBR sont touchés de façon progressive : celui qui demande plus de bande passante est touché avant celui qui en demande moins. Le DRR sait donc mieux gérer les flux que le FIFO.

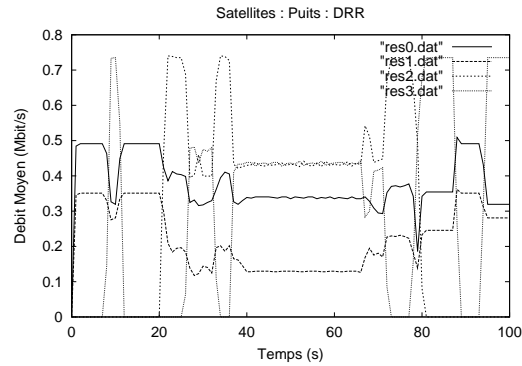


FIG. 7.8 – DRR sur un réseau de satellites

Dans ce cas, avec les satellites, les deux flux CBR sont touchés. L'isolation des flux n'est pas possible. Aucune QoS est possible dans ces conditions. On s'approche du FIFO sans toutefois atteindre son niveau.

Le DRR ou le FIFO sont envisageable pour des sources On-Off quand le nombre de burst est faible ou quand le temps d'un burst est faible. Si tel n'est pas le cas, les flux ne sont plus isolables, et donc on ne peut plus améliorer les performances pour de tels traffics.

7.2.4 SFQ

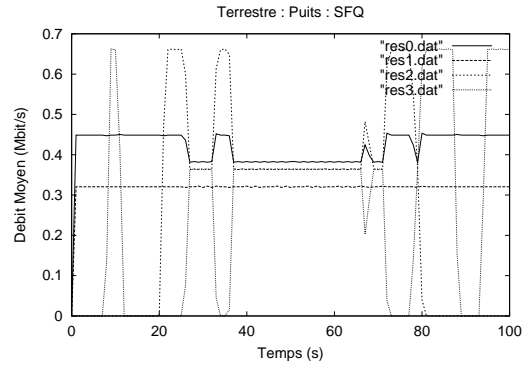


FIG. 7.9 – SFQ sur un réseau terrestre

Il n'y a pas beaucoup de différences entre le SFQ et le DRR. Quand la bande passante est disponible, le DRR redonne plus vite de la bande passante aux connexions que le SFQ, il semble plus réactif. De plus DRR semble privilégier les traffics CBR aux traffics Best-Effort.

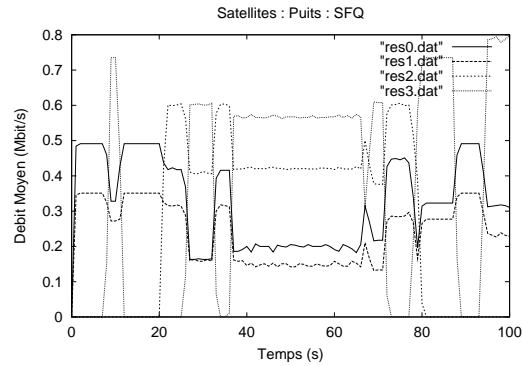


FIG. 7.10 – SFQ sur un réseau de satellites

Au niveau satellite, les débits se dégradent beaucoup trop. Il est presque impossible d'utiliser un tel algorithme d'ordonnancement à ce niveau de dégradation. Ces dégradations sont liées à l'interface radio, au re-routage (hand-over). Une amélioration possible des débits dans ce cas, serait d'ordonnancer au niveau de la couche MAC.

