Self-organizing mechanisms for end-to-end connections

Julien Floquet

université
PARIS-SACLAY

NNT: 2018SACLC102

Thèse de doctorat

CentraleSupélec

Mécanismes auto-organisants pour connexions bout en bout

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

École doctorale n° 580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: mathématiques et informatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 19/12/2018, par

Julien Floquet

Composition du jury:

Inbar Fijalkow Professeur, ENSEA — ETIS	Présidente
Eitan Altman Directeur de recherche, INRIA — Sophia-Antipolis	Rapporteur
Yezekael Hayel Enseignant-Chercheur, Université d'Avignon — LIA	Rapporteur
Tijani Chahed Professeur, Télecom SudParis — Networks and Services Dept	Examinateur
Zwi Altman Ingénieur-Chercheur, Orange — Orange Labs	Directeur
Richard Combes Maître de conférence, CentraleSupélec — L2S	Co-Directeur

Thanks to my supervisors that trusted in me to finish this thesis. Thanks to all my colleagues at L2S in Telecom department and also my colleagues in Orange Labs. Thanks to my wife, my parents and all my family and friends that support me during these 3 years.

Abstract

Fifth generation networks are being defined and their different components are beginning to emerge: new radio access technology, fixed and mobile network convergence and virtualization. The success of these networks requires unified end-to-end management. The effectiveness of the management system can be enhanced by autonomous or SON functions, based on policies. A policy can be a set of rules for activating SON functionalities, activating resources, or choosing data routing.
End-to-end (E2E) control and management of the network have a particular importance for network performance. Having this in mind, we segment the work of the thesis in two parts: the radio access network (RAN) with a focus on Massive MIMO (M-MIMO) technology and the E2E connection from a point of view of the transport layer.
In the first part, we consider hierarchical beamforming in wireless networks. For a given population of flows, we propose computationally efficient algorithms for fair rate allocation including proportional fairness and max-min fairness. We next propose closed-form formulas for flow level performance, for both elastic (with either proportional fairness and max-min fairness) and streaming traffic. We further assess the performance of traffic, and validate the results using numerical experiments. Since the proposed solutions have low complexity compared to conventional beamforming, our work suggests that hierarchical beamforming is a promising candidate for the implementation of beamforming in future cellular networks.
In the second part, we identify an application of SON namely the control of the starvation probability of video streaming service. The buffer receives data from a server with an E2E connection following the TCP protocol. We propose a model that describes the behavior of a buffer content and we compare the analytical formulas obtained with simulations. Finally, we propose a SON function that by adjusting the application video rate, achieves a target starvation probability.

Résumé

Les réseaux de cinquième génération sont en cours de définition et leurs différentes composantes commencent à émerger : nouvelles technologies d'accès radio et virtualisation des réseaux fixes et mobiles. Le succès de ces réseaux nécessitera une gestion de bout en bout unifiée pour la gestion des ressources physiques et virtuelles programmables, de nombreux services, dynamiques, fiables et contextuels, tout en étant économe en énergie et à un coût maîtrisé. L'efficacité du système de gestion peut être améliorée par l'introduction de fonctionnalités autonomes ou SON. Les fonctions SON sont conçues pour configurer, optimiser et réparer de manière autonome le réseau afin de simplifier sa gestion et améliorer ses performances et sa rentabilité. Par exemple dans le réseau LTE l'auto-optimisation permet, entre autres, d'optimiser les paramètres de mobilité et de réduire les interférences grâce à une meilleure allocation des ressources. L'utilisation de fonctions de contrôle autonomes et auto-organisées agissant sur différents segments du réseau vise à optimiser la qualité de service globale.

Dans la première partie du manuscrit, nous avons contribué au MIMO Massif (M-MIMO), l'un des principaux piliers du RAN 5G. Le M-MIMO donne la possibilité d'obtenir des faisceaux focalisés sur l'utilisateur avec un bon rapport de signal sur interférence plus bruit (SINR). Nous proposons de modéliser le M-MIMO avec une table de codage hiérarchique des faisceaux et de mettre en œuvre l'ordonnancement multi-utilisateurs MIMO (MU-MIMO). À cette fin, nous avons développé des modèles analytiques basés sur des systèmes de file d'attente et des techniques d'optimisation. Les modèles analytiques ont été comparés aux résultats de simulation. Tenant compte à la fois de la couche physique et de la nature dynamique du trafic, les algorithmes d'ordonnancement et les performances au niveau du flot d'un M-MIMO constituent un problème difficile et important qui n'a pas reçu beaucoup d'attention dans la littérature. La structure hiérarchique permet de dériver un algorithme d'ordonnancement efficace pour optimiser la fonction utilité α - équité, cela garantit une allocation de ressources équitable relativement facile à mettre en œuvre avec la table de codage hiérarchique des faisceaux, alors que ce n'est pas le cas pour tous les systèmes MIMO. Nous considérons ensuite la dynamique au niveau des flots qui arrivent et partent dynamiquement. Pour le trafic élastique avec allocation d'équité proportionnelle, nous obtenons une formulation exacte (sans approximation) du débit du flot et prouvons que la performance du système est insensible (c'est-à-dire qu'elle ne dépend que de la taille du flot et du taux d'arrivée des flots). Pour le trafic inélastique, nous montrons qu'ils existent des algorithmes pour calculer la probabilité de blocage dans un temps presque linéaire. Nous validons les résultats analytiques par des simulations.

La seconde partie de la thèse vise à développer une fonction de réseau auto-organisant (SON) qui améliore la qualité d'expérience (QoE) des connexions en bout-en-bout. Nous

considérons un service de type vidéo streaming et développons une fonctionnalité SON qui adapte la QoE de bout-en-bout entre le serveur vidéo et l'utilisateur. Le protocole de transport considéré est le TCP (Transmission Control Protocol).
L'objectif de cette partie du manuscrit est de comprendre l'impact des protocoles TCP sur la QoE des clients et d'exploiter ces résultats pour comprendre comment contrôler les ressources. La QoE est représentée par la probabilité de famine qui est le pourcentage que la lecture d'une video streaming se bloque. La raison du blocage est du à la mémoire-tampon (buffer), d'un smartphone par exemple, qui se vide. Une fois le problème ciblé, nous développons un modèle mathématique qui décrit le comportement d'une mémoire-tampon lors d'une diffusion vidéo avec un débit de lecture vidéo fixé au préalable et un débit TCP simulé avec des modèles bien connu dans la littérature. Ce dernier est simulé avec une loi exponentielle qui représente le délai entre deux pertes de paquets consécutives. On utilise le modèle brownien drifté qui nous permet de calculer de manière exacte la probabilité que la mémoire-tampon soit vide. Grâce à ces résultats, nous proposons un mécanisme de contrôle avec un SON qui donne le débit de lecture vidéo. Le but étant d'atteindre la probabilité de famine fixée arbitrairement. Tous les résultats obtenus sont validés par simulation.

Ce manuscrit est organisé en cinq chapitres. Le premier est l'introduction, le deuxième donne le contexte mathématique nécessaire pour comprendre les deux chapitres suivants qui correspondent aux travaux sur la première et deuxième partie de la thèse et le dernier chapitre qui résume et donne des perspectives sur les résultats obtenus. Le deuxième chapitre comporte trois sections. La première section présente les processus de Markov, certaines de leurs propriétés et un cas particulier d'un processus de Markov, utilisé dans le troisième chapitre, la deuxième section parle des processus de naissance et de mort et le troisième définit le modèle brownien drifté. La troisième chapitre explique tous les résultats obtenus en rapport à un algorithme d'ordonnancement multi-utilisateurs pour M-MIMO qu'on proposera et l'évaluation de sa performance. Le quatrième chapitre explique les résultats obtenus pour le contrôle de la QoE d'une connexion E2E. Nous expliquons le modèle mathématique pris en considération pour décrire l'évolution de la mémoire-tampon, nous montrons qu'on peut obtenir une formulation exacte de la probabilité de famine et en fin de chapitre nous proposons une fonction auto-organisante qui ajuste la vitesse de lecture d'une connexion vidéo streaming. Avant la fin du manuscrit, nous donnons une conclusion avec des perspectives sur la suite de travaux à mener sur les sujets abordés, suivie par l'annexe.

Contents

1

Introduction

Fifth generation networks are being defined, and their various components are beginning to emerge: new radio access technologies, fixed and mobile network convergence and programmability (Software Defined). Networking - SDN, Network Function Virtualization - NFV, cloud platforms etc.). The success of these networks will require unified, end-to-end management for managing programmable physical and virtual resources, many diverse, dynamic, reliable and context-aware services, while being energy efficient and a controlled cost. The effectiveness of the management system can be enhanced by autonomous functionalities or SON [[59] - [17]]. SON functions are designed to autonomously configure, optimize, and repair the network to simplify network management, improve performance, and profitability. In the LTE network, for example, self-optimization makes it possible, among other things, to optimize mobility parameters and to reduce interference by better allocation of resources. Several SON for the RAN are developped as the ICIC using a utility-based approach [55], [15] and energy savings [50] and reinforcement learning [21]; and the load balancing [16], [58]. Recently we have works for develop SON for Backhaul, the part of network between the RAN and the CORE, in this work we have an exemple [7]. Several SON for RAN are already deployed by operators as Orange, and could be empowered by a cognitive management system. In the works [19] and [18] they use multi-armed bandit to find the optimal policy for decide to activate or not the automatics mechanisms.
This thesis is part of a research program on the management of networks and services, also covering the future 5th generation networks that is an evolution of traditional 4G-LTE networks and the addition of a new radio access technologies, globally standardized by the 3rd Generation Partnership Project (3GPP) [22], [1]. Among the challenges of the program are the evolutions and solutions for end-to-end quality of experience (QoE) management, including the use of of autonomic, self-organizing control functions acting on different segments of the network to optimize overall quality of service (QoS) and QoE. The management of the end-to-end connection by means of self-organizing functions is the main problem identified for the thesis.

In the first part, we have contributed to Massive MIMO (M-MIMO) that is one of the main pillars of 5G RAN [32], [36]. The benefits are the ability to obtain highly focused beams, giving rise to high signal interference with noise ratio (SINR) and rates. Beamforming in M-MIMO will be used in Frequency Division Duplex (FDD) (codebook based beamforming) and in Time Division Duplex (TDD) (MRT, ZF,) for control and data transmissions. The objectives are to model M-MIMO with hierarchical beamforming, implement Multi-User MIMO (MU-MIMO) scheduling and make a performance evaluation, works are already made in this way as [56]. To this end we have developed analytical models based on queueing systems, and optimization techniques. The analytical models have been compared to simulation results. Scheduling algorithms and flow-level performance of massive Multiple Input Multiple Output (MIMO) taking into account both the physical layer and the dynamic nature of the traffic is a challenging and important problem that has received close to no attention in the literature [30].
Furthermore, the design of efficient Multi-User (MU)-MIMO scheduling algorithms for massive MIMO systems having a hierarchical structure is relatively untouched. We first derive an efficient scheduling algorithm to compute the α-fair allocation, in time $\mathcal{O}(|V| + K)$ where $|V|$ is the number of beams and K the number of flows. This ensures that fair rate sharing is relatively easy to implement for hierarchical beamforming, while this is not the case for all MIMO systems. We then consider flow-level dynamics where flows arrive and depart dynamically. For elastic traffic with Proportional Fair allocation, we derive the expected flow throughput in closed form and prove that the system performance is insensitive (i.e. it depends only on the expected flow size and the arrival rate). For streaming traffic, we show that there exist algorithms to compute the blocking probability in almost linear time $\mathcal{O}\left(|V| \ln |V|\right)$ so that the system is tractable. We conclude with some illustrative numerical experiments.

The second part of the thesis aims at developing Self-Organizing Network (SON) function that optimizes Quality of Experience (QoE) of E2E video streaming connection. There are several indicators of QoE and we consider the starvation of a video streaming represented by a buffer that is emptying during the play-out. The analysis of buffer starvation is considered in several works such as [67], [68].
We have three protocols for sending packets in an E2E connection. We focus on Transmission Control Protocol (TCP) as the transport protocol. The goal of this part of the manuscript is to understand the impact of TCP protocols on the QoE of clients (cf [20] or [12]) and exploit these results for understanding how to control the resources. The QoE is represented with the starvation probability namely that the video streaming stops during its play-out. After having visualized the problem we develop a mathematical model that describes the behavior of a buffer during a video streaming with a particular bitrate and a TCP throughput depending on the characteristics of the traffic. Several works are used markovian queues as model [65], [69] and [66], particularly

they calculate starvation probability with an application-layer approach [38] and [71]. The traffic is simulated with an exponential law that represents the delay between two consecutive packets dropped. The model chosen is the drifted Brownian model, with this model we are able to compute in closed form the probability that the buffer will be empty and, thank this result, we propose a control mechanism as SON, for example, that give the bitrate at the beginning of the E2E connection for obtaining the starvation probability fixed arbitrarily above. All the results obtained are validated with simulations.

This manuscript is organized in five chapters. Following the introduction in Chapter 1, Chapter 2 briefly summarizes the mathematical tools used in the manuscript. The main contributions of the thesis are described in two parts correspond to Chapters 3 and 4. Chapter 5 provides concluding remarks and perspectives for future work.
The second chapter has three sections. The first section presents the Markov processes, some of their properties and a particular case of a Markov process, used in the third Chapter. The second Section recalls birth and death processes, and the third Section defines the Brownian drifted model.
The third Chapter explains all the results obtained in relation to a multi-user scheduling algorithm for hierarchical implementation of M-MIMO that are proposed and the evaluation of its performance.
The fourth Chapter explains the results obtained for controlling the QoE of an E2E connection of a video streaming service. We develop and explain the mathematical models for the buffer and starvation probability, and use these results to develop a self-organizing function that adjusts the rate in which the video server will send the video content.
Chapter 5 provides concluding remarks and perspectives for future investigations. Further details on models and numerical setup used in the manuscript are included in the Appendix.

1.1 List of publications

Journal paper

- J. Floquet, R. Combes, Z. Altman, "Hierarchical beamforming: Resource allocation, fairness and flow level performance", Performance Evaluation, 2018

Conference paper

- J. Floquet, R. Combes, Z. Altman, "Hierarchical beamforming: Resource allocation, fairness and flow level performance", IFIP Performance, 2018

2

Theoretical foundations

2.1 Some properties of Markov processes

This Chapter provides mathematical background and results used thoughout the manuscript. All the results are presented without proofs. The reader with a strong mathematical knowledge can skip this chapter.

2.1.1 Continuous-time Markov chain

A random process $X(t)$, $t \in \mathbb{R}_+$, taking its values in some countable set Ω, is a continuous-time Markov chain if it verifies the following:

$$\mathbb{P}(X(t+\tau) = y | X(t) = x; X(t_1) = x_1, ..., X(t_l) = x_l) = \mathbb{P}(X(t+\tau) = y | X(t) = x)$$

for any $t, \tau \in \mathbb{R}_+$, any states $x, y \in \Omega$, any set of $t_1, ..., t_l \in \mathbb{R}_+$ with $t_1, ..., t_l < t$, and any set of l states $x_1, ..., x_l \in \Omega$.
We denote by $\mu(x \to y)$ the transition rates with homogeneous time of the Markov process:

$$\mu(x \to y) = \frac{d}{dt}\mathbb{P}(X(t+dt) = y | X(t) = x)$$

More details in [54], Section 9.10.

2.1.2 Birth and death process

The birth-death process is a Markov process on $\Omega = \mathbb{N}$ whose transitions increase or decrease the state variable by one. For any $x \in \mathbb{N}^*$, we denote by $\lambda(x) = \mu(x \to x+1)$ the birth rate and $q(x) = \mu(x \to x-1)$ the death rate.
More details in [60], Section 5.17.

2.1.3 Stationary distribution

The probability measure π on Ω is a stationary distribution of a Markov process with continuous time if :

$$\pi(x) \sum_{y \in \Omega} g(x \to y) = 0, \forall x \in \Omega$$

With g the infinitesimal generator defined as $g(x \to y) = \mu(x \to y)$ for $x \neq y$ and $g(x \to x) = -\sum_{y \in \Omega} \mu(x \to y)$. More details in [54], Paragraph 9.10.4.

2.1.4 Reversibility

For a Markov chain with stationary probability π the "detailed balance equations" are:

$$\forall x, y \in \Omega,\ \pi(x)\mu(x \to y) = \pi(y)\mu(y \to y)$$

A Markov chain is reversible if and only if it is in a stationary state and checks the "detailed balance equation". In other terms you have the same probabilty to reach y from x and to reach x from y. More details in [60], Section 4.4.

2.1.5 Stability

Conceptually the stability of a chain is defined by the fact that the number of people waiting to enter the system is a number that even if it takes very large values over time, it will return to take smaller values and will not tend to infinity. The condition for stability of a continuous-time Markov chain is to verify the positive recurrence of the chain. A chain is positive recurrent when all the states are positives recurrents. A state is positive recurrent if

$$\mathbb{E}(T_x | X(0) = x) < \infty$$

with $T_x = \inf\{t > 0 | X(t) = x\}$ the time of the first return to $x \in \Omega$. More details in [60], Section 4.6.

2.1.6 Ergodic theorem

The theorem says that if $X(t)$ with $t \in \mathbb{R}_+$ is irreductible and positive recurrent with a stationary distribution π, then for any bounded function f.

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T f(X(t))dt = E(f(X)) \text{ a.s.}$$

where X is a random variable with distribution π. More details in [60], Section 4.6.

2.2 Queuing systems

2.2.1 Traffic model

For the simulation of traffic through a queueing system we model the arrival rate of customers in the queue and the departure rate of customers. The arrival rate is λ that represents the average number of arrivals in the queue per time unit. The departure rate of customers from a busy server is μ.

To represent the traffic intensity in the server we define the load ρ as the ratio of the arrival rate to the total service rate of the queue, that is:

$$\rho = \frac{\lambda}{\mu}$$

This is a quantity without dimension.
The system works at load $\rho < 1$ and this condition satisfies the stability of the system. More details in [60], Section 1.3.

2.2.2 $M/M/1$ queue

M/M/1 queue models one server with an arrival rate of customers and a service rate of the server. Customers arrive according to a Poisson process of intensity λ and are served during an aleatory time simulated by the exponential distribution with parameter μ. The server can accept an infinite number of customers.

The number of customers in the queue forms a birth-death process, with birth rate λ and death rate of μ.

The stationary mesure of the Markov process is derived from the equation :

$$\forall x \in \mathbb{N},\ \pi(x) = (1-\rho)\rho^x$$

The mean number of customers in the queue is given by:

$$\mathbb{E}(X) = \sum_x x\pi(x) = \frac{\rho}{1-\rho}$$

More details in [60], Section 1.4.

2.2.3 Little formula

If a queue is stable and the arrival rate is equal to λ, we have the following relationship:

$$\mathbb{E}(X) = \lambda\delta$$

with δ the mean time of a customer in the queue. This represents the Little formula and this relationship holds for any stable queueing system.

From the Little formula, we deduce the mean delay, including the queueing and service time, for an M/M/1:

$$\delta = \frac{1}{\mu - \lambda}$$

More details in [60], Section 6.6.

2.2.4 Insensitivity

The queuing system is insensitive when it depends only on the average of the distribution of the service time. That is, if we replace exponential service times of average $\frac{1}{\mu}$ with constant service times equal to $\frac{1}{\mu}$, the distribution of the number of customers does not change. More details in [60], Section 6.8.

2.2.5 PASTA property

PASTA for Poisson Arrivals See Time Averages is a property of Poisson processes. The distribution of customers seen by an arrival to a queueing system is, stochastically, the same as the limiting distribution of customers at that system. More details in [60], Section 6.7.

2.2.6 $M/M/1$ queue with priority

This section is inspired by [2] Chapter 4.5 , we talk about a $M/M/1$ system with traffic units arriving in the queue having k classes of priority indexed by i $(1 < i < k)$, where 1 denotes the class with the highest priority and k the lowest. In this kind of system we have several "priority discipline" that decide how to select the next unit and serves it. We shall talk about the discipline serving units with the highest priority. It must specify the rules for making the decision whether to continue or discontinue the service of the unit being serviced.
The decision to continue the service of the unit currently in service may or may not depend on the state of the system. Since the decision to select the next unit for service depends only upon the priority class, a unit of the i-th class if present is always taken for service prior to the unit of the j-th class $(i < j)$. However, if a unit of the jth class is in service and a unit of the i-th class $(i < j)$ arrives, different alternatives may be followed leading to the disciplines: Preemptive and Head-of-the-line.
The first is when the service of the j-th class unit is immediately interrupted and the server starts serving the ith-class unit. The second is that the service of the j-th class unit is continued to completion, this is also called non-preemptive or postponable.

Preemptive-resume priority

This section is inspired by [2], Section 4.5.1, consider an $M/M/1$ queue with k types of customers. The type i customers arrive according to a Poisson stream with rate λ_i, $i = 1, ..., k$. The service time and residual service of a type i customer is denoted by B_i and R_i, respectively. The type 1 customers have the highest priority, type 2 customers the second highest priority and so on. For the preemptive-resume priority rule interruptions are allowed and after the interruption the service time of the lower priority customer resumes at the point where it was interrupted.
Consider a type i customer, thus there do not exist lower priority customers due to the preemption rule. So we must assume that $\lambda_{i+1} = ... = \lambda_k = 0$. The mean waiting time of a type i customer is denoted by $\mathbb{E}(W_i)$ and define $\rho_i = \lambda_i \mathbb{E}(B_i)$.

$$\mathbb{E}(W_i) = \frac{\sum_{j=1}^{i} \rho_j \mathbb{E}(R_j)}{(1 - (\rho_1 + ... + \rho_i))(1 - (\rho_1 + ... + \rho_{i-1}))}, \ i = 1, ...k$$

2.3 Diffusion modelling

2.3.1 Brownian drift

Suppose that $\mu \in \mathbb{R}$ and $\sigma \in]0, \infty[$. Brownian motion with drift parameter μ and scale parameter σ is a random process $(X_t)_{t \in [0,\infty[}$ with state space $\mathbb{R}$ that are continuous on $[0, \infty[$ with $X_0 = 0$. It has stationary increments, for $s, t \in [0, \infty[$ with $s < t$ the distribution of $X_t - X_s$ is the same as the distribution of X_{t-s}. It has independent increments, for $t_1, t_2, ..., t_n \in [0, \infty[$ with $t_1 < t_2 < ... < t_n$, the random variables $X_{t_1}, X_{t_2} - X_{t_1}, ..., X_{t_n} - X_{t_{n-1}}$ are independent. X_t has the normal distribution with mean μt and variance $\sigma^2 t$ for $t \in [0, \infty[$.
The relation between a Brownian motion with drift and standard Brownian motion is described by

$$X_t = \mu t + \sigma W_t \tag{2.1}$$

for $t \in [0, \infty[$, where $(W_t)_{t \in [0,\infty[}$ is a standard Brownian motion. More details in [40], Chapter 1.

2.3.2 Girsanov theorem

We have $W_t = (W_t^1, ..., W_t^d)$ the d-dimensional Brownian motion and $f = (f_1, ..., f_d)$ the mesurable process such that $f_j \in L^2([0, T] \times \Omega, \lambda \otimes \mathbb{P})$ $\forall$ $0 < T < \infty$ and $j = 1, ..., d$. The λ represents the Lebesgue measure.

Theorem 1. *If* $q_t = \exp(\sum_{j=1}^{d} \int_0^t f_j(s) dW_s^j - \frac{1}{2} \int_0^t |f(s)|^2 ds)$ *is integrable with* $\mathbb{E}(q_t) = 1$,

then for evey $T > 0$ the process

$$B_t := W_t - \int_0^t f(s)ds,\ t \leq T,$$

is a d-dimensional Brownian motion under the new probability measure $\mathbb{Q} = \mathbb{Q}_T$

$$d\mathbb{Q} := q_T d\mathbb{P}$$

Proof in [49], Section 17.3.