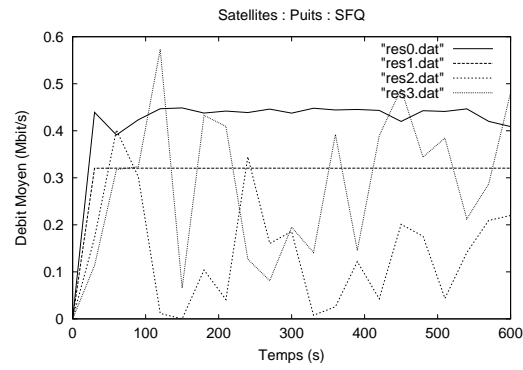


FIG. 7.11 – SFQ sur un réseau terrestre

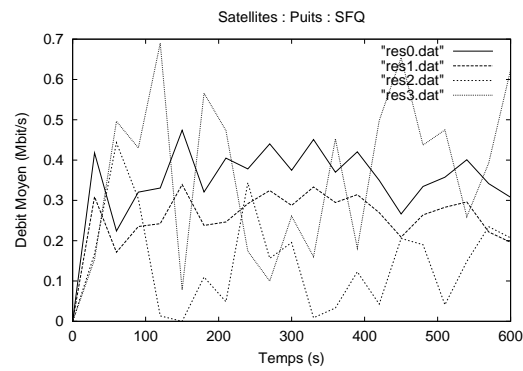


FIG. 7.12 – SFQ sur un réseau de satellites

Avec les courbes 7.11 et 7.12, nous avons une plus grande échelle de temps. Aucun trafic CBR n'a eu sa QoS de respectée dans le cadre d'une constellation de satellites.

Conclusion

Le rôle des satellites devient prépondérant dans le domaine de la communication globale (diffusion TV numérique, radio, Internet, ...) et sera accru dans les prochaines décennies, avec les nouvelles générations de satellites tels que les constellations de LEO et de MEO. De nouveaux services seront accessibles grâce à ces nouvelles technologies comme la téléphonie globale ; ceux-ci reposant sur une demande croissante de qualité de services de la part des utilisateurs.

Nous avons vu que l'environnement satellitaire influence considérablement la qualité de services des connexions, que se soit dû à de mauvaises conditions météorologiques ou bien dû à l'architecture de la constellation elle même. L'ordonnancement des paquets est une solution à ces problèmes : il permet de garantir une qualité de services aux utilisateurs et permet une meilleure gestion des ressources du réseau. Toutefois, aucun algorithme d'ordonnancement terrestre ou cellulaire est applicable tel quel dans le domaine satellitaire, car ils ne tiennent pas compte des différentes propriétés du milieu spatial, comme le délai de bout-en-bout non négligeable, le re-routage des paquets.

La chute des coûts des équipements terrestres a destabilisé ce nouveau marché, entraînant des faillites retentissantes dans le monde satellitaire. Une fois cette crise passée, cette formidable technologie, qu'est la communication à grande échelle par satellites, prendra son essor pour devenir un complément indispensable aux réseaux présents et futurs.

Bibliographie

- [1] Guy Pujolle. *Les Réseaux*. Eyrolles, deuxième édition, 1998
- [2] Sami Tabbane. *Réseaux mobiles*. Hermes.
- [3] S. Keshav *An Engineering Approach to Computer Networking*. Adelson-Wesley.
- [4] Bruno Baynat. *Introduction à la Théorie des Files d'Attente*.
- [5] Bezalel Gavish. *LEO/MEO - Global mobile communication systems*. Telecommunication Systems 8, 1997, pages 99-141.
- [6] Bezalel Gavish, Joakim Kalvenes. *The impact of intersatellite communication links on LEOS performance* Telecommunication Systems 8, 1997, pages 159-190.
- [7] Ralf Boström *A study on the performance of MFTP over satellite* Thèse, mai 1999, <http://home7.swipnet.se/w-75245>.
- [8] Hui Zhang *Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks*
- [9] T.Nandagopal, S.Lu, V.Bharghavan *A Unified Architecture for the Design and Evaluation of Wireless Fair Queuing Algorithms*
- [10] Satellite Data Networks. www.cis.ohio-state.edu/~7Ejain/cis788-97/satellite_data/index.html.
- [11] Lloyd's satellite constellations. www.ee.surrey.ac.uk/Personal/L.Wood/constellations/index.html.
- [12] Iridium. www.iridium.com/index.html
- [13] Network Simulator. <http://www.isi.edu/nsnam/ns/ns-build.html>.

Annexe A

Trajectoires des satellites



FIG. A.1 – Orbite polaire



FIG. A.2 – Orbite équatoriale

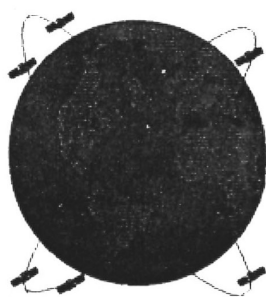


FIG. A.3 – Rossette

Annexe B

Code pour Network Simulator : Terrestre

Voici le code de mon modèle *terrestre* qui m'a servi de bases de comparaisons afin d'étudier les ce même modèle dans un contexte satellitaire.

```
# Stage de DEA
# Vers : 1.0
# Date : 26-09-2k
#
# 0---|
# 1---|
# 2---|----4
# 3---|
#
# 2000 (c) Erwan ERMEL

set ns [new Simulator]

#####
# Creation des nodes
#####

set s0 [$ns node]
set s1 [$ns node]
set s2 [$ns node]
set s3 [$ns node]
set i0 [$ns node]
set d0 [$ns node]

#####
# Creation des links
#####
$ns duplex-link $s0 $i0 5Mb 0ms DropTail
$ns duplex-link $s1 $i0 5Mb 0ms DropTail
$ns duplex-link $s2 $i0 5Mb 0ms DropTail
$ns duplex-link $s3 $i0 5Mb 0ms DropTail
$ns duplex-link $i0 $d0 1.5Mb 0ms DropTail

#####
# Procedure pour attacher des agents expo a des nodes
#####
proc attach-expoo-traffic { node sink size burst idle rate } {
    #Get an instance of the simulator
    set ns [Simulator instance]
    #Create a UDP agent and attach it to the node
```

```

        set source [new Agent/CBR/UDP]
        $ns attach-agent $node $source
        #Create an Expoo traffic agent and set its configuration parameters
        set traffic [new Traffic/Expoo]
        $traffic set packet-size $size
        $traffic set burst-time $burst
        $traffic set idle-time $idle
        $traffic set rate_ $rate
        #Attach the traffic agent to the traffic source
        $source attach-traffic $traffic
        #Connect the source and the sink
        $ns connect $source $sink
        return $source
    }

#####
# Creation des puits pour monitorer les traffics
#####

set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
set sink3 [new Agent/LossMonitor]

$ns attach-agent $d0 $sink0
$ns attach-agent $d0 $sink1
$ns attach-agent $d0 $sink2
$ns attach-agent $d0 $sink3

#####
# Creation des agents et traffics des sources
#####
# source 0 au node s0 : cbr
set source0 [new Agent/RTP]
$ns attach-agent $s0 $source0
set source0agent [new Application/Traffic/CBR]
$source0agent set packetSize_ 200
$source0agent set rate_ 448k
$source0agent attach-agent $source0
$ns connect $source0 $sink0

# source 1 au node s1 : cbr
set source1 [new Agent/RTP]
$ns attach-agent $s1 $source1
set source1agent [new Application/Traffic/CBR]
$source1agent set packetSize_ 200
$source1agent set rate_ 320k
$source1agent attach-agent $source1
$ns connect $source1 $sink1

# source 2 au node s2 : exp
set source2 [attach-expoo-traffic $s2 $sink2 200 5s 10s 768k]

# source 3 au node s3 : exp
set source3 [attach-expoo-traffic $s3 $sink3 200 10s 10s 1024k]

#####
# Declaration des fichiers traces
#####
set f0 [open res0.tr w]
set f1 [open res1.tr w]
set f2 [open res2.tr w]

```

```

set f3 [open res3.tr w]

set d0 [open drop0.dat w]
set d1 [open drop1.dat w]
set d2 [open drop2.dat w]
set d3 [open drop3.dat w]

set r0 [open rece0.dat w]
set r1 [open rece1.dat w]
set r2 [open rece2.dat w]
set r3 [open rece3.dat w]

set s0 [open src0.tr w]
set s1 [open src1.tr w]
set s2 [open src2.tr w]
set s3 [open src3.tr w]

#####
# Procedure record
#####

proc record {} {
    global sink0 sink1 sink2 sink3 f0 f1 f2 f3 d0 d1 d2 d3 r0 r1 r2 r3 s0 s1 s2 s3

    set ns [Simulator instance]

    set time 30

    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    set bw2 [$sink2 set bytes_]
    set bw3 [$sink3 set bytes_]

    set dp0 [$sink0 set nlost_]
    set dp1 [$sink1 set nlost_]
    set dp2 [$sink2 set nlost_]
    set dp3 [$sink3 set nlost_]

    set rp0 [$sink0 set npkts_]
    set rp1 [$sink1 set npkts_]
    set rp2 [$sink2 set npkts_]
    set rp3 [$sink3 set npkts_]

    set now [$ns now]

    puts $f0 "$now [expr $bw0/$time*8]"
    puts $f1 "$now [expr $bw1/$time*8]"
    puts $f2 "$now [expr $bw2/$time*8]"
    puts $f3 "$now [expr $bw3/$time*8]"

    puts $r0 "$now $rp0"
    puts $r1 "$now $rp1"
    puts $r2 "$now $rp2"
    puts $r3 "$now $rp3"

    puts $d0 "$now $dp0 "
    puts $d1 "$now $dp1 "
    puts $d2 "$now $dp2 "
    puts $d3 "$now $dp3 "

    puts $s0 "$now [expr ($rp0+$dp0)*210/$time*8 ]"
    puts $s1 "$now [expr ($rp1+$dp1)*210/$time*8 ]"
    puts $s2 "$now [expr ($rp2+$dp2)*200/$time*8 ]"
    puts $s3 "$now [expr ($rp3+$dp3)*200/$time*8 ]"