2.3.3 Hitting time for Brownian motion with drift

This section shows how to compute the probability of hitting time for drift Brownian motion. We have the following theorem:

Theorem 2. *The probability that drift Brownian motion attains the value $\alpha > 0$, noted $\mathbb{P}(H_\alpha \leq T)$, defined by*

$$H_\alpha = \inf\{t \geq 0 : ct + W_t = \alpha\}$$

is

$$\mathbb{P}(H_\alpha \leq T) = \frac{1}{\sqrt{\pi}} \int_{z \geq \frac{\alpha - cT}{\sqrt{2T}}} \exp(-z^2)dz + \frac{e^{2\alpha c}}{\sqrt{\pi}} \int_{z \geq \frac{\alpha + CT}{\sqrt{2T}}} \exp(-z^2)$$

Proof. We define X_t as a Brownian motion with positive drift $c > 0$ and $X_0 = 0$:

$$X_t = ct + W_t$$

We compute the distribution of the hitting time. For $\alpha = 0$ the hitting time density is trivial because the process starts with $X_0 = 0$. For $\alpha \neq 0$ we notice that :

$$\{H_\alpha \leq t\} = \{\sup_{s \leq t} X_s \geq \alpha\}$$

We need the formula for the joint distribution of $(Y_t, \sup_{s \leq t} Y_s)$, with $(Y_t)_{t \geq 0}$ be a Brownian motion on a probability space $(\Omega, A, \mathbb{Q})$ with $\mathbb{Q} = \exp^{-cY_T - \frac{c^2}{2}T}\mathbb{P}$. The proof of the following joint distribution formula is in [49], Exercise 6.8.

$$\mathbb{Q}\left[Y_t \in dx, \sup_{s \leq t} Y_s \in dy\right] = \frac{2(2y - x)}{\sqrt{2\pi t^3}} \exp\left(-\frac{(2y - x)^2}{2t}\right) 1_{[-\infty, y]}(x)\, dx\, dy. \qquad (2.2)$$

Thanks to Girsanov theorem and joint distribution we obtain :

$$\begin{aligned}\mathbb{P}(H_\alpha \leq T) &= \mathbb{P}(\sup_{s\leq T} X_s \geq \alpha) \\ &= \int \mathbb{I}_{\alpha,\infty}(\sup_{s\leq T} X_s \geq \alpha) d\mathbb{P} \\ &= \int \mathbb{I}_{\alpha,\infty}(\sup_{s\leq T} X_s \geq \alpha) \exp^{cW_t + \frac{c^2}{2}T} d\mathbb{Q}\end{aligned}$$

By Girsanov theorem, $(X_t)_{t\geq T}$ is a Brownian motion with respect to $\mathbb{Q}$ and moreover the joint distribution gives

$$\begin{aligned}\mathbb{P}(H_\alpha \leq T) &= e^{-\frac{c^2}{2}T} \int_{y\geq\alpha}\int_{x\leq y} e^{cx} \frac{2(2y-x)}{\sqrt{2\pi T^3}} e^{-\frac{(2y-x)^2}{2T}} dxdy \\ &= \frac{1}{\sqrt{2\pi T}} \exp\left(-\frac{c^2}{2}T\right)\left(\int_{x\geq\alpha} e^{cx} I_1(x)\, dx + \int_{x\leq\alpha} e^{-cx} I_2(x)\, dx\right)\end{aligned}$$

where

$$\begin{aligned}I_1(x) &:= \int_{y\geq x} \frac{2(2y-x)}{T} \exp\left(-\frac{(2y-x)^2}{2T}\right) dy = \exp\left(-\frac{x^2}{2T}\right) \\ I_2(x) &:= \int_{y\geq\alpha} \frac{2(2y-x)}{T} \exp\left(-\frac{(2y-x)^2}{2T}\right) dy = \exp\left(-\frac{(2\alpha-x)^2}{2T}\right).\end{aligned}$$

Hence,

$$\mathbb{P}(H_\alpha \leq T) = \frac{1}{\sqrt{\pi}} \int_{z\geq\frac{\alpha-cT}{\sqrt{2T}}} \exp(-z^2) dz + \frac{e^{2\alpha c}}{\sqrt{\pi}} \int_{z\geq\frac{\alpha+CT}{\sqrt{2T}}} \exp(-z^2)$$

More details for the proof following the link [52].

□

3

Design and performance analysis of hierarchical beamforming

3.1 Massive MIMO model

Massive MIMO technology is considered as one of the main pillars of 5G RAN, providing means to considerably increase spectral and energy efficiency [43]. In the coming years, large scale antenna systems having hundreds of radiating elements are expected to be deployed (Figure 3.1).
Large scale antenna systems with highly focused beams raise the problem of control channels that, unless being precoded (beamformed), will not match coverage of data channels. In Release 10 of 3rd Generation Partnership Project (3GPP), the concept of precoded Channel State Information Reference Signal (CSI-RS) has been introduced [10]. Release 13 has introduced the Full Dimension (FD)-MIMO with 3D beamforming supported by 2D antenna arrays, with typically dual-polarized 8×8 antenna arrays [28]. With the introduction of massive MIMO in 5G networks, the concept of beam switching or beam sweeping has been proposed. The beams from a given grid of beams are transmitted (in the downlink) or received (in the uplink) in a time interval and in a predetermined way. Beam sweeping can be used in both TDD and FDD for control channels. The attachement of a UE to a beam is performed by means of control channels, namely the UE reports to the base station which is the best received beam. Data is transmitted using the best beam reported by the mobile user in FDD. It is noted that the grid of beams can be useful in TDD when a poor channel or Signal to Interference plus Noise Ratio (SINR) condition occurs.

In prior art solution, beams coverage is generally not uniform across the cells. UEs located at the border of a beam coverage may be penalized with respect to UE located at beam centers. This is due to the beam roll-off, which is unavoidable in the system design. Additionally, there is a significant overlap between beams and UE at beams edge receive interference from the adjacent beams. These two effects are at the source

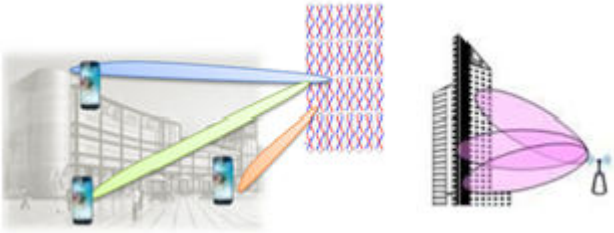


Figure 3.1: Massive MIMO representation

of a SINR degradation, mostly affecting UE at beam edges. The aim of this Chapter is to circumvent this problem while offering optimal cell throughput thanks to spatial reuse.
The purpose of this Chapter is to develop a scheduling algorithm that utilizes the hierarchical beamforming and to analyse the performance of such a scheduler. The concept of beamforming using a codebook is first explained. Following, the proposed model based on flows is presented. We continue explaining how obtain α-fair allocation with scheduler. We present how to obtain the flow-level performance of such a system in closes form.
A flow represents the transmission of data between the base station and a UE using the scheduled beam from the hierarchical codebook. It is noted that as long as the UE velocity is not too high, beamforming schemes perform well.

3.1.1 Antenna modelling

We use the model proposed by A.Tall and Z.Altman [57], which investigates the design of the array of antenna elements for generating focused beams.
In appendix A you find more details about how to compute the gain function of a beam. The following picture schematizes the antenna of a Massive MIMO.

3.1.2 Beamforming with a Codebook

The idea of a dynamic adaptation of the beam to each user often referred to in the literature as beamforming, is to be able to reconfigure the antenna parameters online so that the beam used to serve a particular user is adjusted to get the maximum signal strength. The antenna gains brought by such technology increase with the number of antenna elements of the antenna array, but so does also the feedback overhead needed to adjust the antenna parameters. In order to reduce this overhead, a codebook of antenna beams can be used in which each beam corresponds to a specific configuration of the antenna which defines the beam characteristics, e.g. in terms of direction and beam width. The user feedback can then be used to select the best beam in the codebook. It

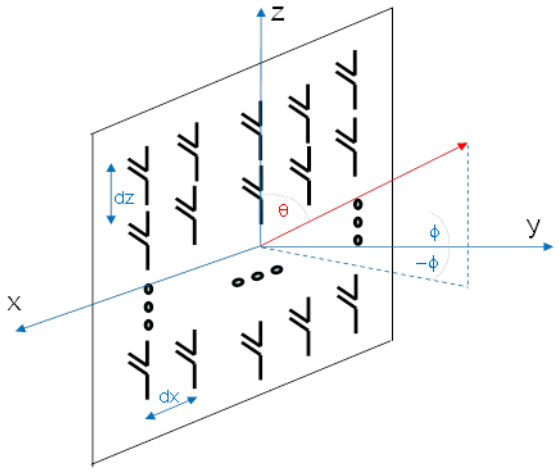


Figure 3.2: Antenna model [57]

is noted that a large codebook will increase the amount of required signaling.

3.1.3 Grid of Beams

Beamforming (BF) is a key advantage of M-MIMO. It boosts the coverage and rate of data channels. Thanks to the new concept of Grid of Beams (GoB), control channels as well can benefit from M-MIMO BF. A GoB is a set of pre-defined narrow beams. Thanks to a beam sweeping mechanism, all beams can be scanned and each UE can select its best beam. However, GoB generates an overhead of control channels and creates small areas between beams with degraded performance. Several new solutions to improve the GoB concept can be envisaged: Beam skipping, Oversampling, Hierarchical Beamforming.
Beam skipping which consists in avoiding to simultaneously activate overlapping beams. The Oversampling which consists of adding overlapping beams on purpose. The Hierarchical beamforming (HBF) consists of a beam planning based codebook according to cell coverage.

Beam Skipping

To reduce overlapping between adjacent beams and the associated inter-beam interference, the concept of beam skipping is introduced: Once a particular UE is scheduled on its best beam, a certain number of configurable beams that are adjacent to the chosen beam are skipped. Subsequent UEs may only be picked from the non-skipped beams, and they are chosen based on the appropriate scheduling metric. It reduces the

interference between beams.

Oversampling

The improvement of cell edge throughput can be achieved by improving the SINR (as with the beam skipping approach). Alternatively, one can improve the received power of UEs that have poor beamforming gain because they are in between adjacent beams using oversampling. Reduce the CSI-RS overhead associated to oversampling, one can combine it with a hierarchical structure. By choosing larger beams, the areas in between beams with performance is reduced and it consequently improves cell edge throughput.

Hierarchical codebook design based on beam planning approach

Recently, the concept of the hierarchical or multilevel codebook of beams (or codebook based beamforming) has been proposed. The term multi-resolution has been used as well and refers to the same concept. One of the motivations behind the idea of hierarchical codebooks is to reduce the signaling overhead of common channels while maximizing the beamforming gain.

Several contributions on hierarchical beamforming are briefly described presently. The first example is a millimeter wave backhaul serving urban pico-cells [26]. In this work, a hierarchical codebook of beams is used to efficiently align a pair of receive-transmit beams. Hierarchical beamforming has also been studied for the Radio Access Networks (RAN). In [3] the authors develop a hierarchical codebook for the training beamforming vectors for millimeter wave cellular systems. The sparse nature of the microwave channel is exploited in order to develop a hybrid analog/digital low complexity precoding algorithm. [63] introduces a generalized detection probability metric for comparing the efficiency of codebooks. An optimization procedure that optimizes this metric by flattening the beam patterns is proposed. The hierarchical codebook of beams can be designed to fit the coverage needs of a cell allowing to further reduce the number of beams in the codebook, as described in [57]. Furthermore, it is shown that the geometry and propagation parameters of the cell should be taken into account when designing the structure of the codebook. For example, in typical dense urban cells, a 3D hierarchical codebook can be used while in the sub-urban environment, a 2D horizontal hierarchical codebook of beams is more appropriate. More references on hierarchical beamforming can be found in [41] which also describes how to design the hierarchical codebook using the Discrete Fourier Transform (DFT) matrix. The hierarchical codebook of beams can be designed to fit cell coverage requirements, and its structure is different according to the type of environment, e.g. sub-urban vs dense urban cells. A remaining challenge is to self-configured the codebook using basic geometrical parameters of the cell, during the thesis we worked on this subject. If certain conditions on the hierarchical codebook structure are satisfied, we will show in chapter 6 of this manuscript that low complexity and efficient MU-MIMO scheduling algorithm

can be derived.
Advantages have been identified to this approach: HBF reduces the overhead of beamformed CSI-RS control channels.

3.2 Multi-User MIMO Scheduling for hierarchical codebook

3.2.1 Mathematical modelization of hierarchical codebook

We consider a cell which is a region in the plane $\mathcal{G} \subset \mathbb{R}^2$. We consider a set V of fixed beams which can be used for transmitting data to flows. To each beam $v \in V = \{1, \ldots, |V|\}$ we associate $\mathcal{G}_v \subset \mathcal{G}$ which is the region covered by beam v, namely one can send data to a flow located in $\mathcal{G}_v$ using beam v. We assume that there exists a directed graph $G = (V, E)$ where the set of vertices V is the set of beams. The graph G is assumed to be a directed tree. For ease of presentation, we assume that the nodes V are sorted by increasing depth, so that 1 is the root of G, and $(v, v') \in E$ implies that $v < v'$. We call the set of beams, the covered regions and the corresponding graph a codebook. For beam $v \in V$ and $x \in \mathcal{G}$, we denote by $g_v(x)$ the signal power received by a flow located at x using beam v. It is noted that in practice, if $x \notin \mathcal{G}_v$, the value of $g_v(x)$ is negligible, but the received signal power is not strictly zero.

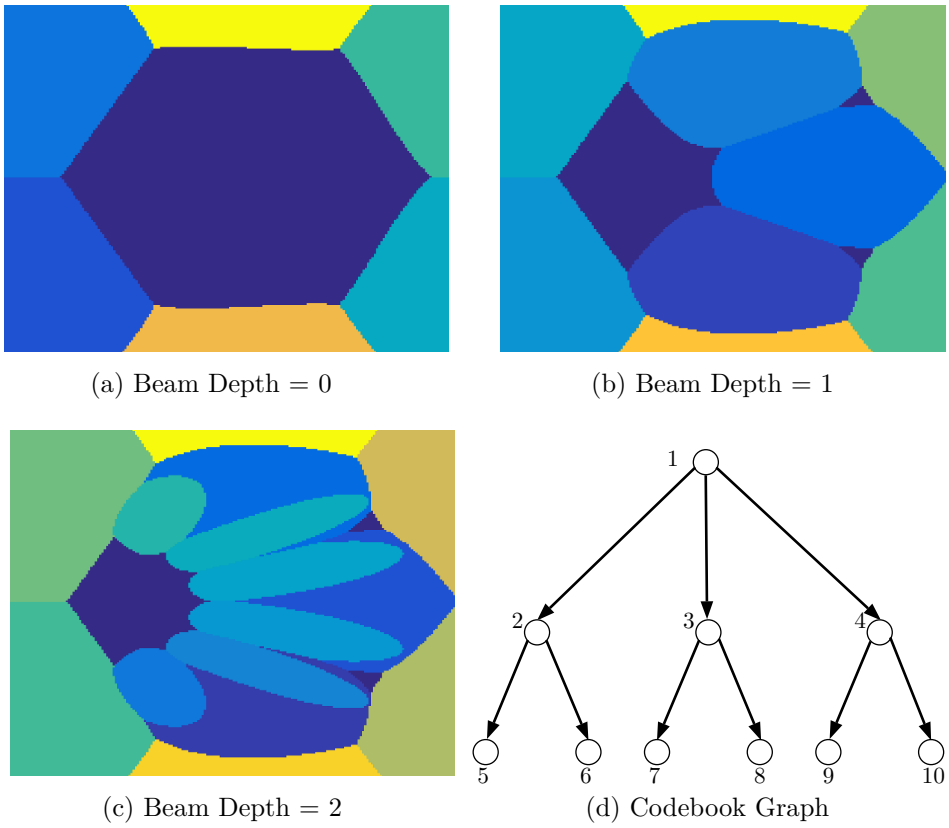
Definition 1. *A codebook is hierarchical if it verifies the following three properties:*

(i) *If* $(v, v') \in E$ *then* $\mathcal{G}_{v'} \subset \mathcal{G}_v$

(ii) *If* $(v, v') \in E$ *and* $(v, v'') \in E$ *then* $\mathcal{G}_{v'} \cap \mathcal{G}_{v''} = \emptyset$.

(iii) *If* $(v, v') \in E$ *and* $x \in \mathcal{G}_{v'}$, *then* $g_{v'}(x) \geq g_v(x)$.

An example of a hierarchical codebook with $|V| = 10$ beams is presented in Figure 3.3. Both the covered regions and the graph G are depicted. As seen from the three above properties and the figure, the regions covered by beams form a hierarchical partition of space. Namely, the children of a beam v cover regions covered by v (property (i)), and the regions covered by distinct children do not overlap (property (ii)). Furthermore if one goes down in the tree, beams become more focused since they cover smaller regions, and the received signal power increases (property (iii)). We also see that hierarchical codebooks are an improvement over a grid-of-beams approach (where one only uses beams of maximal depth): the beams with the highest depth are small, very focused and do not interfere with each other (i.e. the interference is negligible), however any location not covered by those beams is covered by beams of lower depth. Beams of high depth allow for large data rates and beams of low depth ensure coverage to all flows in the cell. Therefore, while the grid-of-beams approach inevitably



(a) Beam Depth = 0 (b) Beam Depth = 1

(c) Beam Depth = 2 (d) Codebook Graph

Figure 3.3: Hierarchical codebook example

creates coverage degradation in between beams, hierarchical codebooks are immune to this problem. For the rest of the chapter we assume that the hierarchical beamforming provides a full coverage of the cell.

3.2.2 Notations

We say that two beams v and v' do not interfere if and only if the regions they cover do not overlap, that is $\mathcal{G}_v \cap \mathcal{G}_{v'} = \emptyset$. We use the following notations:

- $\mathcal{A}(v)$ the set of ancestors of v, and $\bar{\mathcal{A}}(v) = \mathcal{A}(v) \cup \{v\}$.
- $\mathcal{D}(v)$ the set of descendants of v, and $\bar{\mathcal{D}}(v) = \mathcal{D}(v) \cup \{v\}$.
- $d(v)$ the degree of $v \in V$ i.e. the number of its children
- $d(G) = \max_{v \in V} d(v)$ the maximal degree of the graph G.

- $h(G)$ the height of G which is the largest distance between the root and any vertice $v \in V$.

The definition of the main graph-theoretic notions for trees are recalled in the Appendix. We recall a standard fact about trees, which is that the number of edges equals the number of nodes minus one, $|E| = |V| - 1$. Unless stated otherwise, we denote vectors by bold letters (e.g. $\mathbf{x}$), scalars by non-bold letters (e.g. x), and x_k denotes the k-th component of $\mathbf{x}$. Deterministic quantities are denoted by lower-case letters (e.g. x) and random quantities are denoted by upper case letters (e.g. X). For instance $\mathbf{X}$ is a random vector, and X_k is the k-th component of $\mathbf{X}$. Finally sets are denoted by upper case calligraphic letters (e.g. $\mathcal{X}$). For k an integer, we denote by $[k]$ the set $\{1, \ldots, k\}$. We denote by ln the natural logarithm.

3.3 Algorithms for fair rate allocation

In this section we consider a fixed set $\mathcal{K}$ of flows, where $x(k) \in \mathcal{G}$, denotes the location of flow $k \in \mathcal{K}$. We study fair rate allocation strategies. We first determine how flows should be associated to beams, and which sets of beams may be activated simultaneously. We then formulate fair rate allocation as a convex optimization problem and provide a problem-specific efficient algorithm to solve this problem.

3.3.1 Flow association

We assume that each flow $k \in \mathcal{K}$ is associated to a beam $v_k \in V$ which maximizes the signal strength, so that

$$v_k \in \arg\max_{v \in V} g_v(x(k)).$$

Since the codebook is hierarchical, by property (iii), one should associate k to the beam of largest depth which covers location $x(k)$. It is also noted that, from the hierarchy of the codebook, beam allocation to a flow given its location can be performed using a simple and efficient algorithm (Algorithm 1) described below. This algorithm runs

Algorithm 1: Beam association algorithm

Data: Tree G sorted by decreasing height, point x, regions $\mathcal{G}_v$ for $v \in V$
$v \leftarrow 1$;
while $\exists v' : (v, v') \in E, x \in \mathcal{G}_{v'}$ **do**
  $v \leftarrow v'$;
Result: Flow at x associated to beam v

in at most $\mathcal{O}(d(G)h(G))$ iterations. For instance, if G is a regular tree its height is $h(G) = \mathcal{O}(\ln |V|)$, so that the running time is $\mathcal{O}(d(G) \ln |V|)$. This is an advantage of

hierarchical codebooks over the grid-of-beams approach, where flow association usually requires scanning through all the beams, which requires $\mathcal{O}(|V|)$ time. Hence hierarchy allows a considerable improvement from linear to logarithmic complexity. This is critical for real-world implementation of large codebooks, especially since flow association must be updated periodically due to arrival, departure and mobility of flows.

When flow k is served by beam v_k, it may receive data at data rate:

$$r_k = W \log_2 \left(1 + \frac{g_{v_k}(x_k)}{N_0^2}\right),$$

where W is the bandwidth and N_0^2 is the thermal noise power. We denote by $\mathcal{K}(v) = \{k \in \mathcal{K} : v_k = v\}$ the set of flows associated with beam v, and $n_v = |\mathcal{K}(v)|$ the number of flows associated with beam v.

While we do not take into account several cells in this model, we assume that, in a more realistic setting, if all neighboring cells use hierarchical beamforming as well, the resulting inter-cell interference should be negligible since when beamforming with narrow beams is used, two flows would see interference only if they are both physically close to each other and receive data at the same time. Indeed this situation should only happen infrequently.

3.3.2 Beam activation strategies

We now consider that the association of beams to flows v_k, $k \in \mathcal{K}$ is fixed, in other words the flows are static. We determine which sets of beams can be activated simultaneously. To avoid interference, two beams v and v' may be activated at the same time if and only if they do not interfere. By definition of the codebook this is true if and only if v is not a descendant or an ancestor of v'. We say that a subset of V is an admissible activation strategy if all beams in this subset may be activated simultaneously without interfering with each other. We identify subsets of V with vectors of $\{0,1\}^{|V|}$. Denote by $\mathbf{z} \in \{0,1\}^{|V|}$ a subset of beams where $z_v = 1$ if v is activated and $z_v = 0$ otherwise. The set of admissible activation strategies is therefore:

$$\mathcal{Z} = \left\{\mathbf{z} \in \{0,1\}^{|V|} : z_v z_{v'} = 0, \forall v' \in \mathcal{A}(v), v \in V\right\}.$$

Figure 3.4 depicts the possible activation strategies when G is a binary tree of height 2. We denote the activated (resp. non-activated) beams by colored (resp. empty) vertices. It is noted that we solely depict the maximal activation strategies (i.e. for which activating a new beam would render the strategy non admissible). Furthermore, we assume that one can perform time-sharing between the possible activation strategies, since in practice time is slotted and for each time slot one chooses an admissible set of beams to activate. Hence the vector of admissible proportions of time each beam is activated $\boldsymbol{\gamma}$ must verify:

$$\boldsymbol{\gamma} = (\gamma_v)_{v \in V} \in \mathbf{conv}(\mathcal{Z}).$$

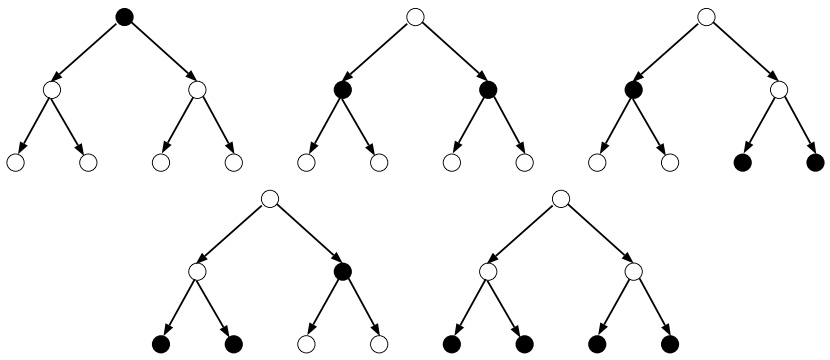


Figure 3.4: Set of admissible activations for a binary tree of height 2.

where **conv** denotes the convex hull, and $\gamma_v \in [0,1]$ is the proportion of time beam v is activated.