```

```
$sink0 set bytes_ 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0
$sink3 set bytes_ 0
$sink0 set nlost_ 0
$sink0 set npkts_ 0
$sink1 set nlost_ 0
$sink1 set npkts_ 0
$sink2 set nlost_ 0
$sink2 set npkts_ 0
$sink3 set nlost_ 0
$sink3 set npkts_ 0

    $ns at [expr $now+$time] "record"
}

$ns at 0.0 "record"
$ns at 0.0 "$source0agent start"
$ns at 0.0 "$source1agent start"
$ns at 0.0 "$source2 start"
$ns at 0.0 "$source3 start"
$ns at 600.01 "exit 0"

$ns run
```

Annexe C

Code pour Network Simulator : Satellites

J'ai donc utilisé des scripts déjà existant sous NS, simulant une constellation de satellites à empreintes défilantes : Iridium. J'y ai ajouté le script de mon modèle : *sat-iridium-flux.tcl* dont voici le code.

```
# Stage de DEA
# File : sat-iridium-flux.tcl
# Vers : 1.0
# Date : 26-09-2k
#
# 0---|
# 1---|
# 2---|----4
# 3---|
#
# 2000 (c) Erwan ERMEL

if {[info exists ns]} {
    puts "Error:  sat-iridium-flux.tcl is a supporting script for the "
    puts "          sat-iridium.tcl script-- run 'sat-iridium.tcl' instead"
    exit
}

#####
#CREATION DES TRAFFICS ET AGENTS DE n100
#####
# AGENT 0 : CBR
set n100ag0 [new Agent/RTP]
$ns attach-agent $n100 $n100ag0
set n100ap0 [new Application/Traffic/CBR]
$n100ap0 set packetSize_ 200
$n100ap0 set rate_ 448k
$n100ap0 attach-agent $n100ag0

# AGENT 1 : CBR
set n100ag1 [new Agent/RTP]
$ns attach-agent $n100 $n100ag1
set n100ap1 [new Application/Traffic/CBR]
$n100ap1 set packetSize_ 200
$n100ap1 set rate_ 320k
$n100ap1 attach-agent $n100ag1

# AGENT 2 : EXP
set n100ag2 [new Agent/CBR/UDP]
```

```

$ns attach-agent $n100 $n100ag2
set n100ap2 [new Traffic/Expoo]
$n100ap2 set packetSize_ 200
$n100ap2 set burst-time 5s
$n100ap2 set idle-time 10s
$n100ap2 set rate_ 768k
$n100ap2 attach-agent $n100ag2

# AGENT 3 : EXP
set n100ag3 [new Agent/CBR/UDP]
$ns attach-agent $n100 $n100ag3
set n100ap3 [new Traffic/Expoo]
$n100ap3 set packet-size 200
$n100ap3 set burst-time 10s
$n100ap3 set idle-time 10s
$n100ap3 set rate_ 1024k
$n100ap3 attach-agent $n100ag3

#####
#CREATION DES PUITES ET AGENTS DE n101
#####
# PUIT 0 <- AGENT 0
set puit0 [new Agent/LossMonitor]
$ns attach-agent $n101 $puite0

# PUIT 1 <- AGENT 1
set puit1 [new Agent/LossMonitor]
$ns attach-agent $n101 $puite1

# PUIT 2 <- AGENT 2
set puit2 [new Agent/LossMonitor]
$ns attach-agent $n101 $puite2

# PUIT 3 <- AGENT 3
set puit3 [new Agent/LossMonitor]
$ns attach-agent $n101 $puite3

#####
#ASSOCIATION DES AGENTS - PUITES
#####
$ns connect $n100ag0 $puite0
$ns connect $n100ag1 $puite1
$ns connect $n100ag2 $puite2
$ns connect $n100ag3 $puite3

#####
#DECLARATION DES FICHIERS RESULTATS
#####
set fic0 [open res0.tr w]
set fic1 [open res1.tr w]
set fic2 [open res2.tr w]
set fic3 [open res3.tr w]

set d0 [open drop0.dat w]
set d1 [open drop1.dat w]
set d2 [open drop2.dat w]
set d3 [open drop3.dat w]

set r0 [open rece0.dat w]
set r1 [open rece1.dat w]
set r2 [open rece2.dat w]
set r3 [open rece3.dat w]

set s0 [open src0.tr w]
set s1 [open src1.tr w]
set s2 [open src2.tr w]