3.3.3 Flow activation strategies

When beam $v \in V$ is activated, at most one flow may be served by this beam at a time, and time sharing between flows associated to a given beam applies as well. Denote by $\delta_k \in [0,1]$ the proportion of time flow k is served by beam v_k divided by the proportion of time v_k is activated. The admissible set for $\boldsymbol{\delta} = (\delta_v)_{v \in V}$ is hence:

$$\Delta = \left\{ \boldsymbol{\delta} \in [0,1]^{\mathcal{K}} : \sum_{k \in \mathcal{K}(v)} \delta_k = 1, v \in V \right\}.$$

3.3.4 Fair rate allocations

We now determine fair rate allocation strategies, which involves finding the optimal values of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ to maximize a utility function of the flows data rates. From the above, given $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$, the effective data rate of flow $k \in \mathcal{K}$ is given by $r_k \gamma_{v_k} \delta_k$. We are interested in the family of α-fair allocations (see [39]), which leads to the following optimization problem:

$$\begin{aligned} &\underset{\boldsymbol{\delta},\boldsymbol{\gamma}}{\text{Maximize}} \sum_{v \in V} \sum_{k \in \mathcal{K}(v)} f_\alpha(r_k \gamma_v \delta_k) \\ &\text{subject to } \gamma \in \mathbf{conv}(\mathcal{Z}) \text{ and } \delta \in \Delta, \end{aligned}$$

where

$$f_\alpha(r) = \begin{cases} \frac{r^{1-\alpha}}{1-\alpha} & \text{if } \alpha \neq 1 \\ \ln(r) & \text{otherwise.} \end{cases}$$

It is noted that this formulation includes Proportional Fairness (PF) ($\alpha = 1$), Maximum Throughput (MT) ($\alpha = 0$) and Max-Min Fairness (MMF) ($\alpha \to \infty$) as particular cases. Since this optimization problem is convex, one may find the optimal solution using an iterative method such as gradient descent or Newton method, but we will show that, for any value of α, an optimal solution may be computed by an efficient, problem specific, algorithm.

3.3.5 Explicit expression for the convex hull

Before deriving the optimal rate allocation, we state Proposition 1, an auxiliary result which allows to express the convex hull $\mathbf{conv}(\mathcal{Z})$ in closed form. The interpretation of Proposition 1 is that γ_v can be written as the product $\kappa_v \prod_{v' \in \mathcal{A}(v)}(1 - \kappa_{v'})$, where κ_v is the proportion of time v is activated divided by the proportion of time none of its ancestors are activated. We recall that v may be activated only if all of its ancestors are not activated.

Proposition 1. *We have that:*

$$\begin{aligned}\mathbf{conv}(\mathcal{Z}) &= \left\{ \boldsymbol{\gamma} : \gamma_v = \kappa_v \prod_{v' \in \mathcal{A}(v)} (1 - \kappa_{v'}), \boldsymbol{\kappa} \in [0,1]^{|V|} \right\} \\ &= \left\{ \boldsymbol{\gamma} \in [0,1]^{|V|} : \sum_{v' \in \bar{\mathcal{A}}(v)} \gamma_{v'} \leq 1, v \in V \right\}.\end{aligned}$$

The proof is relatively straightforward. We have for every $v \in V$:

$$\kappa_v = \frac{\gamma_v}{1 - \sum_{v' \in \mathcal{A}(v)} \gamma_{v'}}$$

$$\kappa_v \geq 0 \text{ because } \gamma_v \geq 0 \text{ and } \sum_{v' \in \mathcal{A}(v)} \gamma_{v'} \leq 1$$

$$\kappa_v \leq 1 \text{ because } \sum_{v' \in \bar{\mathcal{A}}(v)} \gamma_{v'} \leq 1$$

We have

$$\left(\kappa_v = \frac{\gamma_v}{1 - \sum_{v' \in \mathcal{A}(v)} \gamma_{v'}} \right) \Rightarrow \left(\gamma_v = \kappa_v \prod_{v' \in \mathcal{A}(v)} (1 - \kappa_{v'}) \right)$$

3.3.6 Efficient algorithms for fair rate allocation

We now derive the optimal rate allocation in closed form, and provide an efficient algorithm to compute it in practice.

Theorem 3. *Define the vectors $\boldsymbol{\phi}, \boldsymbol{\theta}$ and $\boldsymbol{\kappa}^\star$ as follows. Vector $\boldsymbol{\phi}$ is given by:*

$$\phi_v = \begin{cases} \left(\sum_{k\in\mathcal{K}(v)} r_k^{\frac{1}{\alpha}-1}\right)^\alpha & \text{if } \alpha \neq 1 \\ |\mathcal{K}(v)| & \text{otherwise.} \end{cases}$$

Next define $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ through the following recursion. Let

$$\theta_v = \begin{cases} \frac{\phi_v}{1-\alpha} & \text{if } v \text{ is a leaf} \\ (\kappa_v^\star)^{1-\alpha}\frac{\phi_v}{1-\alpha} + (1-\kappa_v^\star)^{1-\alpha}\sum_{v':(v,v')\in E}\theta_{v'} & \text{otherwise.} \end{cases}$$

$$\kappa_v^\star = \left[1 + \left(\frac{1-\alpha}{\phi_v}\sum_{v':(v,v')\in E}\theta_{v'}\right)^{\frac{1}{\alpha}}\right]^{-1}.$$

Then $(\boldsymbol{\delta}^\star, \boldsymbol{\gamma}^\star)$ the unique α-fair allocation is given by:

$$\delta_k^\star = \begin{cases} \frac{r_k^{\frac{1}{\alpha}-1}}{\sum_{k'\in\mathcal{K}(v_k)} r_{k'}^{\frac{1}{\alpha}-1}}, & \text{if } \alpha \neq 1 \\ \frac{1}{|\mathcal{K}(v_k)|}, & \text{otherwise.} \end{cases}$$

$$\gamma_v^\star = \kappa_v^\star \prod_{v'\in\mathcal{A}(v)} (1-\kappa_{v'}^\star).$$

The proof of Theorem 3 is presented in 3.9.1. From Theorem 3 we deduce Algorithm 2, an efficient dynamic programming algorithm to calculate the α-fair allocation given the flow rates r_k, $k \in \mathcal{K}$ and the allocations v_k, $k \in \mathcal{K}$. This algorithm runs in linear time $\mathcal{O}(|V| + |\mathcal{K}|)$ and is hence efficient, since its input has size $\mathcal{O}(|V| + |\mathcal{K}|)$.

3.3.7 Examples

We now illustrate the outcome of the α-fair allocation for the important sub-cases of $\alpha \in \{0, 1\}$. It is noted that the Max-Min Fair allocation (i.e. $\alpha \to \infty$) does not seem to yield a simple formula for arbitrary trees.

Proportional Fairness ($\alpha = 1$) The PF allocation is given by:

$$\delta_k^\star = \frac{1}{n_{v_k}}\ ,\ k \in \mathcal{K}$$

$$\kappa_v^\star = \frac{n_v}{\sum_{v'\in\bar{\mathcal{D}}(v)} n_{v'}}\ ,\ v \in V$$

So that:

$$\gamma_v^\star = \frac{n_v}{\sum_{v'\in\bar{\mathcal{D}}(v)} n_{v'}} \prod_{v''\in\mathcal{A}(v)} \frac{\sum_{v'\in\mathcal{D}(v'')} n_{v'}}{\sum_{v'\in\bar{\mathcal{D}}(v'')} n_{v'}}.$$

Algorithm 2: α-fair allocation

Data: Tree G sorted by decreasing height, flow rates r_k, $k \in \mathcal{K}$, flow association v_k, $k \in \mathcal{K}$, parameter α

`/* Compute the aggregated data rates */`

for $v \in V$ **do**
  $\phi_v \leftarrow 0$;

for $k \in \mathcal{K}$ **do**
  $\phi_{v_k} \leftarrow (\phi_{v_k}^{\frac{1}{\alpha}} + r_k^{\frac{1}{\alpha}-1})^{\alpha}$;

for $k \in \mathcal{K}$ **do**
  $\delta_k^\star \leftarrow \frac{r_k^{\frac{1}{\alpha}-1}}{\phi_v^{\frac{1}{\alpha}}}$;

`/* Dynamic Programming: ascending phase */`

for $v = |V|, |V|-1, \ldots, 1$ **do**
  $\tau_v \leftarrow 0$;
  for $v' : (v, v') \in E$ **do**
    $\tau_v \leftarrow \tau_v + \theta_{v'}$;
  $\kappa_v^\star \leftarrow \left[1 + \left(\frac{1-\alpha}{\phi_v}\tau_v\right)^{\frac{1}{\alpha}}\right]^{-1}$; $\theta_v \leftarrow (\kappa_v^\star)^{1-\alpha}\frac{\phi_v}{1-\alpha} + (1-\kappa_v^\star)^{1-\alpha}\,\tau_v$;

`/* Dynamic Programming: descending phase */`

$\gamma_1^\star \leftarrow \kappa_1^\star$;$h_1 \leftarrow 1 - \kappa_1^\star$;

for $v = 2, \ldots, |V|$ **do**
  $v' \leftarrow$ father of v;
  $\gamma_v^\star \leftarrow \kappa_v^\star h_{v'}$;
  $h_v \leftarrow h_{v'}(1 - \kappa_v^\star)$;

Result: α-fair allocation $\boldsymbol{\delta}^\star, \boldsymbol{\gamma}^\star$

While the proportion of time each beam is activated does not depend on the flow rates r_k, $k \in \mathcal{K}$, the dependence in the number of flows served by each beam $\mathbf{n} = (n_v)_{v \in V}$ is not completely obvious.

Maximal Throughput ($\alpha = 0$) For simplicity, assume that all flows allocated to the same beam have the same data rate so that $r_k = r_{k'}$ if $v_k = v_{k'}$. Then we obtain:

$$\delta_k^\star = \frac{1}{n_{v_k}} \ , \ k \in \mathcal{K}$$
$$\gamma_v^\star = \prod_{v' \in \mathcal{D}(v)} \mathbf{1}\{n_{v'} = 0\} \ , \ v \in V.$$

Hence a beam is activated all the time if all of its descendants are empty, and never activated otherwise. In short, Max Throughput gives absolute priority to beams with higher depth.

Figure 3.5 illustrates the proportion of time each beam is activated by the α-fair allocation on two simple examples. For simplicity we consider G to be a binary tree of height 2 and we assume that the data rate of each flow is unity, so that $r_k = 1$, $k \in \mathcal{K}$. We can see that lower values of α tend to prioritize the leaves over the root, while higher values of α allocate more ressources to the root. Indeed, in the extreme case of $\alpha = 0$, the root is active if and only if all other nodes are empty. Also, as seen on Example 1, the Proportional Fair allocation does not treat the root and the leaves equally if all of them have the same number of flows. This is due to the fact that activating the root prevents all other nodes from being activated.

3.3.8 Practical implementation

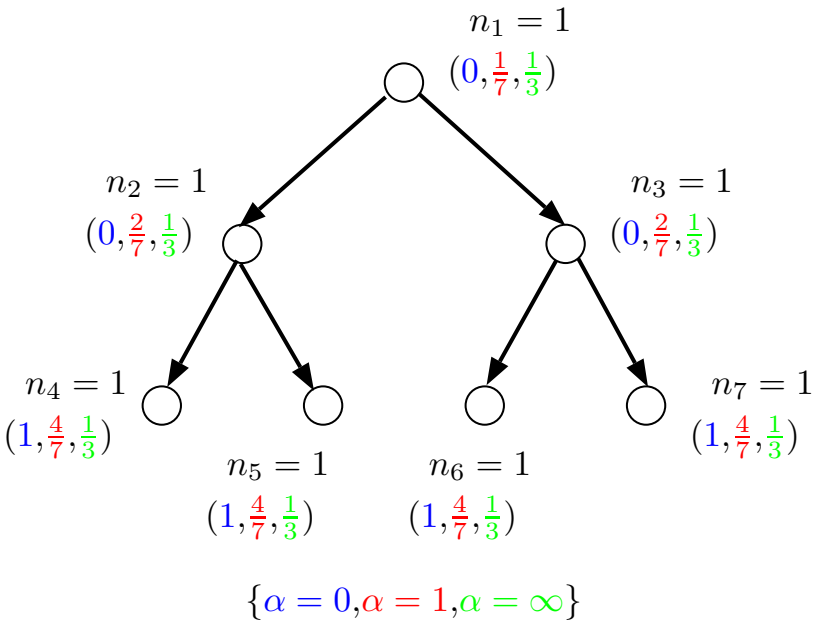
At first sight, an algorithm which simply returns $\boldsymbol{\gamma}^\star \in \mathbf{conv}(\mathcal{Z})$ does not seem to be usable directly, since as said before, in practice time is slotted, and at each time slot one must select an admissible beam configuration from $\mathcal{Z}$, so that the average proportion of time v (over a large number of time slots) is activated equals $\gamma_v^\star$. In fact one may do so directly using a simple randomized algorithm. Given $\boldsymbol{\kappa}^\star$ calculated by our algorithm, at each time slot, draw $\mathbf{Y} = (Y_v)_{v \in V}$ independent Bernoulli random variables where $\mathbb{E}(\mathbf{Y}) = \boldsymbol{\kappa}^\star$. Then define $\mathbf{Z} = (Z_v)_{v \in V} \in \{0,1\}^{|V|}$ as:

$$Z_v = Y_v \prod_{v' \in \mathcal{A}(v)} (1 - Y_{v'}), v \in V$$

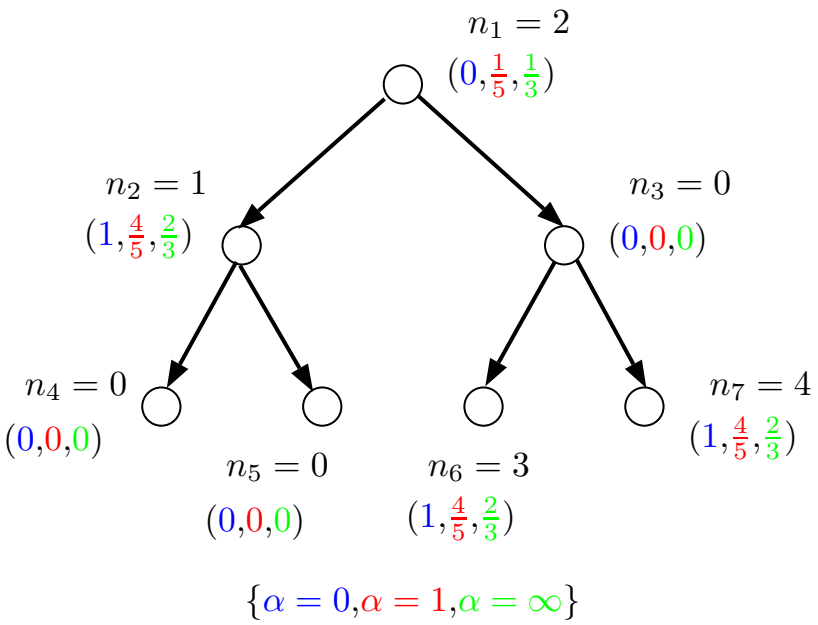
and activate beam v if and only if $Z_v = 1$. One may readily check that $\mathbf{Z} \in \mathcal{Z}$ since $Z_v Z_{v'} = 0$ whenever v is an ancestor or a descendant of v', and furthermore:

$$\mathbb{E}(Z_v) = \kappa_v^\star \prod_{v' \in \mathcal{A}(v)} (1 - \kappa_{v'}^\star) = \gamma_v^\star, v \in V.$$

Hence $\mathbb{E}(\mathbf{Z}) = \boldsymbol{\gamma}^\star$, and repeating the above procedure for each time slot provides a practical implementation of the α-fair allocation, and, as a bonus, involves calculating $\boldsymbol{\gamma}^\star$ only once.



(a) Example 1



(b) Example 2

Figure 3.5: Fair rate allocation

3.4 Flow Level Performance for Elastic Traffic

We now study the flow-level performance of hierarchical beamforming for elastic traffic. Flows arrive at random time instants, download a file of exponentially distributed size and depart upon download completion. For each configuration of flows $\mathbf{n} \in \mathbb{N}^{|V|}$, a given rate allocation scheme chooses a rate allocation $\boldsymbol{\delta}^\star(\mathbf{n}), \boldsymbol{\gamma}^\star(\mathbf{n})$, from which the throughput of each flow present in the system can be deduced. As done usually when studying the flow-level performance of wireless networks, we assume that the rate allocation mechanism adapts instantaneously upon arrival and departure of flows (separation of time scales), so that the throughput of any given flow at any given time is equal to the throughput given by the rate allocation scheme. Our aim is to calculate the throughput experienced by a typical flow.

3.4.1 Model

We assume that flows arrive in each region $\mathcal{G}_v$ according to a Poisson process with rate λ_v. Each flow represents an amount of data whose size is exponentially distributed with expectation 1. To simplify the analysis, we assume that any flow associated with beam v has data rate r_v (this can be relaxed in cases of interest, as shown below). We denote by $\mathbf{N}(t) = (N_v(t))_{v \in V}$ the state of the system at time t, which is a random variable, where $N_v(t)$ denotes the number of active flows present in $\mathcal{G}_v$ at time t. Define the load for each beam:

$$\rho_v = \frac{\lambda_v}{r_v}$$

It is noted that $\mathbf{N}(t)$ is a continuous-time Markov process. We say that the system is stable if and only if $\mathbf{N}(t)$ is positive recurrent. When the system is stable, Little law guarantees that the expected amount of time a typical flow is served by beam v equals $\frac{\mathbb{E}(N_v(t))}{\lambda_v}$. We then define the flow throughput ([11]):

$$\psi_v = \frac{\lambda_v}{\mathbb{E}(N_v(t))},$$

which represents the expected throughput experienced by a typical flow served by beam v.

3.4.2 Transition rates

The transition rates of $\mathbf{N}(t)$ depend on the rate allocation mechanism. In state $\mathbf{N}(t)$, the proportion of time beam v is activated equals $\gamma_v^\star(\mathbf{N}(t))$, so that the throughput of any flow served by beam v equals

$$r_v \gamma_v^\star(\mathbf{N}(t)) \delta_v(\mathbf{N}(t)) = \frac{r_v \gamma_v^\star(\mathbf{N}(t))}{N_v(t)}.$$

Denote by:

$$\mu(\mathbf{n} \to \mathbf{n}') = \frac{d}{dt}\mathbb{P}(\mathbf{N}(t+dt) = \mathbf{n}' | \mathbf{N}(t) = \mathbf{n})$$

the transition rate of the Markov process $\mathbf{N}(t)$ between states $\mathbf{n}$ and $\mathbf{n}'$. Denote by $\mathbf{e}^v = (0, \ldots, 0, 1, 0, \ldots, 0)$ the v th-canonical base vector of $\mathbb{R}^{|V|}$. The only possible transitions from $\mathbf{n}$ are $\mathbf{n} \to \mathbf{n} + \mathbf{e}^v$ and $\mathbf{n} \to \mathbf{n} - \mathbf{e}^v$, $v \in V$, which correspond to an arrival and a departure of a flow served by beam v respectively. The transition rates are:

$$\mu(\mathbf{n} \to \mathbf{n} + \mathbf{e}^v) = \lambda_v,$$
$$\mu(\mathbf{n} \to \mathbf{n} - \mathbf{e}^v) = r_v \gamma_v^\star(\mathbf{n}) \mathbf{1}\{n_v \geq 1\}.$$

3.4.3 Stability region

The stability of the system depends on the rate allocation mechanism which maps the system state $N(t)$ into the corresponding rate allocation $\boldsymbol{\delta}(\mathbf{N}(t)), \boldsymbol{\gamma}(\mathbf{N}(t))$. We may derive the stability region from our earlier analysis of fair scheduling.

Proposition 2. *There exists a rate allocation scheme ensuring stability if and only if* $\sum_{v' \in \bar{\mathcal{A}}(v)} \rho_{v'} < 1$ *for all* $v \in V$.

Proof. There exists a rate allocation scheme ensuring stability if and only if $\boldsymbol{\rho}$ lies in the interior of $\mathbf{conv}(\mathcal{Z})$. Furthermore, from Proposition 1

$$\mathbf{conv}(\mathcal{Z}) = \left\{ \boldsymbol{\gamma} \in [0,1]^{|V|} : \sum_{v' \in \bar{\mathcal{A}}(v)} \gamma_{v'} \leq 1, v \in V \right\}$$

which concludes the proof. □

3.5 Flow level performance under proportional fairness

3.5.1 PF allocation

We now compute the stationary distribution of $\mathbf{N}(t)$ for Proportional Fair rate allocation which is the α-fair rate allocation scheme described in the previous section with $\alpha = 1$. We recall that the PF allocation is given by:

$$\gamma_v^\star(\mathbf{n}) = \frac{n_v}{\sum_{v' \in \bar{\mathcal{D}}(v)} n_{v'}} \prod_{v'' \in \mathcal{A}(c)} \frac{\sum_{c' \in \mathcal{D}(v'')} n_{v'}}{\sum_{v' \in \bar{\mathcal{D}}(v'')} n_{v'}}$$
$$\delta_v^\star(\mathbf{n}) = \frac{1}{n_v}.$$

Stationary Distribution and Flow Throughput Both the stationary distribution and the flow throughput are given by Theorem 4. The proof of Theorem 4 is based on the fact that $\mathbf{N}(t)$ is reversible, so that its stationary distribution is known up to a normalization constant, and then calculating this constant by recursion. It should be noted that, in general, the PF allocation does not lead to reversible systems. It is also noted that the result holds not only for exponential flow size distributions, but in fact holds for all flow sizes distributions i.e. the system is insensitive.

Theorem 4. *Consider PF rate allocation. Then* $\mathbf{N}(t)$ *is a reversible Markov process with stationary distribution:*

$$\pi(\mathbf{n}) = \frac{1}{c(\boldsymbol{\rho})} \prod_{v \in V} \rho_v^{n_v} \binom{\sum_{v' \in \bar{\mathcal{D}}(v)} n_{v'}}{n_v}$$

with:

$$c(\boldsymbol{\rho}) = \prod_{v \in V} \frac{1 - \sum_{v' \in \mathcal{A}(v)} \rho_{v'}}{1 - \sum_{v' \in \bar{\mathcal{A}}(v)} \rho_{v'}}.$$

The expected number of customers in stationary state is:

$$\mathbb{E}[N_v(t)] = \rho_v \sum_{v' \in \bar{\mathcal{D}}(v)} \frac{1 - d(v')}{1 - \sum_{v'' \in \bar{\mathcal{A}}(v')} \rho_{v''}}$$

and the flow throughput is:

$$\psi_v = r_v \left[\sum_{v' \in \bar{\mathcal{D}}(v)} \frac{1 - d(v')}{1 - \sum_{v'' \in \bar{\mathcal{A}}(v')} \rho_{v''}} \right]^{-1}$$

Furthermore, the above holds for all flow time distributions, i.e. the system is insensitive.

3.5.2 Proof of Theorem 4

To prove that $\mathbf{N}(t)$ is both reversible and has stationary distribution π, it is sufficient to check that the detailed balance condition holds:

$$\pi(\mathbf{n})\mu(\mathbf{n} \to \mathbf{n} + \mathbf{e}^v) = \pi(\mathbf{n} + \mathbf{e}^v)\mu(\mathbf{n} + \mathbf{e}^v \to \mathbf{n})$$

For all $\mathbf{n} \in \mathbb{N}^{|V|}$ and $v \in V$. Detailed balance holds by inspection since:

$$\frac{\mu(\mathbf{n} \to \mathbf{n} + \mathbf{e}^v)}{\mu(\mathbf{n} + \mathbf{e}^v \to \mathbf{n})} = \frac{\rho_v}{\gamma_v^\star(\mathbf{n} + \mathbf{e}^v)} = \frac{\pi(\mathbf{n} + \mathbf{e}^v)}{\pi(\mathbf{n})}.$$

Hence $\mathbf{N}(t)$ has stationary distribution π and the value of $c(\boldsymbol{\rho})$ can be found by normalization since $\sum_{\mathbf{n} \in \mathbb{N}^{|V|}} \pi(\mathbf{n}) = 1$:

$$c(\boldsymbol{\rho}) = \sum_{\mathbf{n} \in \mathbb{N}^{|V|}} \prod_{v \in V} \rho_v^{n_v} \binom{\sum_{v' \in \bar{\mathcal{D}}(v)} n_{v'}}{n_v}.$$

We proceed by recursion, and for $v \in V$, we define $c^v(\boldsymbol{\rho})$ the value of $c(\boldsymbol{\rho})$ when $\rho_{v'} = 0$ for all $v' \notin \bar{\mathcal{D}}_v$, i.e. $c^v(\boldsymbol{\rho})$ is the value of the above sum only considering $\mathbf{n} \in \mathbb{N}$ with $n_{v'} = 0$ unless $v' \notin \bar{\mathcal{D}}_v$. Another interpretation is that $c^v(\boldsymbol{\rho})$ is the value of $c(\boldsymbol{\rho})$ when considering the subgraph made of v and its descendants. Finally it is also noted that $c(\boldsymbol{\rho}) = c^1(\boldsymbol{\rho})$ since 1 is the root. We use the following fact for binomial coefficients:

Fact 1. *For $z, \beta \in \mathbb{C}$, we have $\frac{1}{(1-z)^{\beta+1}} = \sum_{k\geq 0} \binom{k+\beta}{k} z^k$.*

Summing over $n_1 \in \mathbb{N}$ the above expression we obtain the following relation:

$$c(\boldsymbol{\rho}) = c^1(\boldsymbol{\rho}) = \frac{1}{1-\rho_1} \prod_{v:(1,v)\in E} c^v\left(\frac{\boldsymbol{\rho}}{1-\rho_1}\right).$$

More generally we have the following recursive relation:

$$c^v(\boldsymbol{\rho}) = \frac{1}{1-\rho_v} \prod_{v':(v,v')\in E} c^{v'}\left(\frac{\boldsymbol{\rho}}{1-\rho_v}\right).$$

Iterating the relation above starting at $v = 1$ we obtain the announced result:

$$c(\boldsymbol{\rho}) = \prod_{v\in V} \frac{1 - \sum_{v'\in\mathcal{A}(v)} \rho_{v'}}{1 - \sum_{v'\in\bar{\mathcal{A}}(v)} \rho_{v'}}.$$

The expected number of customers is calculated using the following trick. Recall the definition of $c(\rho)$:

$$c(\boldsymbol{\rho}) = \sum_{\mathbf{n}\in\mathbb{N}^{|V|}} \prod_{v'\in V} \rho_{v'}^{n_{v'}} \binom{\sum_{v''\in\bar{\mathcal{D}}(v')} n_{v''}}{n_{v'}}$$

Taking logarithms and differentiating with respect to ρ_v:

$$\begin{aligned}\rho_v \frac{\partial \ln c(\rho)}{\partial \rho_v} &= \frac{1}{c(\rho)} \sum_{n\in\mathbb{N}^{|V|}} n_v \prod_{v'\in V} \rho_{v'}^{n_{v'}} \binom{\sum_{v''\in\bar{\mathcal{D}}(v')} n_{v''}}{n_{v'}} \\ &= \sum_{n\in\mathbb{N}^{|V|}} n_v \pi(\mathbf{n}) = \mathbb{E}(N_v(t)).\end{aligned}$$

Plugging the previous expression of $c(\rho)$ gives:

$$\begin{aligned}\mathbb{E}(N_v(t)) &= \rho_v \frac{\partial \ln c(\rho)}{\partial \rho_v} \\ &= \rho_v \frac{\partial}{\partial \rho_v} \sum_{v'\in V} \ln\left(\frac{1 - \sum_{v''\in\mathcal{A}(v')} \rho_{v''}}{1 - \sum_{v''\in\bar{\mathcal{A}}(v')} \rho_{v''}}\right) \\ &= \rho_v \sum_{v'\in\bar{\mathcal{D}}(v)} \frac{1 - d(v')}{1 - \sum_{v''\in\bar{\mathcal{A}}(v')} \rho_{v''}}.\end{aligned}$$

as announced.

3.6 Flow level performance under maximal throughput

We now study the flow-level performance for the Maximal Throughput rate allocation which is the α-fair rate allocation scheme with $\alpha = 0$. As we can see, while in this case $\mathbf{N}(t)$ is not reversible (nor insensitive), it has a hiearchical structure which can be exploited in order to provide tractable expressions for performance.

Transition Rates We first calculate the transition rates. From the hierarchical nature of the codebook, we have that $r_{v'} > r_v$ if $v' \in D(v)$ (see Definition 1). As stated in the previous section, in state $\mathbf{n} \in \mathbb{N}^{|V|}$, the proportion of time beam v is activated equals:

$$\gamma_v^\star(\mathbf{n}) = \prod_{v' \in \mathcal{D}(v)} \mathbf{1}\{n_{v'} = 0\}.$$

namely, beam v is activated all the time if all of its descendants are empty, and never activated otherwise.

3.6.1 Performance for line graphs

We first consider the case of line graphs, where we have $E = \{(1,2),(2,3),\dots,(|V|-1,|V|)\}$. From definition 1, beam v is activated at time t if and only if all of its descendants are empty, namely $n_{v'}(t) = 0$ $v' = v+1,\dots,|V|$. From this observation, we can reduce the system to a M/M/1 queue with preemptive-resume priority as follows. Indeed, consider a single server with $|V|$ classes of users, where class v represents users served by beam v. In this system, at time t, users of class v are served at rate r_v if $n_{v'}(t) = 0$, $v' = v+1,\dots,|V|$ and at rate 0 otherwise. This new system is equivalent to the original system, which yields Theorem 5. The proof is provided in appendix 3.9.2.

Theorem 5. *Consider MT allocation and G a line graph. Then $\mathbf{N}(t)$ is positive recurrent (stable) if and only if $\sum_{v\in V} \rho_v < 1$. Furthermore, if $\sum_{v\in V} \rho_v < 1$ we have for all $v \in V$:*

$$\mathbb{E}(N_v(t)) = \frac{\rho_v \left(1 + \sum_{v' \geq v} \rho_{v'} \left(\frac{r_v}{r_{v'}} - 1\right)\right)}{\left(1 - \sum_{v' \geq v} \rho_{v'}\right)\left(1 - \sum_{v' > v} \rho_{v'}\right)}$$

and the flow throughput is :

$$\psi_v = \frac{r_v(1 - \sum_{v' \geq v} \rho_{v'})(1 - \sum_{v' > v} \rho_{v'})}{1 + \sum_{v' \geq v} \rho_{v'}(\frac{r_v}{r_{v'}} - 1)}$$

3.6.2 Performance for generic graphs

We now consider the case where G is a generic tree. In that case as well we compute the flow throughput and the analysis is much more involved than for the previous case.

Under the MT allocation, a beam is activated if and only if all of its descendants are empty. However the state of a beam does not depend on the state of its ancestors. Therefore, the evolution of $\mathbf{N}_v(t)$ depends on $\mathbf{N}_{v'}(t)$ if and only if $v' \in \bar{\mathcal{D}}(v)$. In order to study the distribution of the state of v, i.e. $N_v(t)$, it is natural to study the process $(N_v(t))_{v' \in \bar{\mathcal{D}}(v)}$ which describes its state and that of its descendants. We define the busy period of this process which plays a crucial role in our analysis. We define $\ell_v = \sum_{v' \in \mathcal{D}(v)} \lambda_{v'}$ the total arrival rate in the descendants of v.

Definition 2. *For $v \in V$ define the Markov processes*

$$\mathbf{N}^v(t) = (N_{v'}(t))_{v' \in \bar{\mathcal{D}}(v)} \in \mathbb{N}^{|\bar{\mathcal{D}}(v)|},$$
$$\mathbf{M}^v(t) = (N_{v'}(t))_{v' \in \mathcal{D}(v)} \in \mathbb{N}^{|\mathcal{D}(v)|},$$

where $\mathbf{N}^v(t)$ describes the state of v and its descendants, and $\mathbf{M}^v(t)$ describes the state of the descendants of v. Define the random variable B_v which is the busy period of process $\mathbf{M}^v(t)$. We recall that the busy period is $B_v = T_2 - T_1$ in a system where $\mathbf{M}^v(0) = \mathbf{0}$, and $T_1 > 0$ is the first random instant after 0 such that $\mathbf{M}^v(T_1) \neq \mathbf{0}$, and $T_2 > T_1$ is the first random instant after T_1 such that $\mathbf{M}^v(T_2) = \mathbf{0}$.

In Theorem 6, we show that the expected number of customers in stationary state can be computed as a function of the two first moments of the busy period.

Theorem 6. *Under MT rate allocation, the expected number of flows in beam v satisfies, for all $v \in V$:*

$$\mathbb{E}(N_v(t)) = \frac{\frac{\lambda_v \ell_v}{2}\mathbb{E}(B_v^2) + \rho_v \left(1 + \mathbb{E}(B_v)\ell_v\right)^2}{(1 - \rho_v(1 + \mathbb{E}(B_v)\ell_v))(1 + \mathbb{E}(B_v)\ell_v)}$$

Furthermore, $\mathbb{E}(B_v)$ can be computed recursively using the following relations:

$$\mathbb{E}(B_v) = \frac{1}{\ell_v}\left(\frac{1}{\prod_{v':(v,v')\in E}\mathbb{P}(\mathbf{N}^{v'}(t) = 0)} - 1\right).$$

and:

$$\mathbb{P}(\mathbf{N}^v(t) = 0) + \rho_v = \prod_{v':(v,v')\in E} \mathbb{P}(\mathbf{N}^v(t) = 0), \forall v \in V.$$

3.6.3 Proof of Theorem 6

Expected number of customers. We say that a customer is waiting if it has never received service, and it is in service if it has received service but has not left the system yet. We compute the expected time spent in service, denoted by $\mathbb{E}(M_v)$. When a flow first receives service by beam v, by definition, no flows from classes $v' \in \mathcal{D}(v)$ must be present in the system. Once she has started to receive service, there are two possible outcomes: either no flows from classes $v' \in \mathcal{D}(v)$ arrive before her service completion

(this occurs with probability $\frac{r_v}{r_v+\ell_v}$), otherwise she must wait for the duration of a busy period of $\mathbf{M}^v(t)$ and we are back in the initial situation, since the exponential distribution is memoryless (this occurs with probability $\frac{\ell_v}{r_v+\ell_v}$). The expected time before either of these events happens is $\frac{1}{r_v+\ell_v}$. Therefore the expected service time verifies:

$$\mathbb{E}(M_v) = \frac{r_v}{r_v+\ell_v}\frac{1}{r_v} + \frac{\ell_v}{r_v+\ell_v}(\mathbb{E}(B_v)+\mathbb{E}(M_v))$$

so that:

$$\mathbb{E}(M_v) = \frac{1}{r_v}(1+\mathbb{E}(B_v)\ell_v)$$

We now calculate the expected time spent waiting denoted by $\mathbb{E}(W_v)$. Assume that a flow arrives at time 0. From the PASTA property, $N_v(0)$ is distributed as the stationary distribution. Denote by T_v^- the first instant at which $\mathbf{M}^v(T_v^-)=0$. Denote by $T_v^- + T_v^+$ the first instant at which all customers present at time 0 have been served. Therefore the waiting time may be decomposed as:

$$\mathbb{E}(W_v) = \mathbb{E}(T_v^-) + \mathbb{E}(T_v^+).$$

First one notices that T_v^- is expressed as a function of the residual busy period of the process $\mathbf{M}^v(t)$ and is given by:

$$\mathbb{E}(T_v^-) = \frac{1}{2}\frac{\mathbb{E}(B_v^2)}{\mathbb{E}(B_v)+\mathbb{E}(A_v)} = \frac{1}{2}\frac{\mathbb{E}(B_v^2)}{\mathbb{E}(B_v)+\ell_v^{-1}}.$$

where A_v is the time that the process $\mathbf{M}^v(t)$ spends in state $\mathbf{0}$. Indeed A_v is exponentially distributed with mean $\frac{1}{\ell_v}$.

From Little law, $\mathbb{E}(N_v(0)) = \lambda_v(\mathbb{E}(W_v)+\mathbb{E}(M_v))$. No flow of beam v may receive service before T_v^-, therefore T_v^+ is the amount of time necessary to serve $N_v(0)$ flows and is given by:

$$\mathbb{E}(T_v^+) = \mathbb{E}(N_v(0))\mathbb{E}(M_v) = \lambda_v\mathbb{E}(M_v)(\mathbb{E}(W_v)+\mathbb{E}(M_v)).$$

using the previous relation. Substituting we obtain:

$$\mathbb{E}(W_v) = \mathbb{E}(T_v^-) + \mathbb{E}(T_v^+) = \mathbb{E}(T_v^-) + \lambda_v\mathbb{E}(M_v)(\mathbb{E}(W_v)+\mathbb{E}(M_v)),$$

which yields:

$$\mathbb{E}(W_v) = \frac{\mathbb{E}(T_v^-) + \lambda_v\mathbb{E}(M_v)^2}{1-\lambda_v\mathbb{E}(M_v)}$$

Therefore the total expected time spent by a typical flow is given by:

$$\mathbb{E}(W_v) + \mathbb{E}(M_v) = \frac{\mathbb{E}(T_v^-) + \mathbb{E}(M_v)}{1-\lambda_v\mathbb{E}(M_v)}$$

Substituting the value of $\mathbb{E}(M_v)$ we get

$$\begin{aligned}\mathbb{E}(W_v)+\mathbb{E}(M_v) &= \frac{\frac{1}{2}\frac{\mathbb{E}(B_v^2)}{\mathbb{E}(B_v)+\ell_v^{-1}}+\frac{1}{r_v}\left(1+\mathbb{E}(B_v)\ell_v\right)}{1-\rho_v(1+\mathbb{E}(B_v)\ell_v)}\\ &= \frac{\frac{\ell_v}{2}\mathbb{E}(B_v^2)+\frac{1}{r_v}\left(1+\mathbb{E}(B_v)\ell_v\right)^2}{(1-\rho_v(1+\mathbb{E}(B_v)\ell_v))(1+\mathbb{E}(B_v)\ell_v)}.\end{aligned}$$

Using Little law $\mathbb{E}(N_v(t)) = \lambda_v(\mathbb{E}(W_v)+\mathbb{E}(M_v))$ yields the announced result.

Expected duration of the busy periods. Consider the Markov process $\mathbf{M}^v(t)$. The expected duration this process spends in state $\mathbf{0}$ is $\frac{1}{\ell_v}$, and the duration between two visits to state $\mathbf{0}$ equals $\mathbb{E}(B_v)$. Therefore, the probability that $\mathbf{M}^v(t)$ is in state $\mathbf{0}$ satisfies:

$$\mathbb{P}(\mathbf{M}^v(t)=0) = \frac{\frac{1}{\ell_v}}{\frac{1}{\ell_v}+\mathbb{E}(B_v)},$$

so that the expected busy period is:

$$\mathbb{E}(B_v) = \frac{1}{\ell_v}\left(\frac{1}{\mathbb{P}(\mathbf{M}^v(t)=0)}-1\right).$$

Furthermore, using the fact that, if v' and v'' are distinct children of v, processes $\mathbf{N}^{v'}(t)$ and $\mathbf{N}^{v''}(t)$ are independent we get:

$$\begin{aligned}\mathbb{P}(\mathbf{M}^v(t)=0) &= \mathbb{P}(\mathbf{N}^{v'}(t)=0, \forall v' : (v,v')\in E)\\ &= \prod_{v':(v,v')\in E}\mathbb{P}(\mathbf{N}^{v'}(t)=0).\end{aligned}$$

Substituting yields the second claim:

$$\mathbb{E}(B_v) = \frac{1}{\ell_v}\left(\frac{1}{\prod_{v':(v,v')\in E}\mathbb{P}(\mathbf{N}^{v'}(t)=0)}-1\right).$$

Void probabilities To complete the proof, we must compute the void probabilities $\mathbb{P}(\mathbf{N}^v(t)=0)$ for $v\in V$. We have:

$$\mathbb{P}(\mathbf{M}^v(t)=0) = \mathbb{P}(\mathbf{M}^v(t)=0, N_v(t)\neq 0)+\mathbb{P}(\mathbf{M}^v(t)=0, N_v(t)=0)$$

As explained before, $\mathbb{P}(\mathbf{M}^v(t)=0) = \prod_{v':(v,v')\in E}\mathbb{P}(\mathbf{N}^{v'}(t)=0)$ from independence, and by definition $\mathbb{P}(\mathbf{M}^v(t)=0, N_v(t)=0) = \mathbb{P}(\mathbf{N}^v(t)=0)$. Finally, beam v serves its users if and only if $\mathbb{P}(\mathbf{M}^v(t)=0, N_v(t)\neq 0)$, so that, by conservation of work we must have $\lambda_v = r_v\mathbb{P}(\mathbf{M}^v(t)=0, N_v(t)\neq 0)$. Substituting yields the announced relation:

$$\prod_{v':(v,v')\in E}\mathbb{P}(\mathbf{N}^{v'}(t)=0) = \rho_v+\mathbb{P}(\mathbf{N}(t)^v=0)$$

and from this relation, the value of $\mathbb{P}(\mathbf{N}^v(t)=0)$ may be computed by backwards induction for all $v\in V$ which concludes our proof.

3.6.4 Busy periods: exponential approximation

As shown by Theorem 6, the only missing piece in order to compute $\mathbb{E}(N_v(t))$ is the value of the second moment of the cycle times $\mathbb{E}(B_v^2)$. In general, computing the second moment of the busy period does not seem completely straightforward. We propose to approximate the busy period by an exponential, so that $\frac{\mathbb{E}(B_v^2)}{2} \approx \mathbb{E}(B_v)^2$. The flow throughput is given by corollary 1. The rationale for this approximation is as follows: if $\boldsymbol{\rho} \ll \mathbf{1}$, then the busy period is simply the service time of the first customer which enters the system, and this time indeed has exponential distribution. As shown in our numerical experiments, this approximation yields tractable formulas which are rather accurate.

Corollary 1. *Assume that the busy period can be approximated by an exponential. Then, under MT rate allocation, the number expected number of flows in beam v satisfies, for all $v \in V$:*

$$\mathbb{E}(N_v(t)) = \frac{\lambda_v \ell_v \mathbb{E}(B_v)^2 + \rho_v \left(1 + \mathbb{E}(B_v)\ell_v\right)^2}{(1 - \rho_v(1 + \mathbb{E}(B_v)\ell_v))(1 + \mathbb{E}(B_v)\ell_v)}$$

3.7 Flow Level Performance for Streaming Traffic

3.7.1 Model

We now consider streaming traffic where flows require a fixed amount of resources throughout their stay in the system. Namely the available resources are spit into an integer number $\xi \geq 1$ of circuits, where a circuit represents the smallest unit of resource that can be allocated to a flow. Each flow served by beam $v \in V$ requires a number of circuits $s_v \in \{1, \ldots, \xi\}$, and to ensure that that all flows are allocated enough resources admission control is used. When a new flow arrives, one checks whether or not one can guarantee that all active flows can be allocated enough circuits. If the answer is yes, the flow is admitted, otherwise she is blocked. Flows arrive in beam v according to a Poisson process with rate λ_v and remain there during an exponentially distributed time with mean $\frac{1}{r_v}$. Define the load $\rho_v = \frac{\lambda_v}{r_v}$. In fact the system is insensitive, so that considering exponential service times is sufficient. The state of the system at time t is $\mathbf{N}(t) \in \mathcal{N} \subset \mathbb{N}^{|V|}$ where $\mathbf{N}_v(t)$ represents the number of flows served by beam v and $\mathcal{N}$ is the set of states in which all active flows can be allocated enough circuits. The only possible transitions of $N(t)$ correspond to the arrival and departure of a flow, and the transition rates are for all $\mathbf{n} \in \mathcal{N}$:

$$\mu(\mathbf{n} \to \mathbf{n} + \mathbf{e}^v) = \lambda_v \mathbf{1}\{\mathbf{n} + \mathbf{e}^v \in \mathcal{N}\},$$
$$\mu(\mathbf{n} \to \mathbf{n} - \mathbf{e}^v) = r_v \mathbf{1}\{n_v \geq 1\}.$$

3.7.2 Admission control

Let us now characterize the set of admissible states allowed by admission control i.e. states in which all flows can be allocated enough circuits by some policy. Consider the system in state $\mathbf{n} \in \mathbb{N}^{|V|}$, and assume that there exists a rate allocation policy $\boldsymbol{\gamma}^\star(\mathbf{n}) \in \mathbf{conv}(\mathcal{Z})$ which ensures that all flows are allocated enough circuits. Flows served by beam v require s_v circuits so that the proportion of time v is activated should satisfy $\boldsymbol{\gamma}^\star_v(\mathbf{n}) = n_v \frac{s_v}{\xi}$ for all v. Furthermore, since any feasible rate allocation policy must satisfy $\boldsymbol{\gamma}(\mathbf{n}) \in \mathbf{conv}(\mathcal{Z})$. Clearly, both of those conditions can be satisfied if and only if:

$$\sum_{v' \in \bar{\mathcal{A}}(v)} n_{v'} \frac{s_{v'}}{\xi} \leq 1, v \in V.$$

The set of admissible states $\mathcal{N}$ is therefore:

$$\mathcal{N} = \left\{ \mathbf{n} \in \mathbb{N}^{|V|} : \sum_{v' \in \bar{\mathcal{A}}(v)} n_{v'} s_{v'} \leq \xi \, , \, v \in V \right\}.$$

Furthermore, if the system is in state $\mathbf{n} \in \mathcal{N}$, and a flow associated with beam v arrives, it is admitted if and only if his entering the system leads to an admissible state, and blocked otherwise. Define the set of blocking states for beam v:

$$\mathcal{N}(v) = \{\mathbf{n} \in \mathbb{N}^{|V|} : \mathbf{n} \in \mathcal{N}, \mathbf{n} + e^v \notin \mathcal{N}\}.$$

It is noted that a flow associated with beam v will be blocked if and only if it arrives in a blocking state for beam v, that is $\mathbf{n} \in \mathcal{N}(v)$.

3.7.3 Stationary distribution and blocking probability

We may now derive the stationary distribution of the system in closed form, as well as the blocking probability for each beam, which constitutes the performance figure of the system. The stationary distribution is known in closed form up to a normalization constant, so that the main difficulty is to compute a sum over the state space $\mathcal{N}$. Clearly, brute force summation is not feasible as $|\mathcal{N}|$ grows exponentially with $|V|$. We show that there exists a low complexity algorithm (stated as Algorithm 3) to compute the stationary distribution as well as blocking probabilities. The result is stated as Theorem 7 and proven in subsection 3.7.4.