```

```

set s3 [open src3.tr w]

#####
#PROCEDURE RECORD
#####
proc record {} {
    global ns puit0 puit1 puit2 puit3 fic0 fic1 fic2 fic3 d0 d1 d2 d3 r0 r1 r2 r3 s0 s1 s2 s3

    #get an instance of the simulator
    set ns [Simulator instance]

    #Set the time after witch the procedure should be called again
    set time 30

    #How many bytes have been received by the client, traffic sinks ... ?
    set bw0 [$puit0 set bytes_]
    set bw1 [$puit1 set bytes_]
    set bw2 [$puit2 set bytes_]
    set bw3 [$puit3 set bytes_]

    set dp0 [$puit0 set nlost_]
    set dp1 [$puit1 set nlost_]
    set dp2 [$puit2 set nlost_]
    set dp3 [$puit3 set nlost_]

    set rp0 [$puit0 set npkts_]
    set rp1 [$puit1 set npkts_]
    set rp2 [$puit2 set npkts_]
    set rp3 [$puit3 set npkts_]

    #Get the current time
    set now [$ns now]

    #Calculate the bandwidth (in MBit/s) and write it to the files
    puts $fic0 "$now [expr $bw0/$time*8]"
    puts $fic1 "$now [expr $bw1/$time*8]"
    puts $fic2 "$now [expr $bw2/$time*8]"
    puts $fic3 "$now [expr $bw3/$time*8]"

    puts $r0 "$now $rp0"
    puts $r1 "$now $rp1"
    puts $r2 "$now $rp2"
    puts $r3 "$now $rp3"

    puts $d0 "$now $dp0 "
    puts $d1 "$now $dp1 "
    puts $d2 "$now $dp2 "
    puts $d3 "$now $dp3 "

    puts $s0 "$now [expr ($rp0+$dp0)*210*8/$time ]"
    puts $s1 "$now [expr ($rp1+$dp1)*210*8/$time ]"
    puts $s2 "$now [expr ($rp2+$dp2)*200*8/$time ]"
    puts $s3 "$now [expr ($rp3+$dp3)*200*8/$time ]"

    #Reset the bytes_ values on the traffic sinks
    $puit0 set bytes_ 0
    $puit1 set bytes_ 0
    $puit2 set bytes_ 0
    $puit3 set bytes_ 0
    $puit0 set nlost_ 0
    $puit1 set nlost_ 0
    $puit2 set nlost_ 0
    $puit3 set nlost_ 0
    $puit0 set npkts_ 0
    $puit1 set npkts_ 0
    $puit2 set npkts_ 0
    $puit3 set npkts_ 0

```

```

    $puit1 set npkts_ 0
    $puit2 set nlost_ 0
    $puit2 set npkts_ 0
    $puit3 set nlost_ 0
    $puit3 set npkts_ 0

    #Re-schedules the procedure
    $ns at [expr $now+$time] "record"

    puts "Tps(s) : $now"
}

```

```

#####
#LANCEMENT DES AGENTS
#####
$ns at 0.0 "record"
$ns at 0.0 "$n100ap0 start"
$ns at 0.0 "$n100ap1 start"
$ns at 0.0 "$n100ap2 start"
$ns at 0.0 "$n100ap3 start"

```