It is noted that Algorithm 3 computes the blocking probabilities in time $\mathcal{O}(\xi|V|h(G))$. In most cases of interest, for instance if G is a regular tree, $h(G) = \mathcal{O}(\ln|V|)$, so that the running time of Algorithm 3 is $\mathcal{O}(\xi|V|\ln|V|)$ which is linear in the number of circuits ξ and almost linear in the number of beams $|V|$. Since its input has size $\mathcal{O}(|V|)$, we deduce that the dependency of the running time of this algorithm on $|V|$ is optimal up to a logarithmic factor $\ln|V|$. This algorithm is reminiscent of the Kaufman-Roberts

algorithm [29, 48] used to compute blocking probabilities in multi-rate Erlang systems, and leverages the hierchical structure of the codebook by computing the blocking probabilities recursively.

Theorem 7. *(i) Under the above assumptions,* $\mathbf{N}(t)$ *is a reversible Markov process with stationary distribution* π*:*

$$\pi(\mathbf{n}) = \frac{1}{c(\boldsymbol{\rho})} \prod_{v \in V} \frac{\rho_v^{n_v}}{n_v!} \ , \ \mathbf{n} \in \mathcal{N},$$

where $c(\rho)$ *is the normalization constant defined as:*

$$c(\boldsymbol{\rho}) = \sum_{\mathbf{n} \in \mathcal{N}} \prod_{v \in V} \frac{\rho_v^{n_b}}{n_b!}.$$

(ii) The blocking probability of class v *is:*

$$p_v(\boldsymbol{\rho}) = \mathbb{P}(\mathbf{N}(t) \in \mathcal{N}(v)) = \frac{\sum_{n \in \mathcal{N}(v)} \prod_{v' \in V} \frac{\rho_{v'}^{n_{v'}}}{n_{v'}!}}{\sum_{n \in \mathcal{N}} \prod_{v' \in V} \frac{\rho_{v'}^{n_{v'}}}{n_{v'}!}}.$$

(iii) Algorithm 3 outputs of value of the blocking probability vector $\mathbf{p}(\boldsymbol{\rho})$ *using time* $\mathcal{O}(\xi |V| h(G))$ *and memory* $\mathcal{O}(|V|\xi)$*.*

3.7.4 Proof of Theorem 7

Stationary Distribution. By inspection, one may readily check that π verifies the detailed balance conditions:

$$\frac{\pi(\mathbf{n} + \mathbf{e}^v)}{\pi(\mathbf{n})} = \frac{\mu(\mathbf{n} \to \mathbf{n} + \mathbf{e}^v)}{\mu(\mathbf{n} + \mathbf{e}^v \to \mathbf{n})} = \frac{\lambda_v}{r_v} = \rho_v.$$

so that $\mathbf{N}(t)$ is indeed reversible with stationary distribution π.

Normalization Constant It now remains to compute the normalization constat $c(\boldsymbol{\rho})$. Furthermore, for $s = 0, 1, \ldots, \xi$, and $v \in V$ define $\mathcal{N}(s, v)$ the set of admissible sets when there are s circuits, and where beams which do not descend from v are empty:

$$\mathcal{M}(s,v) = \Big\{ \mathbf{n} \in \mathbb{N}^{|V|} : \sum_{v'' \in \bar{\mathcal{A}}(v')} n_{v''} s_{v''} \leq s, \forall v' \in V \\ \text{and } n_{v'} = 0, \forall v' \notin \bar{\mathcal{D}}(v) \Big\}.$$

and we define the corresponding partial sums:

$$c_v(s) = \sum_{\mathbf{n} \in \mathcal{M}(s,v)} \prod_{v' \in \bar{\mathcal{D}}(v)} \frac{\rho_v^{n_{v'}}}{n_{v'}!}.$$

Algorithm 3: Computation of blocking probabilities

Data: Tree G sorted by decreasing height, loads $\boldsymbol{\rho}$, circuit requirements $\mathbf{s}$, number of circuits ξ

for $v = 1, \ldots, |V|$ ***and*** $s = 1, \ldots, \xi$ **do** $c_v(s) \leftarrow 0$;

`/* Phase 1: Compute the normalization constant */`

for $v = |V|, \ldots, 1$ ***and*** $s = 1, \ldots, \xi$ **do**

  for $\ell = 0, \ldots, \lfloor s/s_v \rfloor$ **do**

    $t \leftarrow 1$;

    for $v' : (v, v') \in E$ **do** $t \leftarrow t c_{v'}(s - \ell s_v)$;

    $c_v(s) \leftarrow c_v(s) + \frac{\rho_v^\ell}{\ell!} t$;

  for $v' = |V|, \ldots, 1$ **do** $q_{v,v'}(s) \leftarrow c_{v'}(s)$;

`/* Phase 2: compute blocking probabilities */`

for $v = 1, \ldots, |V|$ ***and*** $s = 1, \ldots, \xi$ **do**

  $v' \leftarrow v$;

  if $s > s_v$ **do** $q_{v,v'}(s) \leftarrow c_v(s - s_v)$;

  while $v' \neq 1$ **do**

    $v' \leftarrow$ father of v';

    $q_{v,v'}(s) \leftarrow 0$;

    for $\ell = 0, \ldots, \lfloor s/s_{v'} \rfloor$ **do**

      $t \leftarrow 1$;

      for $v'' : (v', v'') \in E$ **do** $t \leftarrow t q_{v,v''}(s - \ell s_{v'})$;

      $q_{v,v'}(s) \leftarrow q_{v,v'}(s) + \frac{\rho_{v'}^\ell}{\ell!} t$;

  $p_v \leftarrow 1 - \frac{q_{v,1}(\xi)}{c_1(\xi)}$;

Result: Blocking probabilities $\mathbf{p}$

It is noted that $\mathcal{M}(\xi,1)=\mathcal{N}$, so that $c=c_1(\xi)$. The idea of the proposed algorithm is to compute the value of $c_v(s)$ for $v\in V$ and $s=0,\ldots,\xi$ by starting at the leaves and then using backward induction. Consider the subtree rooted at v and v' such that $(v,v')\in E$. If there are s circuits and beam v is serving n_v users, $s_v n_v$ circuits need to be allocated to beam v, so that the number of circuits available to all the beams in the subtree rooted at v' is $s-s_v n_v$. Therefore, $\mathbf{n}\in\mathcal{M}(s,v)$ if and only if it admits the following decompostion:

$$\mathbf{n}=n_v\mathbf{e}^v+\sum_{v':(v,v')\in E}\mathbf{m}^{v'}\text{ with }\mathbf{m}^{v'}\in\mathcal{M}(v',s-s_v n_v),$$

This fact gives rise to the following recursion, by summing over n_v in the definition of $c_v(s)$:

$$c_v(s)=\sum_{n_v=0}^{\lfloor\frac{s}{s_v}\rfloor}\frac{\rho_v^{n_v}}{n_v!}\prod_{v':(v,v')\in E}c_{v'}(s-n_v s_v).$$

The above relation readily gives an algorithm to compute $c_v(s)$, for all $v\in V$ and $s\leq\xi$ which consistutes the first part of our algorithm.

Blocking Probabilities As said before, the blocking probability p_v is the probability for a flow to arrive at a time where the system is in a blocking state $\mathbf{n}\in\mathcal{N}(v)$, so that the blocking rate for class v is:

$$p_v=\mathbb{P}(\mathbf{N}(t)\in\mathcal{N}(v))=\frac{\sum_{\mathbf{n}\in\mathcal{N}(v)}\prod_{v'\in V}\frac{\rho_{v'}^{n_{v'}}}{n_{v'}!}}{\sum_{\mathbf{n}\in\mathcal{N}}\prod_{v'\in V}\frac{\rho_{v'}^{n_{v'}}}{n_{v'}!}}.$$

We compute the blocking rate using a similar approach as that used above for the normalization constant. We have that $\mathbf{n}\in\mathcal{N}\setminus\mathcal{N}(v)$ if and only if $\mathbf{n}+\mathbf{e}^v\in\mathcal{N}$, which, by definition of $\mathcal{N}$ translates to:

$$\sum_{v'\in\bar{\mathcal{A}}(v'')}n_{v'}s_{v'}\leq\xi-s_v\mathbf{1}\{v\in\bar{\mathcal{A}}(v'')\}\ ,\ \forall v''\in V.$$

In order to compute a sum over $\mathcal{N}(v)$ recursively, similarly to the case of computing the normalization constant, define $\mathcal{Q}(s,v,v')$ the set of allowed states in a system where there are s circuits, where a user arriving at beam v does not cause blocking, and where all the beams which are not descendants of v' are empty:

$$\mathcal{Q}(s,v,v')=\Big\{\mathbf{n}\in\mathbb{N}^{|V|}:\sum_{v''\in\bar{\mathcal{A}}(v''')}n_{v''}s_{v''}\leq s-s_v\mathbf{1}\{v\in\bar{\mathcal{A}}(v''')\},$$
$$\forall v'''\in V\text{ and }n_{v''}=0,\forall v''\notin\bar{\mathcal{D}}(v')\Big\}.$$

and we define the partial sums:

$$q_{v,v'}(s) = \sum_{\mathbf{n}\in\mathcal{Q}(s,v,v')} \prod_{v''\in\bar{\mathcal{D}}(v')} \frac{\rho_{v''}^{n_{v''}}}{n_{v''}!}.$$

It is noted that since $\mathcal{M}(\xi, v, 1) = \mathcal{N} \setminus \mathcal{N}(v)$, we have:

$$q_{v,1}(\xi) = \sum_{\mathbf{n}\in\mathcal{N}\setminus\mathcal{N}(v)} \prod_{v'\in V} \frac{\rho_{v'}^{n_{v'}}}{n_{v'}!},$$

and

$$p_v = 1 - \frac{q_{v,1}(\xi)}{c(\boldsymbol{\rho})}.$$

so that computing the value of $q_{v,v'}(s)$ for all $v, v' \in V$ and $s \leq \xi$ yields the blocking probabilities. It is also noted that if v' and v are not descendants or ancestors of each other, the arrival of a user at beam v does not affect the number of circuits available to the subtree rooted at v', so that we simply have $q_{v,v'}(s) = c_{v'}(s)$, which we already have computed in the first phase of the algorithm. It is also noted that $q_{v,v}(s) = c_v(s - s_v)$. Now, similarly to the previous paragraph, $q_{v,v'}(s)$ obeys the following recursion:

$$q_{v,v'}(s) = \begin{cases} \sum_{n_{v'}=0}^{\lfloor \frac{s}{s_{v'}} \rfloor} \frac{\rho_{v'}^{n_{v'}}}{n_{v'}!} \prod_{v'':(v',v'')\in E} q_{v,v''}(s - n_{v'}s_{v'}) & \\ & \text{if } v' \in \mathcal{A}(v) \\ c_v(s - s_v) & \text{if } v' = v \\ c_{v'}(s) & \text{otherwise} \end{cases}$$

Complexity It is noted that the required memory for this algorithm to run is $\mathcal{O}(\xi|V|)$ since, as soon as p_v has been computed at the end of the loop of Phase 2, one can simply remove the values of $q_{v,v'}(s)$ from the memory. It is also noted that the time to compute $c_v(s)$ for $s \leq \xi$ and $v \in V$ is $\mathcal{O}(\xi|E|)$, and, since G is a tree, $|E| = |V| - 1$. So the time required for Phase 1 is $\mathcal{O}(\xi|V|)$. For Phase 2, for a given value of v, p_v is computed by inspecting all the ancestors of v, in time $\mathcal{O}(\xi|\bar{\mathcal{A}}(v)|) = \mathcal{O}(\xi h(G))$. Hence the time required for Phase 2 is $\mathcal{O}(\xi|V|h(G))$. As announced, the algorithm requires time $\mathcal{O}(\xi|V|h(G))$ and memory $\mathcal{O}(\xi|V|)$ to compute the blocking probabilities.

3.8 Numerical experiments

Finally we present some numerical experiments to illustrate the flow-level performance of hierarchical beamforming for both elastic and streaming traffic. The simulations presented below have been performed using a Matlab simulator. The considered parameters are gived in C.

3.8.1 Setup

In order to obtain a realistic setting and take into account the geometry of a cell in a physical network, instead of selecting arbitrary values of G, $\boldsymbol{\lambda}$, $\boldsymbol{\rho}$ and $\mathbf{r}$, we create a codebook using a similar techniques as in [57], where the main goal is to pack as many beams as possible on each level of the tree, while ensuring that they remain separated with negligible interference.

As for the cell geometry, we consider an hexagonal cell, with an antenna height of 30 meters in a strong Line of Sight environment. We use the standard Nakagami propagation model [14], and parameters as done in [57]. The main simulation parameters used for the two scenarios are summarized in Table C.1 in the appendix. The output of this process is a graph G, as well as the corresponding arrival rate $\boldsymbol{\lambda}$, loads $\boldsymbol{\rho}$ and data rates $\mathbf{r}$. We obtain G a tree with $|V| = 10$ beams, and the set of edges is given by $E = \{(1,2),(1,3),(1,4),(2,7),(2,9),(3,5),(3,6),(4,8),(4,10)\}$. It is noted that G is similar to the graph depicted in figure 3.3. The expected flow size is 8 Mbits, the arrival rates are taken proportional to $\boldsymbol{\lambda} = [0.16, 0.09, 0.12, 0.09, 0.10, 0.10, 0.10, 0.10, 0.09, 0.09]$. The value of $\boldsymbol{\lambda}$ used here is obtained by assuming that λ_v is proportional to the size of the zone covered by v. The service rate r_v is given by the harmonic mean of the data rate over the region covered by v. The loads are proportional to $\boldsymbol{\rho} = [0.11, 0.20, 0.31, 0.20, 0.59, 0.59,\ 0.61, 0.61, 0.18, 0.18]$. In order to avoid scaling problems, in all curves we present the *normalized* flow throughput $\frac{\psi_v}{r_v}$. It is noted that, when the load is very low (respectively very high), the normalized flow throughput should be close to 1 (respectively 0).

3.8.2 Elastic traffic

The flow throughput under PF allocation using the tree described above is depicted in Figure 3.6. Under PF allocation, the flow throughput as a function of the arrival rate is roughly linear for beams of high depth, while it is convex for beams of low depth. The smaller the depth, the higher the curvature. We deduce that PF tends to slightly favor beams with higher depth, and this difference is especially sensible at moderate loads. We can also see in Figure 3.6 that the flow throughput of all beams is strictly positive if $\boldsymbol{\rho}$ lies in the stability region. The flow throughput under MT allocation using the tree described above is depicted in Figure 3.7. In order to assess the accuracy of the approximate formula provided in subsection 3.6.4 we show both the approximate analytical expressions as well as the correct value estimated using simulations. In figure 3.7 "sim" denotes the values of the flow throughput estimated by simulation, and "ana" denotes the approximation given in subsection 3.6.4. It is noticed that the proposed approximation is rather accurate. On the other hand, under MT allocation, the system is not stable across the whole stability region, and beams of smaller depth get overloaded (so that the flow throughput becomes 0) when the arrival rate is sufficiently large. This is due to the fact that MT gives absolute priority to

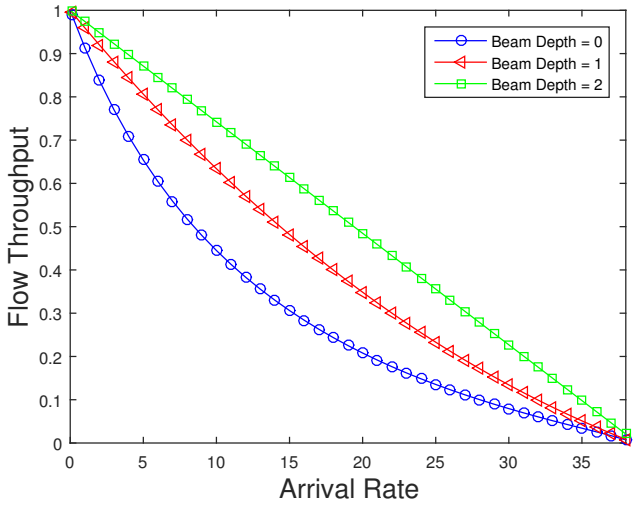


Figure 3.6: Elastic traffic, normalized flow throughput for PF allocation

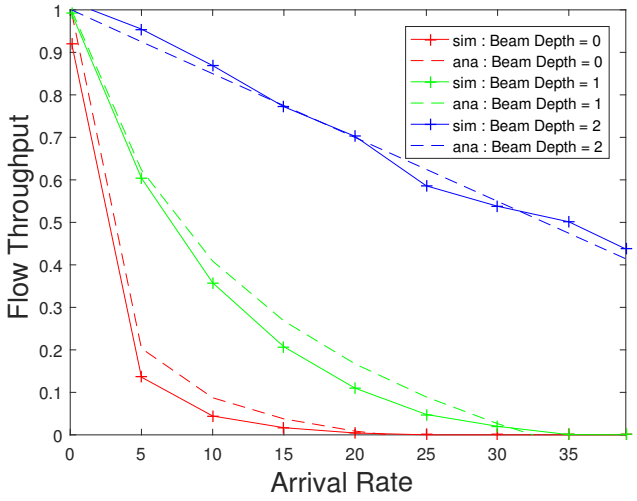


Figure 3.7: Elastic traffic, normalized flow throughput for MT allocation, approximation vs simulations.

beams with high depth. This does not mean that MT is bad per-se: when flows are static as considered in our model, MT is not throughput optimal, and tends to create overload in beams with small depth. However, in the case where flows are allowed to move during their transmission, MT would essentially have an opportunistic behaviour

and serve flows only when they can be served by a beam of high depth, at a high data rate. This should dramatically increase the throughput (at the expense of delay, of course). We do not analyze this case here, but it seems rather natural that MT would have such a behaviour.

3.8.3 Streaming Traffic

We now turn to streaming traffic. We use the same values for the loads $\boldsymbol{\rho}$ and the arrival rates $\boldsymbol{\lambda}$ as in the previous case (up to a scalar multiplicative constant). On figure 3.8 we plot the blocking probabilities of the various beams $\mathbf{p}(\boldsymbol{\rho})$ with different depth as a function of the system load. As in all Erlang-like models, the blocking rate is almost equal to 0 for low loads, then rapidly increases once the load reaches a certain threshold. Furthermore, we can see that the system tends to penalize beams with low depth, in particular the root. It would be interesting to design admission control policies which correct that problem, although it does not seem straightforward.

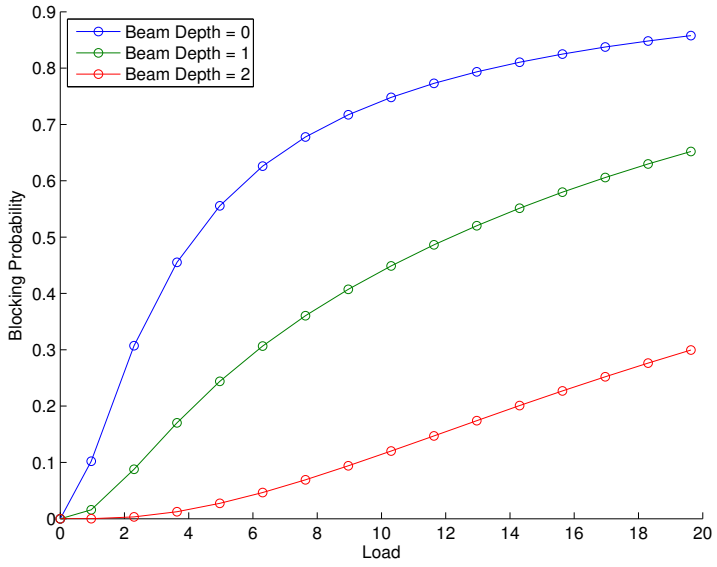


Figure 3.8: Streaming traffic, blocking probability

3.9 Proof

3.9.1 Proof of Theorem 3

Stage 1 First consider $\boldsymbol{\gamma} \in \mathbf{conv}(\mathcal{Z})$ and maximize the objective function with respect to $\boldsymbol{\delta} \in \Delta$. The objective function is separable, so that we obtain $|V|$ independent

problems:

$$\text{Maximize} \sum_{k\in\mathcal{K}(v)} f_\alpha(r_k\gamma_v\delta_k) \text{ subject to } \sum_{k\in\mathcal{K}(v)} \delta_k = 1$$

From the Karush-Kuhn-Tucker conditions, there exists ℓ_v such that the optimal value of $\boldsymbol{\delta}$ satisfies: $(\gamma_v r_k)^{1-\alpha}\delta_k^{-\alpha} = \ell_v$ for all $k \in \mathcal{K}(v)$. Using $\sum_{k\in\mathcal{K}(v)} \delta_k = 1$ we get:

$$\delta_k^\star = \frac{r_k^{\frac{1}{\alpha}-1}}{\sum_{k'\in\mathcal{K}(v)} r_{k'}^{\frac{1}{\alpha}-1}}$$

It is noted that $\boldsymbol{\delta}^\star$ is the optimal solution of the original problem since it does not depend on $\boldsymbol{\gamma}$. Now set $\boldsymbol{\delta} = \boldsymbol{\delta}^\star$ and it is noted that

$$\sum_{k\in\mathcal{K}(v)} f(r_k\gamma_v\delta_k^\star) = \phi_v f_\alpha(\gamma_v) \text{ with } \phi_v = \Big(\sum_{k\in\mathcal{K}(v)} r_k^{\frac{1}{\alpha}-1}\Big)^\alpha.$$

Stage 2 The original problem reduces to a simpler one:

$$\text{Maximize} \sum_{v\in V} \phi_v f_\alpha(\gamma_v) \text{ subject to } \boldsymbol{\gamma} \in \mathbf{conv}(\mathcal{Z}).$$

Using Proposition 1, we have that:

$$\mathbf{conv}(\mathcal{Z}) = \left\{\boldsymbol{\gamma} : \gamma_v = \kappa_v \prod_{v'\in\mathcal{A}(v)} (1-\kappa_{v'}), \boldsymbol{\kappa} \in [0,1]^{|V|}\right\}$$

allowing another simplification:

$$\begin{aligned}&\text{Maximize} \sum_{v\in V} \phi_v f_\alpha\Big(\kappa_v \prod_{v'\in\mathcal{A}(c)} (1-\kappa_{v'})\Big)\\ &\text{subject to } \boldsymbol{\kappa} \in [0,1]^{|V|}.\end{aligned}$$

Define θ_v the optimal value of this optimization problem when only considering v and its descendants, i.e. the value of:

$$\begin{aligned}&\text{Maximize} \sum_{v'\in\bar{\mathcal{D}}(v)} \phi_{v'} f_\alpha\Big(\kappa_{v'} \prod_{v''\in\mathcal{A}(v')} (1-\kappa_{v''})\Big)\\ &\text{subject to } \boldsymbol{\kappa} \in [0,1]^{|V|}.\end{aligned}$$

Now maximizing with respect to κ_v:

$$\theta_v = \max_{\kappa_v\in[0,1]} \Big[\phi_v \frac{\kappa_v^{1-\alpha}}{1-\alpha} + (1-\kappa_v)^{1-\alpha} \sum_{v':(v,v')\in E} \theta_{v'}\Big].$$

Solving the maximization over κ_v gives:

$$\kappa_v^\star = \left[1 + \left(\frac{1-\alpha}{\phi_v} \sum_{v':(v,v')\in E} \theta_{v'}\right)^{\frac{1}{\alpha}}\right]^{-1}.$$

and substituting in the definition of θ_v we get:

$$\theta_v = (\kappa_v^\star)^{1-\alpha} \frac{\phi_v}{1-\alpha} + (1-\kappa_v^\star)^{1-\alpha} \sum_{v':(v,v')\in E} \theta_{v'}.$$

When v is a leaf, we have $\theta_v = \frac{\phi_v}{1-\alpha}$, and the previous relation allows to calculate the value of θ_v and $\kappa_v^\star$ for all $v \in V$. Finally, the solution is given by the relation:

$$\gamma_v^\star = \kappa_v^\star \prod_{v'\in\mathcal{A}(v)} (1-\kappa_{v'}^\star)$$

which concludes the proof.

3.9.2 Proof of Theorem 5

As mentionned above, the system is equivalent to a single M/M/1 queue with FIFO service order and priorities with preemptive resume, where customers of class $v \in V$ represent the users served by beam V. Namely, customers of class v receive service if and only if no customers of class $v+1, \ldots, |V|$ are present in the system. The result is similar to [2], and we present a proof for completeness. Users of class $|V|$ have the highest priority and are not influenced by other classes so that $N_{|V|}(t)$ follows an M/M/1 process and we have:

$$\mathbb{E}(N_{|V|}(t)) = \frac{\rho_{|V|}}{1-\rho_{|V|}}.$$

Now consider a class $v < |V|$. We denote by $\mathbb{E}(W_v)$ the mean amount of time a customer of class v spends in the system, including when receiving service. The waiting time of a customer served by v can be divided in epochs of length $X_1, X_2, ...$ as follows. The duration X_1 of first epoch equals the amount of work associated with all the customers with the same or higher priority present in the queue upon his arrival, including himself. The length X_2 of the second epoch equals the amount of higher priority work arriving during the first epoch (of duration X_1). The length of the third epoch equals the amount of higher priority work arriving during the second epoch (of duration X_2), etc. This allows to express the waiting time as a sum:

$$\mathbb{E}(W_v) = \sum_{k\in\mathbb{N}} \mathbb{E}(X_k)$$

From the PASTA property, the amount of work with equal or higher priority upon arrival of a customer of class v (including himself) is:

$$\mathbb{E}(X_1) = \frac{1}{r_v} + \sum_{v' \geq v} \frac{\mathbb{E}(N_{v'}(t))}{r_{v'}}$$

The expected amount of work with higher priority arriving during a duration X_k is given by recursion:

$$\mathbb{E}(X_{k+1}) = \mathbb{E}(X_k) \left(\sum_{v'>v} \rho_{v'} \right) = \mathbb{E}(X_1) \left(\sum_{v'>v} \rho_{v'} \right)^{k-1}.$$

so that, summing:

$$\mathbb{E}(W_v) = \frac{\mathbb{E}(X_1)}{1 - \sum_{v'>v} \rho_{v'}}$$

Replacing $\mathbb{E}(X_1)$ by its expression:

$$\mathbb{E}(W_v) = \frac{1}{1 - \sum_{v'>v} \rho_{v'}} \left(\frac{1}{r_v} + \sum_{v' \geq v} \frac{\mathbb{E}(N_{v'}(t))}{r_{v'}} \right)$$

From Little law, $\lambda_v \mathbb{E}(W_v) = \mathbb{E}(N_v(t))$ so that:

$$\frac{\mathbb{E}(N_v(t))}{\lambda_v} = \frac{1}{1 - \sum_{v'>v} \rho_{v'}} \left(\frac{1}{r_v} + \sum_{v' \geq v} \frac{\mathbb{E}(N_{v'}(t))}{r_{v'}} \right),$$

which yields the relation:

$$\mathbb{E}(N_v(t)) = \frac{\rho_v + \lambda_v \sum_{v'>v} \frac{\mathbb{E}(N_{v'}(t))}{r_{v'}}}{1 - \sum_{v' \geq v} \rho_{v'}}.$$

One may readily check by recursion that:

$$\mathbb{E}(N_v(t)) = \frac{\rho_v \left(1 + \sum_{v' \geq v} \rho_{v'} \left(\frac{r_v}{r_{v'}} - 1 \right) \right)}{\left(1 - \sum_{v' \geq v} \rho_{v'} \right) \left(1 - \sum_{v'>v} \rho_{v'} \right)}$$

which is the announced result.

3.10 Conclusion

We have considered hierarchical beamforming in wireless networks, which is an attractive alternative to other beamforming techniques due to the existence of efficient algorithms for flow association and multi-flow scheduling. We have provided computationally efficient algorithms for fair rate allocation including proportional fairness, maximum throughput and max-min fairness, in order to perform resource allocation in real time. We next have proposed closed-form formulas for flow level performance, for both elastic (flow throughput) and streaming traffic (blocking rates) [23].

4

Model of starvation for connection E2E

This chapter aims at developing Self-Organizing Network (SON) function that optimizes Quality of Experience (QoE) of E2E video streaming connection. For the QoE, we consider the starvation of video streaming represented by a buffer size below a predetermined threshold. In 2016 video accounted for 60% of the global data traffic in cellular networks[61]. Inspite of the fast deployment of 4G networks, the support of high bit rate streaming services such as 4K Ultra HD or 360° live videos remains a major challenge for network operators.

Three types of transport protocols are used today in communication networks. The first one is User Datagram Protocol (UDP), is a minimal message-oriented transport layer protocol. UDP provides no guarantees to the upper layer protocol for message delivery and the UDP layer retains no state of UDP messages once sent. The second protocol is Transmission Control Protocol (TCP), and unlike UDP, it provides an acknowledgment to ensure the transmission and it uses algorithms for congestion avoidance, some of which will be studied in this Chapter. Recently Google has developed a new type of transport protocol, the Quick UDP Internet Connections (QUIC) [31]. It is a UDP protocol designed to provide security protection, one of QUIC main goal is to move control of the congestion avoidance algorithms into the application space at both endpoints.

For the rest of the thesis the TCP is assumed as the transport protocol. The goal of this part of the manuscript is to understand the impact of TCP protocols on the QoE of end users and exploit this knowledge to devise a SON algorithm to control E2E QoE. To this end we first explain in Section 4.1 how video streaming works as well as the related QoE indicator for this service. Section 4.2 introduces several TCP protocols such as CUBIC (actually employed with LINUX system), VEGAS (used in Compound-TCP protocol employed in Windows system) and AIMD (a family of TCP protocols that are the first to be used). Section 4.3 develops the throughput models for the above TCP protocols. A buffer model for the E2E connection using a drifted Brownian model is

proposed in Section 4.4. Section 4.5 develops a closed form expression for the starvation probability for the video service, and is validated numerically in Section 4.6. The latter Section also introduces the SON function that controls the the starvation probability by adjusting the video quality. Concluding remarks are summarized in Section 4.7.

4.1 Motivation and related work

The growing portion of video streaming in the internet traffic has motivated researchers to model QoE of end users consuming this type of service on the one hand, and to develop techniques to dynamically control QoE on the other hand.
Several works address the modeling and prediciton of E2E QoE of video streaming services. [70] proposes an analytical framework to compute a closed-form starvation probability on the basis of ordinary differential equations (ODEs). To this end, view records from one of the largest streaming providers in China were collected. [34] applies supervised machine learning technique to predict video starvation using as input: channel conditions and number of active users. The prediction of starvation can be used by service providers to adapt the video bitrate. In [45] the authors use a Markov chain based analysis to compute the user probability of starvation. [38] analyzes starvation probabilities with a discrete-time Markov chain and show that the adaptation of the video quality (rate) can significantly reduce the starvation probability.
Recently, Google has introduced the QUIC protocol which encrypts the video streaming packets. This evolution hinders the capability of network operator to control QoS in the network. As a result, a number of contributions have proposed traffic profiling (see for example [61]), which is a non-intrusive solution that observes packet flows at the transmit queue of base stations, edge-routers, or gateways. By analyzing IP flows in real time, the presented scheme identifies different phases of an HAS session and estimates important application-layer parameters, such as play-back buffer state and video encoding rate. In [8] the authors present an optimal solution that adapts the video rate to the current and predicted channel state as well. Optimal formulation is based on Markov Decision Process. The solution is tested in a live network with real content and utilizes channel traces from vehicular users. In [37] the context of a UE is related to its coverage condition, e.g. indoor or outdoor, inside or outside a tunnel etc. The purpose of this SON function is to optimize HAS resource allocation by detecting significant context change, such as a mobile entering a building or a car entering a tunnel. In [62] the channel prediction is used to improve communication performance. This paper describes a new approach for allocating resources to video streaming traffic using long term channel prediction while accounting for quality of service. The linearity of the proposed model allows to formulate a Linear Programming problem that optimizes the trade-off between the allocated resources and the stalling time of the media stream. In

[25] the authors propose a QoE Manager that can deliver superior QoE for each session by loosening the radio efficiency targets to maximize the customer experience.
The ultimate objective of this Chapter is to develop a low complexity SON function for controlling the starvation probability of video streaming traffic over a TCP protocol. The motivation of this approach is to benefit from the access the operator can have to the transport layer parameters. It is noted that other contributions in the literature are based on application layer parameters only. To this end, we develop closed form formula for the starvation probability based on the drifted Brownian model, as well on expressions for statistical parameters of different TCP schemes.

4.1.1 Video Streaming

A video stream is a data flow sent, after compression, from a service provider (for example Dailymotion) to the end user. The main concept behind streaming video is that the user does not have to wait to download the entire file or the streaming data to play it but instead, the media content is sent in a continuous stream of packets and is played as it arrived at the client terminal. Generally video streaming uses adaptative mechanism such as Dynamic and Adaptive Streaming over HTTP (DASH). The video content is stored in the web server in chunks, each of which is available with different qualities. According to the UE state, it requests a video chunk with the adequate quality. Figure 4.1 schematizes the video stream connection between a server and a user. Further details can be found in [44].

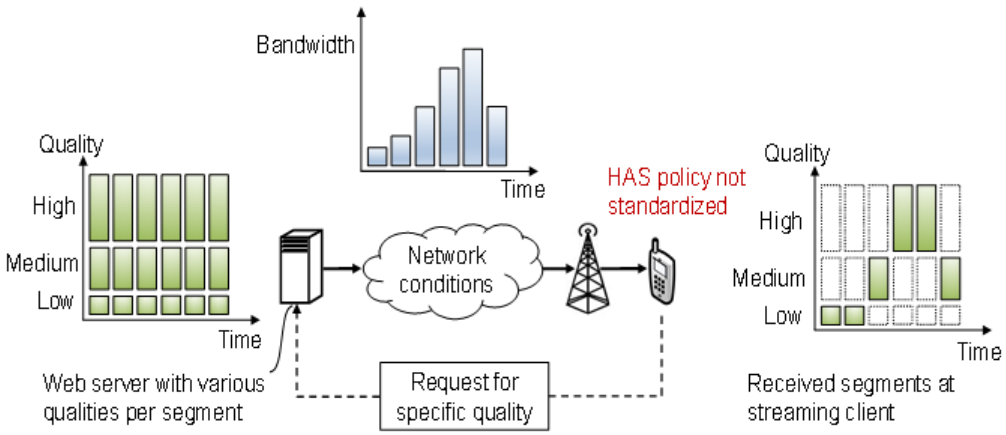


Figure 4.1: Figure representing the video stream connection between server to user (Source: Huawei France Research Center, presentation on AI Challenges in 5G, 2017)

4.1.2 QoE for video user

The QoE is defined here as the measure of the overall acceptability of an application or a service, which is perceived subjectively by the end user. Several indicators describe QoE of video streaming services such as video freeze, video rate, the variability and start-up delay.
The Video freeze happens during the video playback, if the video hangs, it is said to be in video freeze state. In the literature, this phenomenon is known as video starvation or stalling. More details are found in [35].
In this Chapter we focus on the modelling of video starvation and how a SON algorithm can prevent and control this phenomenon considering TCP parameters. Figure 4.2 shows an example of evolution of a buffer content. In blue the buffering phase, in orange the steady-state phase when the play-out starts and in green the depletion due to the fact that the play-out continue to read packets but the buffer does not receive data.

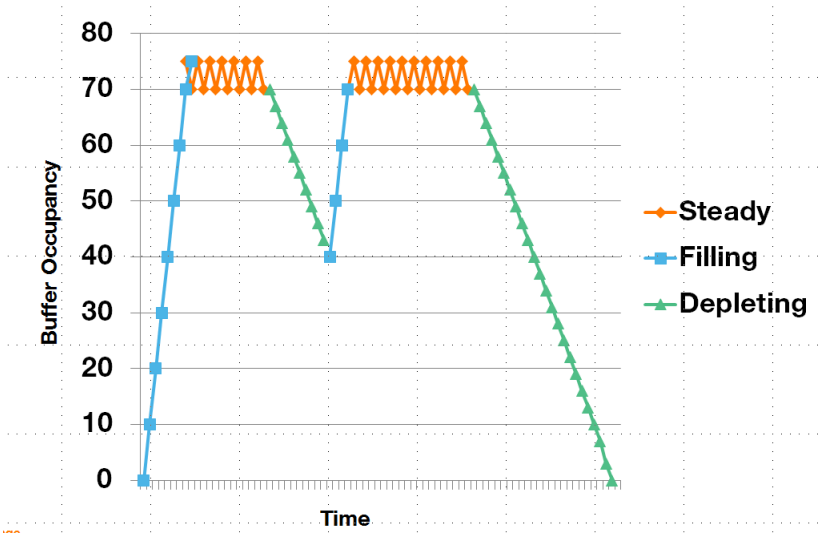


Figure 4.2: Figure representing the evolution of buffer content

4.2 Algorithms

TCP has been the dominat transport protocol in the Internet. There exist several types of TCP protocols and in this Section we describe the principle ones [42], with a focus on two TCP protocols.
We start with the Additive Increase and Multiplicative Decrease (AIMD), which is a family of TCP protocols (such as Reno or Tahoe). These protocols are share a common feature, namely for each round trip time (RTT) they linearly increase the number of

packets sent, until a packet loss occurs.. In this case, the protocol divides by a rational coefficient between 0 and 1 the number of packets sent. We continue with CUBIC which is a recent TCP protocol used in Unix systems. It is similar to AIMD but the additive increase is not linear but is rather the result of a cubic function that has good mathematical properties for reducing the congestion. Finally we present VEGAS which is a delay based window. The protocols notation are described in appendix D

4.2.1 AIMD

AIMD (Additive Increase and Multiplicative Decrease) is a feedback control algorithm [33], [53] for allocating bandwidth or transmission rate to the different jobs in a E2E connection. In this algorithm, AIMD increases the bandwidth of each job (the number of packets sent) linearly at a rate $\alpha > 0$ (typically $\alpha = 1$) until it detects that one of the capacity bottlenecks through which a transmission passes through is reached, at which point, the algorithm cuts the connection bandwidth by a multiplicative factor of $0 < \beta < 1$ (typically $\beta = \frac{1}{2}$).

Algorithm 4: AIMD protocol

Data: $S = 65535$; $W = 1$
while *Data to download* **do**
  if $Z = 0$ *and* $Z = 3$ **then**
    $S = W * \beta$
  if $Z = 1$ *and* $W \leq S$ *(SS)* **then**
    $W = W + 1$
  else if $Z = 1$ *and* $W > S$ *(CA)* **then**
    $W = W + \frac{\alpha}{W}$
  else if $Z = 3$ *(FRc)* **then**
    $W = W * \beta$
  else if $Z = 0$ *(FRt)* **then**
    $W = 1$

4.2.2 CUBIC

CUBIC is the mostly used TCP protocol for Unix systems. It has been developped in [24] where the authors enhance BIC TCP [64] to give the new CUBIC protocol. The difference with AIMD protocol is in the window growth function that is a cubic function instead of a linear function.
We introduce $cubic(t) = \alpha(t - T_i - \sqrt[3]{\beta\frac{W_{max}}{\alpha}})^3 + W_{max}$, τ describing the elapsed time from the last packet loss, the constant α denoting the window growth factor and the constant β representing the multiplicative decrease factor.

Algorithm 5: CUBIC protocol

Data: $W = 1$; $W_{max} = 100$
while *Data to download* **do**
  if $Z = 0$ *or* $Z = 3$ **then**
    $W_{max} = \beta W$
  if $Z = 1$ *and* $W < W_{max}(1-\beta) + 3\frac{\beta*\tau}{(2-\beta)RTT}$ **then**
    $W = W_{max}(1-\beta) + 3\dfrac{\beta * \tau}{(2-\beta)RTT}$
  else if $Z = 1$ **then**
    $W = W + \dfrac{cubic(t + RTT) - W}{W}$
  else
    $W = (1-\beta) * W$

4.2.3 VEGAS

VEGAS is a new TCP protocol, namely a delay-based window [13] as briefly explained below. Until the mid-1990s, all of TCP's set timeouts and measured round-trip delays were based upon only the last transmitted packet in the transmit buffer. University of Arizona researchers Larry Peterson and Lawrence Brakmo introduced TCP Vegas, named after the largest Nevada city, in which timeouts were set and round-trip delays were measured for every packet in the transmit buffer. In addition, TCP Vegas uses additive increases in the congestion window. This variant was not widely deployed outside Peterson's laboratory. In a comparison study of various TCP congestion control algorithms, TCP Vegas appeared to be the smoothest followed by TCP CUBIC [27].
RTT_{min} is the smallest round trip time seen since the connection has started. RTT is the current. Typically $\alpha = 2$, $2 \leq \beta \leq 4$ and $\gamma = 1$.

4.3 TCP throughput

We present the mathematical models used for simulating the TCP throughput. We recall that TCP is a reliable window-based flow control protocol where the window is increased until a packet loss is detected. Once the lost packets are recovered, the source resumes its window increase. As a performance measure, we consider the throughput of a long time TCP connection having an infinite amount of data to send. The mathematical analysis of TCP requires two steps. First, we need to construct a model for the window size evolution. Most of the existing models ignore the Slow Start phase and make the assumption that the source stays always in Congestion Avoidance mode. The phase of

Algorithm 6: VEGAS protocol

Data: $W = 1$
while *Data to download* **do**
  if $Z = 0$ *(FRt)* **then**
    $W = 1$
  if $Z = 3$ *(FRc)* **then**
    $W = \frac{W}{2}$
  if $Z = 1$ *and* $\frac{W}{RTT_{min}} - \frac{W}{RTT} < \frac{\gamma}{RTT_{min}}$ *(SS)* **then**
    $W = W + 1$
  else if $Z = 1$ *and* $\frac{W}{RTT_{min}} - \frac{W}{RTT} < \frac{\alpha}{RTT_{min}}$ *(CA)* **then**
    $W = W + \frac{1}{W}$
  else if $Z = 1$ *and* $\frac{W(n)}{RTT_{min}} - \frac{W(n)}{RTT} > \frac{\beta}{RTT_{min}}$ *(CA)* **then**
    $W = W - \frac{1}{W}$

recovery, FRt, is assumed to be negligible and the source resumes its increase directly after the reduction. Second TCP analysis requires a characterization of the time between congestion events. In the remaining sections we suppose that the traffic is in a stationary state and the time between two losses follows an exponential distribution, so that $(T_i)_i$, the i-th time where a loss occurs, forms a Poisson process with rate λ.
Moreover we show how to get the first, second and correlation moments in closed form for different TCP protocols. This allows to reduce the complexity of E2E SON mechanism, as will be explained later in the sequal.

4.3.1 Altman model

We use the Altman model described in [4] that is the generalization of square-root model [6]. This model describes the throughput behavior of AIMD protocol in this way:

$$X_{T_{i+1}} = \alpha(X_{T_i} + \beta(T_{i+1} - T_i))$$

where X_t is the data rate at time t, $(T_i)_i$ are the successive instants at which a loss occurs, $\alpha \in (0, 1)$ and $\beta \in \mathbb{R}$ are the two parameters of the AIMD algorithm. It is noted that X_t is linear by parts between two loss instants, so that:

$$X_t = X_{T_i} + \beta(t - T_i), \text{ for } t \in [T_i, T_{i+1}).$$

First moment

In [4] the first moment of X_t is described by the following theorem:

Theorem 8. *The first moment of X_t:*

$$\mathbb{E}(X_t) = \frac{\beta}{\lambda(1-\alpha)}$$

The proof proposed is not adapted for our work and we use a different approach for the proof:

Proof. Recall that a loss occurs in the interval $[t, t+dt]$ with probability λdt, and that the occurrence of a loss is independent of X_t. Therefore:

$$\mathbb{E}(X_{t+dt}) = (1-\lambda dt)\mathbb{E}(X_t + \beta dt) + \lambda dt \mathbb{E}(\alpha X_t)$$

Since $\mathbb{E}(X_{t+dt}) = \mathbb{E}(X_t)$ we get:

$$\mathbb{E}(X_t) = \mathbb{E}(X_t) + dt(\lambda \mathbb{E}(X_t)(\alpha - 1) + \beta) + o(dt)$$

Solving for $\mathbb{E}(X_t)$ we get the result announced.

Second moment

For the second moment of X_t we obtain an simpler formula than in [4]. In the next section we will see why we need this representation.

Theorem 9. *The second moment of X_t is:*

$$\mathbb{E}(X_t^2) = \frac{2\beta\mathbb{E}(X_t)}{\lambda(1-\alpha^2)} = \frac{2\beta^2}{\lambda^2(1-\alpha)(1-\alpha^2)}$$

Proof. From the same reasoning as in theorem (8):

$$\begin{aligned}
\mathbb{E}(X_{t+dt}^2) &= (1-\lambda dt)\mathbb{E}((X_t + \beta dt)^2) + \lambda dt \mathbb{E}(\alpha^2 X_t^2) \\
&= (1-\lambda dt)(\mathbb{E}(X_t^2) + 2\beta\mathbb{E}(X_t)dt + \beta^2(dt)^2) + \lambda dt \alpha^2 \mathbb{E}(X_t^2) \\
&= \mathbb{E}(X_t^2) + dt(\lambda(\alpha^2 - 1)\mathbb{E}(X_t^2) + 2\beta\mathbb{E}(X_t)) + o(dt).
\end{aligned}$$

Now using the fact that $\mathbb{E}(X_{t+dt}^2) = \mathbb{E}(X_t^2)$:

$$0 = \lambda(\alpha^2 - 1)\mathbb{E}(X_t^2) + 2\beta\mathbb{E}(X_t),$$

which leads to the announced result.

Correlation moment

The correlation moment is described by the following theorem:

Theorem 10. *The correlation moment* $\mathbb{E}[X_t X_{t'}]$ *when* X_t *is stationary is:*

$$\mathbb{E}[X_0 X_t] = \mathbb{E}(X_0)^2 + (\mathbb{E}(X_0^2) - \mathbb{E}(X_0)^2)e^{-t(1-\alpha)\lambda}$$

Proof. Since X_t is stationary, we have that $\mathbb{E}(X_t X_{t'}) = \mathbb{E}(X_0 X_{t'-t})$. Define $\gamma(t) = \mathbb{E}(X_0 X_t)$ the autocorrelation of X_t. We now calculate γ. We have:

$$\begin{aligned}\mathbb{E}(X_0 X_{t+dt}) &= (1-\lambda dt)\mathbb{E}(X_0(X_t + \beta dt)) + \lambda dt \mathbb{E}(\alpha X_0 X_t) \\ &= \mathbb{E}(X_0 X_t) + dt(\mathbb{E}(X_0)\beta + (\alpha-1)\lambda\mathbb{E}(X_0 X_t)) + o(dt)\end{aligned}$$

By definition of γ:

$$\gamma(t+dt) = \gamma(t) + dt(\mathbb{E}(X_0)\beta + (\alpha-1)\lambda\gamma(t)) + o(dt),$$

so that γ obeys the differential equation:

$$\frac{d}{dt}\gamma(t) + (1-\alpha)\lambda\gamma(t) = \mathbb{E}(X_0)\beta. \tag{4.1}$$

Solving (4.1) we get:

$$\gamma(t) = \frac{\mathbb{E}(X_0)\beta}{(1-\alpha)\lambda} + \left(\gamma(0) - \frac{\mathbb{E}(X_0)\beta}{(1-\alpha)\lambda}\right)e^{-t(1-\alpha)\lambda}$$

Using the fact that $\gamma(0) = \mathbb{E}(X_0^2)$, and $\mathbb{E}(X_0) = \frac{\beta}{(1-\alpha)\lambda}$ we get:

$$\gamma(t) = \mathbb{E}(X_0)^2 + (\mathbb{E}(X_0^2) - \mathbb{E}(X_0)^2)e^{-t(1-\alpha)\lambda}$$

□

One can obtain the following result that will be useful for the end of the Chapter:

Corollary 2. *Thanks to theorem (10) we obtain for* X_t *stationary:*

$$\int_0^t \int_0^t \mathbb{E}\left(X_s X_{s'}\right) ds ds' = t^2\mathbb{E}(X_0)^2 + 2(\mathbb{E}(X_0^2) - \mathbb{E}(X_0)^2)\left(\frac{t}{(1-\alpha)\lambda} + \frac{1-e^{-t(1-\alpha)\lambda}}{(1-\alpha)^2\lambda^2}\right).$$

Proof.

$$\begin{aligned}\int_0^t \int_0^t \mathbb{E}\left(X_s X_{s'}\right) ds ds' &= 2\int_0^t \int_0^t \mathbb{E}\left(X_s X_{s'}\right) \mathbf{1}\{s \leq s'\} ds' ds \\ &= 2\int_0^t \left(\int_s^t \mathbb{E}\left(X_s X_{s'}\right) ds'\right) ds \\ &= 2\int_0^t \left(\int_s^t \gamma(s'-s) ds'\right) ds \\ &= 2\int_0^t \left(\int_0^{t-s} \gamma(s') ds'\right) ds.\end{aligned}$$

where we used the fact that, since $(X_t)_t$ is stationary,

$$\mathbb{E}\left(X_t X_{t'}\right) = \mathbb{E}\left(X_{t'-t} X_0\right) = \gamma(t'-t).$$

We have:

$$\begin{aligned}\int_0^t \left(\int_0^{t-s} e^{-s'(1-\alpha)\lambda} ds'\right) ds &= \frac{1}{(1-\alpha)\lambda}\int_0^t (1 - e^{-(t-s)(1-\alpha)\lambda}) ds \\ &= \frac{t}{(1-\alpha)\lambda} - \frac{1-e^{-t(1-\alpha)\lambda}}{(1-\alpha)^2\lambda^2}\end{aligned}$$

By adding the different results we have the desired result.

□

4.3.2 CUBIC model

This Section develops a closed form approximation for the average throughput of the CUBIC protocol. Several works have reported analytical approximation for the average CUBIC throughput as in [9],[46] and [47]. Our contribution in this Section is to propose a closed form expression for the first and second moment of CUBIC TCP, the latter being an original contribution. We consider $(T_i)_i$ are the successive instants at which a loss occurs and $T_{i+1} - T_i \simeq Exp(\lambda)$ the exponential probability with parameter λ, with T_i^- the left limit of the throughput in X_{T_i}. We consider X_{T_i} a point process and suppose that the throughput reached the stationary state. For time $t \in [T_i, T_{i+1}]$ we have :

$$X_t = X_{T_i^-} + \alpha\left((t - T_i) - \sqrt[3]{\frac{X_{T_i^-}\beta}{\alpha}}\right)^3$$

We note that it is hard to obtain a closed form formula for the average throughput and other indicators.

First moment

Theorem 11. *The first moment of TCP CUBIC, using* $\mu = \frac{\mathbb{E}(X_{T_i})\beta}{\alpha}$*, throughput* X_t *is:*

$$\mathbb{E}[X_t] = \frac{1}{t}\int_0^t \mathbb{E}[X_u]du = \frac{\mathbb{E}[\int_0^t X_u du]}{t} = \frac{\mathbb{E}[\int_{T_i}^{T_{i+1}} X_u du]}{\mathbb{E}[T_{i+1} - Ti]} = \mathbb{E}(X_{T_i}) + \alpha(\frac{6}{\lambda^3} - \frac{6}{\lambda^2}\mu^{1/3} + \frac{3}{\lambda}\mu^{2/3} - \mu)$$

Proof. We have for time T_{i+1}:

$$X_{T_{i+1}^-} = X_{T_i^-} + \alpha\left((T_{i+1} - T_i) - \sqrt[3]{\frac{X_{T_i^-}\beta}{\alpha}}\right)^3$$

and

$$X_{T_{i+1}} = (1-\beta)X_{T_{i+1}^-} = X_{T_i} + \alpha(1-\beta)\left((T_{i+1} - T_i) - \sqrt[3]{\frac{X_{T_i}\beta}{(1-\beta)\alpha}}\right)^3$$

Thanks to the stationary assumption and the stability of CUBIC protocol we can write $\mathbb{E}[X_{T_{i+1}}] = \mathbb{E}[X_{T_i}]$ and use the approximation $\mathbb{E}[X_{T_i}] \simeq X_{T_i}$ and $\mathbb{E}[X_{T_i^-}] \simeq X_{T_i^-}$. To obtain the first moment we introduce $Z = Exp(1)$ that give $\mathbb{E}[Z] = \mathbb{E}[(T_{i+1} - T_i)\lambda]$:

$$\begin{aligned}
\mathbb{E}\left[\left((T_{i+1} - T_i) - \sqrt[3]{\frac{X_{T_i}\beta}{(1-\beta)\alpha}}\right)^3\right] = 0 &\Leftrightarrow \mathbb{E}\left[\left(\frac{Z}{\lambda} - \sqrt[3]{\frac{\mathbb{E}[X_{T_i}]\beta}{(1-\beta)\alpha}}\right)^3\right] = 0 \\
&\Leftrightarrow \mathbb{E}\left[\left(\frac{Z}{\lambda} - x\right)^3\right] = 0 \\
&\Leftrightarrow \mathbb{E}\left[(Z - \lambda x)^3\right] = 0 \\
&\Leftrightarrow \mathbb{E}\left[(Z - z)^3\right] = 0 \\
&\Leftrightarrow z^3 - 3z^2 + 6z - 6 = 0 \\
&\Leftrightarrow z = \sqrt[3]{1+\sqrt{2}} + \sqrt[3]{1-\sqrt{2}}
\end{aligned}$$

with $x = \sqrt[3]{\frac{\mathbb{E}[X_{T_i}]\beta}{(1-\beta)\alpha}}$ and $\lambda x = z$. Thanks to the Cardan method we have a real solution for the cubic function:

$$\mathbb{E}[X_{T_i}] = \frac{(1-\beta)\alpha x^3}{\beta} = \frac{(1-\beta)\alpha(\frac{z}{\lambda})^3}{\beta}$$

In order to compute $\mathbb{E}[X_t]$ we use the steady-state assumption to admit that $w = \mathbb{E}(X_{T_i})\ \forall i \in \mathbb{N}$. Now we model the CUBIC throughput in stationary state by an aproximation. We call *cubic+* the algorithm described by the relation :

$$X_t = w + \alpha \left((t - T_i) - \sqrt[3]{\frac{w\beta}{\alpha}} \right)^3$$

We use the notation $\mu = \frac{w\beta}{\alpha}$ and thanks to the property of point processes [5] we obtain the result.

Second moment

We consider time $s' > s$, $T_{i(s)}$ being the latest time in which packet loss occured before time s and $Z = s - T_{i(s)} = Exp(\lambda)$. For the second moment we obtain (knowing that $\mathbb{E}[(Z - \mu^{\frac{1}{3}})^3] = 0$):

Theorem 12. *The second moment of CUBIC throughput X_t is:*

$$\mathbb{E}[X_s^2] = \mu 4!(1 + \frac{3}{\lambda} + \frac{1}{\lambda^2}) + \mu^{\frac{4}{3}}(12 + 2\lambda^2 + 16\lambda) - \mu^{\frac{5}{3}}(4\lambda^3 + 2\lambda^4) + \mu^2\lambda^4)$$

Proof.

$$\begin{aligned}
\mathbb{E}[X_s^2] &= \mathbb{E}[(w + \alpha((s - T_{i(s)}) - \mu^{\frac{1}{3}})^3)^2] \\
&= w^2 + 2w\alpha\mathbb{E}[(Z - \mu^{\frac{1}{3}})^3] + \alpha^2\mathbb{E}[(Z - \mu^{\frac{1}{3}})^6] \\
&= w^2 + \alpha^2\mathbb{E}[(Z - \mu^{\frac{1}{3}})^6] \\
&= w^2 - \alpha^2(\frac{6!}{\lambda^6} - \mu^{\frac{1}{3}}2\frac{5!}{\lambda^4}(2 - \frac{1}{\lambda}) + \mu^{\frac{2}{3}}\frac{4!}{\lambda^2}(6 + \frac{8}{\lambda} + \frac{1}{\lambda^2}) \\
&- \mu 4!(1 + \frac{3}{\lambda} + \frac{1}{\lambda^2}) + \mu^{\frac{4}{3}}(12 + 2\lambda^2 + 16\lambda) - \mu^{\frac{5}{3}}(4\lambda^3 + 2\lambda^4) + \mu^2\lambda^4)
\end{aligned}$$

Thanks to the previous results we have an approximation of the first and second moment of a throughput of a CUBIC connection. These results are obtained using the CUBIC model proposed above. We show that the approximation is accurate. A future work will be to find a closed formula that describes the correlation moment. Such a result could be used to calculate the starvation probability of an E2E connection that uses a TCP CUBIC protocol.

4.3.3 VEGAS model

The paper [51] has developed a simple and accurate model to estimate the throughput of a Vegas flow as a function of packet loss rate, average round trip time, minimum observed round trip time, and protocol parameters α, β. The model provides two closed-form

analytic throughput estimates, respectively for the cases that the network conditions do and do not permit to the TCP VEGAS flow to acquire its target backlog in the connection path. A simple constraint on packet loss rate was developed to determine which of the two expressions should be used. Obtaining the first and second moment for TCP VEGAS in closed form is not straightforward. The next Section explains in details how we obtain a closed formula to calculate the starvation probability.

4.4 Diffusion model

We introduce the diffusion model that describes the buffer evolution and provides a closed formula for starvation probability. This model is interesting because it can be applied to all kinds of TCP protocols. This model needs first, second and correlation moments, which explains the motivation for the above developments of closed form expressions.

4.4.1 Buffer model

Buffer model corresponds to incoming packet minus packets read during the duration of a video. Using the notation for buffer content B_t, we can write :

$$B_t = B_0 + \int_0^t X_s ds - rt$$

where the TCP throughput is represented by the process X_s, t is the duration of a video and r the constant bitrate video. We presently focus on modelling the incoming packets, we use the drifted Brownian model :

$$\int_0^t X_s ds \approx mt + \sigma W_t \tag{4.2}$$

Figure 4.3 presents the time variation of the TCP throughput X_s and the incoming packets $\int_0^t X_s ds$. One can see that X_s varies on a much faster time scale than $\int_0^t X_s ds$. The evolution of incoming packets, (blue curve) in the buffer follows a tendency (the drift m in the model) but the TCP throughput (green curve) evolves with high fluctuations. This phenomenon is represented by the Brownian motion W_t in the model.

We represent the buffer content as a drifted Brownian model, with a drift of $m - r$, so that:

$$B_t \approx B_0 + (m - r)t + \sigma W_t \tag{4.3}$$

where $(W_t)_t$ is a standard Brownian motion, r the bitrate video and m and σ are the drift and standard deviation of the diffusion approximation.

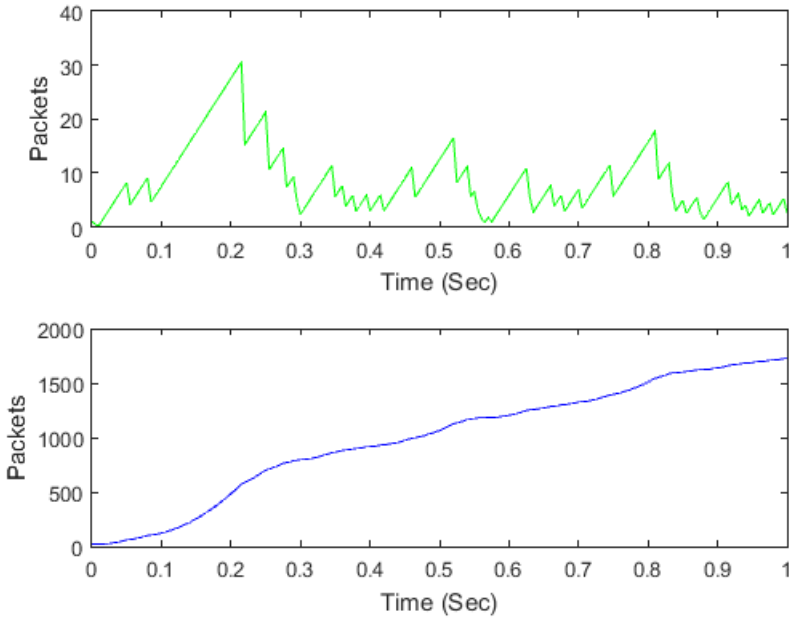


Figure 4.3: Time evolution of the TCP throughput (upper Figure in green) and the incoming packets (lower Figure in blue)

We assume that X_t is in a stationary state and we consider the starvation probability in this model. For the diffusion approximation to be useful, we now compute the parameters m and σ.

Theorem 13. *To compute m and σ the parameters of the diffusion approximation are given by:*

$$m = \mathbb{E}(X_t)$$

and

$$\sigma^2 = \lim_{t\to\infty} \frac{1}{t}\left(\int_0^t \int_0^t \mathbb{E}\left(X_s X_{s'}\right) ds ds' - t^2 \mathbb{E}(X_0)^2\right).$$

Proof. Taking the first moment of (4.2) we get:

$$\int_0^t \mathbb{E}\left(X_s\right) ds = mt$$

so that:

$$\mathbb{E}(X_t) = m$$

since X_t is in stationary regime. Taking the second moment we get:

$$\mathbb{E}\left(\left(\int_0^t X_s ds\right)^2\right) = t^2 \mathbb{E}(X_0)^2 + \sigma^2 t$$

The above holds for all t, so we get:

$$\sigma^2 = \frac{1}{t}\left(\mathbb{E}\left(\left(\int_0^t X_s ds\right)^2\right) - t^2\mathbb{E}(X_0)^2\right).$$

We note that the buffer model can be applied to different TCP protocols. This is due to the fact that you can approximate m and σ^2 if you do not have the closed form of them.

4.5 Starvation control

Using the expression for the starvation probability, we propose a SON function that controls the QoE and steering it to a predefined value by adjusting the video bitrate. It computes a starvation probability in a closed form. This SON function needs as input, transport layer parameters and the initial buffer content of the user. The main goal of the proposed solution is to design a low complexity control mechanism.

4.5.1 Hitting time density of a Brownian drift

In order to obtain the bitrate of a video playout using a TCP connection we need a analytical formula for the probability that $(B_t)_{t\leq T}$, described in (4.3), reaches starvation before an arbitraty time T. Before we derive r, we present the closed formula for the probability mentionned above, denoted by $\mathbb{P}(H_0 \leq T)$:

$$\begin{aligned}
H_0 &= inf\{t \geq 0 : B_t = 0\} \\
&= inf\{t \geq 0 : B_0 + (m-r)t + \sigma W_t = 0\} \\
&= inf\{t \geq 0 : (m-r)t + \sigma W_t = -B_0\} \\
&= inf\{t \geq 0 : \frac{(m-r)}{\sigma}t + W_t = -\frac{B_0}{\sigma}\} \\
&= inf\{t \geq 0 : ct + W_t = \delta\}
\end{aligned}$$

Where inf is the function that gives the minimum of a set and $c = \frac{(m-r)}{\sigma}$ and $\delta = -\frac{B_0}{\sigma}$.
Now we must find the density of the hitting time H, denoted by f, for a Brownian motion with drift c, $X_t = ct + W_t$, and $X_0 = 0$.
For $\delta = 0$ the hitting time density is trivial because the process start with $X_0 = 0$. For $\delta \neq 0$ we have the following result :

$$f(t) = \frac{|\delta|}{\sqrt{2\pi t^3}} e^{-\frac{(\delta - ct)^2}{2t}}$$

Thank to the Girsanov theorem (2.3.2) and the joint distribution of $(X_t, \sup_{s\leq t} X_s)$ we obtain :

$$\mathbb{P}(H_0 \leq T) = \frac{1}{\sqrt{\pi}} \int_{z\geq \frac{\delta - cT}{\sqrt{2T}}} \exp(-z^2)dz + \frac{e^{2\delta c}}{\sqrt{\pi}} \int_{z\geq \frac{\delta + cT}{\sqrt{2T}}} \exp(-z^2)dz \tag{4.4}$$

4.4 is a consequence of the results presented in Section 2.3.3, Theorem 2.

4.5.2 SON for automatic bitrate video

We propose an E2E SON that gives the video bitrate represented by r in (4.3). This SON calculates via using a classical binary seach algorithm the optimal r that satisfies:

$$\mathbb{P}(H_0 \leq T) = \epsilon \tag{4.5}$$

where ϵ is a fixed arbitrary limit, which represents the starvation probability accepted by an operator for the E2E connection with video streaming play-out. We can see in (4.4) that $\mathbb{P}(H_0 \leq T)$ depends on r, m, σ, B_0 and T. r is found by the SON; m and σ are parameters depending on the TCP traffic which are estimated by the SON function using standard statistical techniques. B_0 is the quantity of data downloaded by the buffer during the buffering phase before the play-out of the video; finally T is the duration in seconds of the video. Figure 4.4 represents the functional architecture of the SON function controlling the starvation probability.

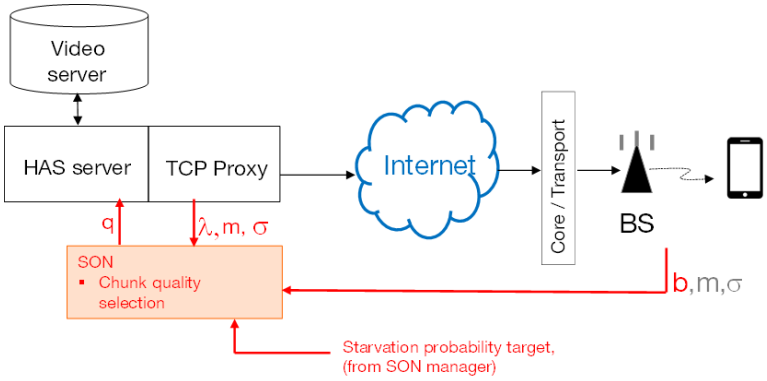


Figure 4.4: Architecture proposed for the SON function that controls the starvation probability

4.6 Numerical experiments

This Section assesses the different analytical approximations by comparing them to simulation results.

4.6.1 TCP average throughput

Figure 4.5 compares the analytical and simulated results for the average throughput of a TCP AIMD connection. One can see a very good agreement between the two approaches. Figure 4.6 shows that the analytical approximation of the average throughput coincide with the simulation with a TCP CUBIC connection. Figure 4.7 shows analytical approximation of σ^2 coincides with the simulated results using a TCP AIMD connection.

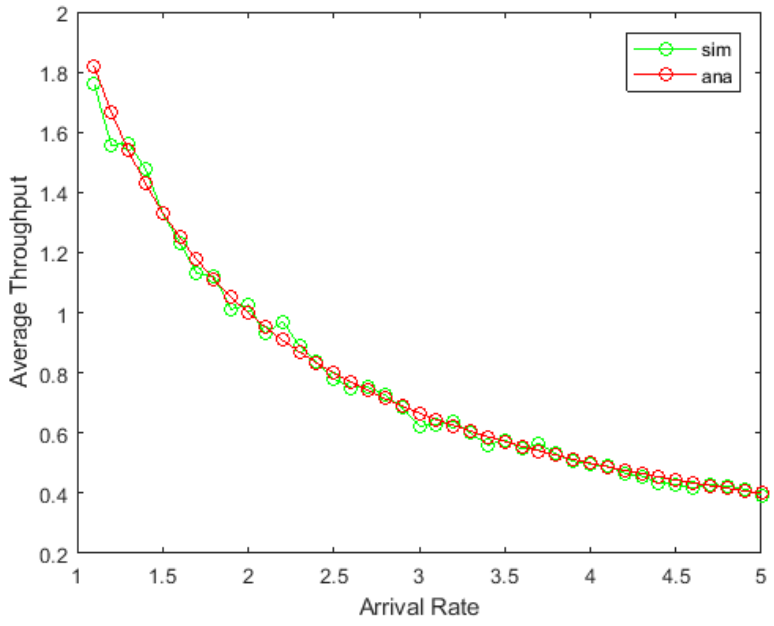


Figure 4.5: AIMD average throughput as a function of arrival rate for the analytical approximation (in red) and simulated results (in green)

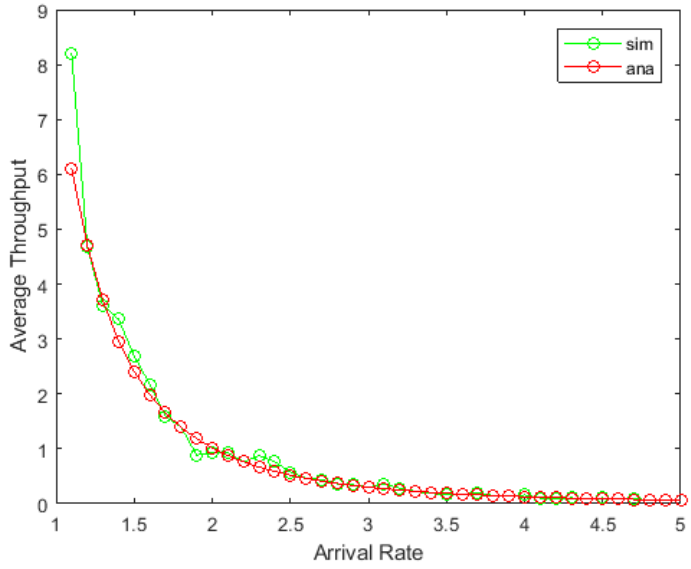


Figure 4.6: CUBIC average throughput as a function of arrival rate for the analytical approximation (in red) and simulated results (in green)

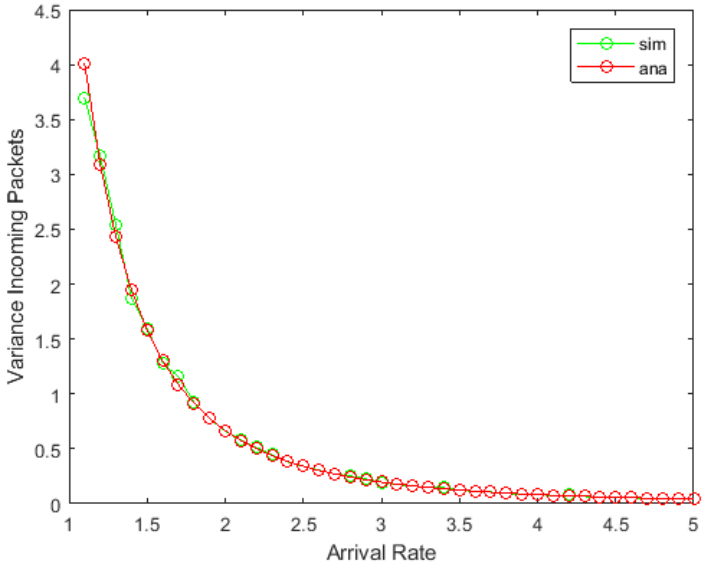


Figure 4.7: σ^2 as a function of arrival rate for the analytical approximation (in red) and simulated results (in green)

4.6.2 Starvation probability

Figure 4.8 compares the analytical approximation and the simulated results for the starvation probability. One can see that the analytical approximation is close to the simulated results.
The simulated results are obtained by simulating several times an E2E connection. We calculate the number of simulations in which at least one starvation event occurs divided by the total numner of simulations. The starvation probability is obtained using the ratio of the two numbers.

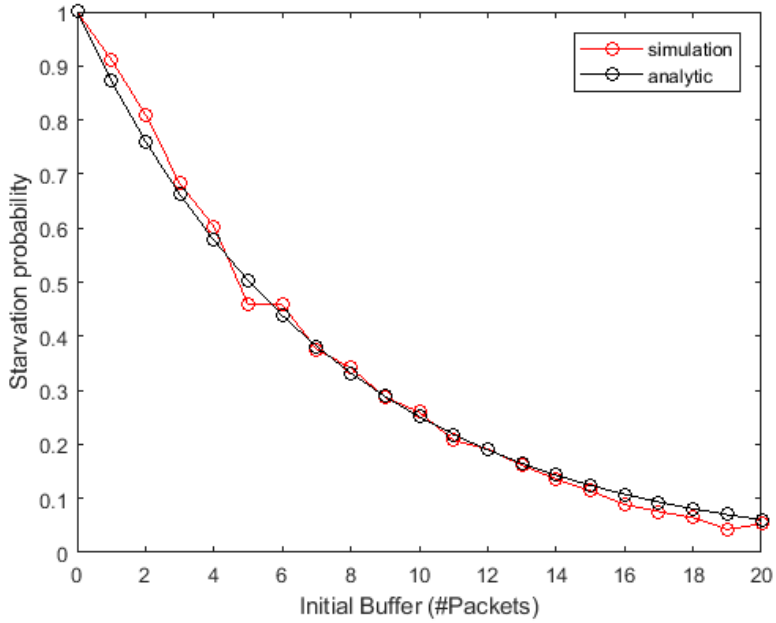


Figure 4.8: Starvation probability: Diffusion approximation vs simulation.

4.6.3 Adaptive bitrate video straming

In this Subsection, we want to show how one can exploit the starvation closed form formula to control the starvation probability by adapting the video bit rate. The video bitrate is fixed with the goal to reach a starvation probability equal to a desired target value ϵ. The bisectrix blue line in Figure 4.9 represents the otimal control, namely the simulated starvation probability is equal to the target probability. The red line corresponds to the starvation probability simulated with a video bitrate obtained solving (4.5).

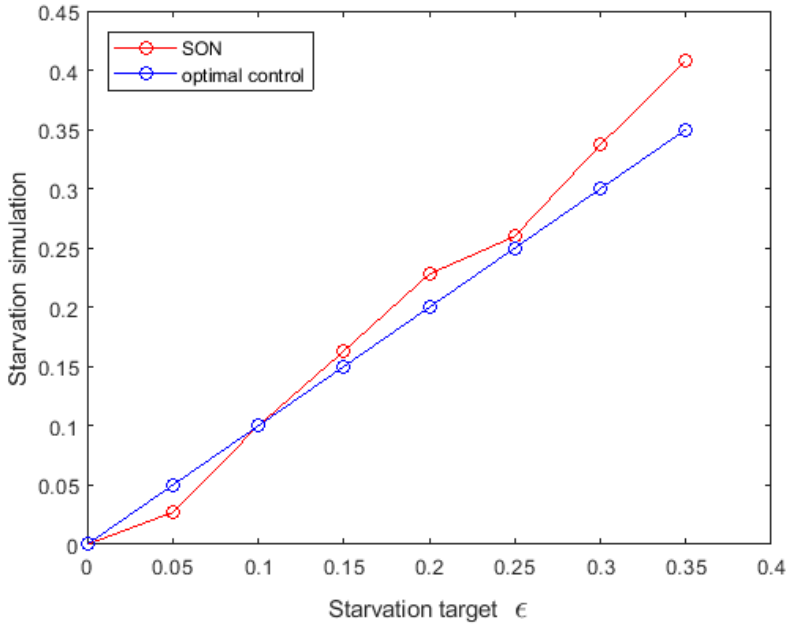


Figure 4.9: HAS control: starvation probability vs starvation target

It is noted that the results in Figure 4.9 assume that the SON function adapts the rate once, namely in the beginning of the video transmission. However, the rate can be adjusted several times during the video, e.g. after the reception of each chunk. It is recalled that a chunk duration can be of the order seconds (e.g. 10 seconds in [8]).

4.7 Conclusion

This Chapter has presented contributions on modeling and control of E2E connection of video streaming service. We have proposed a drifted Brownian model describing the behavior of buffer content receiving packets through a TCP connection. Using this result, we have provided a closed form expression for the starvation probability which is the key QoE indicator for the video streaming service. We have compared the analytical result for the starvation probability with numerical simulations and have shown a good agreement between the two approaches. Using the previous results, we have proposed a SON function that controls the starvation probability by adapting the video quality requested from the video server.

5

Conclusion and perspectives

5.1 Conclusion

This thesis aimes at contributing to E2E performance and QoS optimization. This challenge has been tackled in two parts. In the first we have studied the design, modeling and performance evaluation of Massive MIMO with hierarchical beamforming. We have developed a scheduling algorithm with α-fair allocation having a closed form and a very low computational complexity. Flow-level dynamics where users arrive and depart dynamically have been taken into account using queuing models. Next we have proposed closed-form formulas for flow level performance, for both elastic (with either proportional fairness and max-min fairness) and streaming traffic. We have assessed the performance of hierarchical beamforming using numerical experiments.
In the second part, we have developped a SON function to control E2E connections for video streaming traffic. The key QoE for video streaming is the starvation probability. The SON function aims at minimizing the starvation probability of the play-out buffer. Using analytical model for TCP connection on the one hand, and Brownian drifted model for the buffer content evolution on the other hand, we have derived a closed formula that relates the TCP parameters, the buffer state and the video quality to the starvation probability. We have then validated the closed form formula with numerical simulations. Finally we have developped a SON function that selects the video quality from the video server to reach a desired starvation target. The SON can be activated several times during the transmission with the purpose to adapt when the traffic condition changes during the E2E connection.
We find a model that represents the behavior of a buffer of a smartphone that visualizes a video streaming play-out. Thank to the Brownian drifted model we can know, considering the TCP parameters of the E2E connection, how to dynamically set a bitrate video that permits to reach an arbitrary probability of starvation during the transmission. These results are the first that show how to exploit parameters from the transport layer to control QoE at the application layer.

5.2 Perspectives

The PhD work opens the door to several perspectives. From the hierarchical beamforming perspective, the automatic design of the beams at the different levels comprising the codebook remains a challenging task. More particularly, how can one design the beams using a small set of geometrical parameter of the cells, i.e. antenna height, the cell width and length.
A second challenge is the automatic adaptation of the shape of the cell covered by the codebook of beams. Such a SON function can be applied to a massive MIMO with hierarchical codebook, but also to a standard Massive MIMO system. Interestingly, such a SON function recalls the automatic cell planning used in previous radio access generations without MIMO.
The E2E QoE/QoS control perspective, further improvements can be obtained by combining several SON functions along the E2E connection. One example can be a SON function that identifies context change of a UE (e.g. from outdoor to indoor or a car getting into a tunnel), and informs the scheduler to start an aggressive buffering to delay as much as possible starvation. An orchestrator is then needed to activate the different SON functions in-order to achieve a global optimal operation.

Bibliography

[1] 3GPP. "study on new radio (NR) access technology - physical layer aspects - release 14". *TR 38.802*, 2017.

[2] I. Adan and J. Resing. *Lecture Notes on Queueing Theory*. Department of Mathematics and Computing Science Eindhoven University of Technology, 2015.

[3] A. Alkhateeb, O. El Ayach L. Geert, and R. Heath. Channel estimation and hybrid precoding for millimeter wave cellular systems. *Journal of Selected Topics in Signal Processing*, 8(5), 2014.

[4] E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. *Transaction Network*, 30(4), 2005.

[5] F. Baccelli, P. Brémaud, and F. Papangelou. *Elements of Queueing Theory Palm Martingale Calculus and Stochastic Recurrences*. Metrika, 1996.

[6] F. Baccelli and D. McDonald. A square root formula for the rate of non-persistent TCP flows. *NGIN*, 2005.

[7] L. Bajzik, T. Kárász, Z. Vincze, C. Vulkán, W. Ben Ameur, Z. Altman, and V. Diascorn. SON for mobile backhaul. *WCNCW*, 2018.

[8] W. Bao and S. Valentin. Bitrate adaptation for mobile video streaming based on buffer and channel state. In *ICC*, 2015.

[9] W. Bao, V. WS Wong, and V. CM Leung. A model for steady state throughput of TCP cubic. *GLOBECOM*, 2010.

[10] F. Boccardi, B. Clerckx, A. Ghosh, E. Hardouin, G. Jongren, K. Kusume, E. Onggosanusi, and Y. Tang. Multiple-antenna techniques in LTE-advanced. *Communications Magazine*, 50(3), 2012.

[11] T. Bonald and A. Proutière. Wireless Downlink Data Channels: User Performance and Cell Dimensioning. In *Mobicom*, 2003.

[12] M. Bouzian. *Modeling and optimization of the quality of customer experience QoE of data services on the mobile network. Application to video streaming.* PhD thesis, Université Côte d'Azur, 2017.

[13] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. TCP vegas: New techniques for congestion detection and avoidance. *SIGCOMM*, 1994.

[14] U. Charash. Reception through nakagami fading multipath channels with random delays. *Transactions on Communications*, 27(4), 1979.

[15] R. Combes, Z. Altman, and E. Altman. Self-organizing fractional power control for interference coordination in OFDMA networks. *ICC*, 2011.

[16] R. Combes, Z. Altman, and E. Altman. Self-organization in wireless networks: A flow-level perspective. 2012.

[17] R. Combes, Z. Altman, and E. Altman. Self-organizing relays: Dimensioning, self-optimization and learning. *Transactions on Network Management*, 9(4), 2012.

[18] T. Daher, S. Ben Jemaa, and L. Decreusefond. Cognitive management of self — organized radio networks based on multi armed bandit. *PIMRC*, 2017.

[19] T. Daher, S. Ben Jemaa, and L. Decreusefond. Linear ucb for online SON management. *VTC Spring*, 2018.

[20] M. T. Diallo. *Quality of experience and video services adaptation.* PhD thesis, Institut National des Télécommunications, 2015.

[21] M. Dirami and Z. Altman. A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks. *WiOpt*, 2010.

[22] Ericsson. "5G new radio: designing for the future". *Ericsson Technology Review*, 2017.

[23] J. Floquet, R. Combes, and Z.Altman. Hierarchical beamforming: Resource allocation, fairness and flow level performance. *PEVA*, 2018.

[24] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *SIGOPS*, 2008.

[25] B Héder, P Szilágyi, and C Vulkán. Dynamic and adaptive QoE management for OTT application sessions in LTE. In *PIMRC*, 2016.

[26] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh. Millimeter wave beamforming for wireless backhaul and access in small cell networks. *Transactions on Communications*, 61(10), 2013.

[27] H. Jamal and K. Sultan. Performance analysis of TCP congestion control algorithms. *International journal of computers and communications*, 2(1), 2008.

[28] H. Ji, Y. Kim, J. Lee, E. Onggosanusi, Y. Nam, J. Zhang, B. Lee, and B. Shim. Overview of full-dimension MIMO in LTE-advanced Pro. *Communications Magazine*, 55(2), 2017.

[29] J. Kaufman. Blocking in a shared resource environment. *Transactions on Communications*, 29(10), 1981.

[30] A. Khlass, T.Bonald, and S.E. Elayoubi. Flow-level modeling of multi-user beamforming in mobile networks. In *WiOpt*, 2014.

[31] A Langley, A Riddoch, A Wilk, A Vicente, C Krasic, D Zhang, F Yang, F Kouranov, I Swett, J Iyengar, et al. The QUIC transport protocol: Design and internet-scale deployment. In *SIGDC*, 2017.

[32] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta. Massive mimo for next generation wireless systems. *Communications Magazine*, 52(2), 2014.

[33] D. J. Leith, R. N. Shorten, G. McCullagh, J. Heffner, L. Dunn, and F. Baker. *Delay-based AIMD congestion control*. eprint maynoothuniversity, 2007.

[34] Y. Lin, E. M. R. Oliveira, S. Ben Jemaa, and S. E. Elayoubi. Machine learning for predicting QoE of video streaming in mobile networks. *ICC*, 2017.

[35] Y. Liu and J. Y. B. Lee. A unified framework for automatic quality-of-experience optimization in mobile video streaming. In *INFOCOM*, 2016.

[36] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang. An overview of massive mimo: Benefits and challenges. *Journal of Selected Topics in Signal Processing*, 8(5), 2014.

[37] S Mekki, T Karagkioules, and S Valentin. HTTP adaptive streaming with indoors-outdoors detection in mobile networks. In *INFOCOM WKSHPS*, 2017.

[38] M. Mitsumura, H. Masuyama, S. Kasahara, and Y. Takahashi. Buffer-overflow and starvation probabilities for video streaming services with application-layer rate-control mechanism. *Queueing Theory and Network Applications*, 2011.

[39] J. Mo and J. Walrand. Fair End-to-End Window Based Congestion Control. *Transactions on Networking*, 8(5), 2000.

[40] Peter Mörters and Yuval Peres. *Brownian motion*. Cambridge University Press, 2010.

[41] S. Noh, M. D. Zoltowski, and D. J. Love. Multi-resolution codebook based beamforming sequence design in millimeter-wave systems. In *GLOBECOM*, 2015.

[42] J. Olsén. *Stochastic modeling and simulation of the TCP protocol*. PhD thesis, Avdelningen för matematisk statistik, 2003.

[43] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren. Scenarios for 5G mobile and wireless communications: the vision of the metis project. *Communications Magazine*, 52(5), 2014.

[44] Yassine ouga, Malika Bourenane, Abdelhamid Mellouk, and Yassine Hadjadj-Aoul. Tcp based-user control for adaptive video streaming. *Multimedia Tools and Applications*, 75(18), 2016.

[45] S. Poojary, R. El-Azouzi, E. Altman, A. Sunny, I. Triki, M. Haddad, T. Jimenez, S. Valentin, and D. Tsilimantos. Analysis of QoE for adaptive video streaming over wireless networks. *WiOpt*, 2018.

[46] S. Poojary and V. Sharma. Analytical model for congestion control and throughput with TCP cubic connections. *GLOBECOM*, 2011.

[47] S Poojary and V Sharma. An asymptotic approximation of TCP cubic. *arXiv*, 2015.

[48] J. Roberts. A service system with heterogeneous user requirements. *Performance of Data Communications systems and their applications*, 1981.

[49] L.Partzsch R.Schilling. *Brownian Motion - An Introduction to Stochastic Processes*. Walter de Gruyter GmbH & Co KG, 2014.

[50] L. Saker, S. Elayoubi, R. Combes, and T.Chahed. Optimal control of wake up mechanisms of femtocells in heterogenous networks. *JSAC*, 2012.

[51] C. B. Samios and M. K. Vernon. Modeling the throughput of TCP vegas. In *SIGMETRICS*, 2003.

[52] Saz. Density of first hitting time of brownian motion with drift. 2014. `https://math.stackexchange.com/questions/1053294/density-of-first-hitting-time-of-brownian-motion-with-drift`.

[53] R. Shorten, F. Wirth, and D. Leith. A positive systems model of TCP like congestion control: Asymptotic results. *Transaction Network*, 14(3), 2006.

[54] William J Stewart. *Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling*. Princeton University Press, 2009.

[55] A. Stolyar and H. Viswanathan. Self-organizing dynamic fractional frequency reuse for best-effort traffic through distributed inter-cell coordination. *INFOCOM*, 2009.

[56] L. B. Le T. E. Bogale. Beamforming for multiuser massive mimo systems: Digital versus hybrid analog-digital. *arXiv*, 2014.

[57] A. Tall, Z. Altman, and E. Altman. Multilevel beamforming for high data rate communication in 5G networks. In *arXiv*, 2015.

[58] A. Tall, Z. Altman, and E. Altman. Self-optimizing load balancing with backhaul-constrained radio access networks. *Wireless Communications Letters*, 4(6), 2015.

[59] A. Tall, R. Combes, Z. Altman, and E. Altman. Distributed coordination of self-organizing mechanisms in communication networks. *Transactions on Control and Network Systems*, 1(4), 2013.

[60] M.Feuillet T.Bonald. *Network Performance Analysis*. John Wiley and Sons, 2013.

[61] D. Tsilimantos, T. Karagkioules, A. Nogales-Gómez, and S. Valentin. Traffic profiling for mobile video streaming. *ICC*, 2017.

[62] D Tsilimantos, A Nogales-Gómez, and S Valentin. Anticipatory radio resource management for mobile video streaming with linear programming. *arXiv*, 2016.

[63] Z. Xiao, P. Xia, and X. G. Xia. Codebook design for millimeter-wave channel estimation with hybrid precoding structure. *Transactions on Wireless Communications*, 16(1), 2017.

[64] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control for fast long-distance networks. *INFOCOM*, 2004.

[65] Y. Xu, E. Altman, R. El-Azouzi, S. E. Elayoubi, and M. Haddad. QoE analysis of media streaming in wireless data networks. *International conference on Networking*, 2012.

[66] Y. Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, and T. Jimenez. Probabilistic analysis of buffer starvation in markovian queues. *INFOCOM*, 2012.

[67] Y. Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, and T. Jimenez. Analysis of buffer starvation with application to objective QoE optimization of streaming services. *Transactions on Multimedia*, 16(3), 2014.

[68] Y. Xu, S. E. Elayoubi, E. Altman, and R. El-Azouzi. Impact of flow-level dynamics on QoE of video streaming in wireless networks. *INFOCOM*, 2013.

[69] Y. Xu, S. E. Elayoubi, E. Altman, R. El-Azouzi, and Y. Yu. Flow-level QoE of video streaming in wireless networks. *Transactions on Mobile Computing*, 15(11), 2016.

[70] Y. Xu, Z. Xiao, H. Feng, T. Yang, B. Hu, and Y. Zhou. Modeling buffer starvations of video streaming in cellular networks with large-scale measurement of user behavior. *Transactions on Mobile Computing*, 16(8), 2017.

[71] Z. Ye. *Performance Analysis of HTTP Adaptive Video Streaming Services in Mobile Networks*. PhD thesis, Université d'Avignon, 2017.

Appendix A

Gain function

Considering a N_xxN_z (sub) antenna array of vertical dipoles. For sake of clarity and completeness of the chapter, we include the model [57] in the sequel. To simplify the model, the reflector is approximate as an infinite Perfect Electrical Conductor (PEC). The N_x and N_z elements in each row and column are equally spaced with distances dx and d_z respectively (see the following figure).
They use the spherical coordinates θ and ϕ to determine the angle (θ_e,ϕ_e) that represents the direction of a beam to obtain a maximum gain. Knowing these notations, the antenna gain of a beam in a direction (θ,ϕ) is written as

$$G(\theta, \phi, \theta_e, \phi_e) = G_0 f(\theta, \phi, \theta_e, \phi_e)$$

where f is a normalized gain function and G_0 the maximum gain in the (θ_e, ϕ_e) direction. (θ_e,ϕ_e) corresponds to the direction of a beam where it reaches the peak antenna gain. A separable excitation in the x and z directions is assumed, resulting in the following separable form of f:

$$f(\theta, \phi, \theta_e, \phi_e) = |AF_x(\theta, \phi, \theta_e, \phi_e) AF_y(\theta, \phi) AF_z(\theta, \theta_e)|^2 G_d(\theta)$$

$AF_x(\theta, \phi, \theta_e, \phi_e)$ and $AF_z(\theta, \theta_e)$ are the array factors in the x and z directions and are given by

$$AF_x(\theta, \phi, \theta_e, \phi_e) = \frac{1}{\sum_{m=1}^{N_x}} \sum_{m=1}^{N_x} w_m a_m$$

and

$$AF_z(\theta, \theta_e) = \frac{1}{\sum_{n=1}^{N_z}} \sum_{n=1}^{N_z} v_n b_n$$

The weights w_m and v_n for the radiating elements in the m-th row and n-th columns define a Gaussian tapering function used to control the sidelobe level of the gain pattern

$$w_m = \exp\big(-(\frac{L_m - \frac{L_m}{2}}{\sigma_x})^2\big)$$

$$v_n = \exp(-(\frac{L_n - \frac{L_n}{2}}{\sigma_z}^2)$$

where L_m and L_n are the array size in the x and z directions respectively with $L_m = (N_x - 1)d_x$ and $L_n = (N_z - 1)d_z$. The values for $\sigma_s, s \in \{x, z\}$, are defined by fixing the ratio between the extreme and center dipole amplitudes respectively to a given value of α_s:

$$\sigma_s^2 = -(\frac{L_s}{2})\frac{1}{\log(\alpha_s)}; s \in \{x, z\}$$

a_m and b_n are complex amplitude contributions of the radiating element located at $(m-1)d_x$ and $(n-1)d_z$, respectively:

$$a_m = \exp\big(-2j\pi\frac{(m-1)d_x}{\lambda}(\sin\theta\sin\phi - \sin\theta_e\sin\phi_e)\big)$$

$$b_n = \exp\big(-2j\pi\frac{(n-1)d_z}{\lambda}(\cos\theta - \cos\theta_e)\big)$$

The impact of the PEC can be modeled by replacing it with the images of the radiating elements it creates. The term $AF_y(\theta, \phi)$ takes into account the images and is written as

$$AF_y(\theta, \phi) = \sin(\frac{\pi}{2}\sin\theta\cos\phi)$$

The normalized gain pattern of the dipoles,$G_d(\theta)$, is approximated as $G_d(\theta) = \sin^3\theta$. The term G_0 is obtained from the power conservation equation:

$$G_0 = \frac{4\pi}{\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\int_0^\pi f(\theta,\phi)\sin\theta d\theta d\phi}$$

A beam is defined by the (rectangular) sub-array size, and the couple (θ_e, ϕ_e) defines its direction.

Appendix B

Definitions for trees

Given a directed tree $G = (V, E)$ we use the following terminology.

- Parent: v is a parent of v' iff $(v, v') \in E$,
- Child: v is a child of v' iff $(v', v) \in E$,
- Root: v is the root iff it has no parent.
- Leaf: v is a leaf if it has no children.
- Path: a path from v to v' of length ℓ is a set of vertices (v_0, v_1), $\ldots$,$(v_{\ell-1}, v_\ell)$ in E such that $v_0 = v$ and $v_\ell = v'$.
- Depth: the depth of v is the length of the path from the root to v.
- Descendant: v' is a descendant of v if there exists a path from v to v'
- Ancestor: v' is an ancestor of v if there exists a path from v' to v
- Height of the tree: $h(G)$ is the maximal depth of a leaf.
- Degree of the tree: $d(G)$ is the maximal number of children of a node.
- Regular tree: a tree is d-regular if all nodes have degree 0 (for leaves) or d (for non-leaves).

Appendix C

Numericals parameters

The following table present the parameters used for the simulation of traffic served by a BS Massive MIMO.

Table C.1: Network and Traffic characteristics

Network parameters	
Number of sectors with hierarchical beamforming	1
Macro Cell layout	hexagonal trisector
Antenna height	30 m
Bandwidth	20MHz
Channel characteristics	
Thermal noise	-174 dBm/Hz
Path loss (d in km)	128.1 + 37.6 $\log_{10}(d)$ dB
Traffic characteristics	
Service type	FTP
Average file size	8 Mbits

Appendix D

TCP Notation

All the protocols notation :

- W : discrete function representing the sliding window used by TCP
- S : discrete function threshold for indicating a change of algorithm from a slow start (SS) to congestion avoidance (CA)
- $Z(t)$: ternary value representing 0 if at time t a packet is time-out (To), 3 if (3)-ACK else 1.
- RTT : Round Trip Time
- FRt : (Fast retransmit) re-sending of dropped or duplicate ACK and change the regime in (SS)
- FRc : (Fast recovery) re-sending of dropped or duplicate ACK, wait for an ACK for all the packets of the precedent window size and continue with the regime in (CA)

Titre: Mécanismes auto-organisants pour connexions de bout en bout

Mots clés: Systèmes de files d'attente, évaluation des performances, planification, Massive MIMO, QoE, E2E SON

Résumé: Les réseaux de cinquième génération sont en cours de définition et leurs différentes composantes commencent à émerger: nouvelles technologies d'accès à la radio, convergence fixe et mobile des réseaux et virtualization.
Le contrôle et la gestion de bout en bout (E2E) du réseau ont une importance particulière pour les performances du réseau. Cela étant, nous segmentons le travail de thèse en deux parties: le réseau d'accès radio (RAN) axé sur la technologie MIMO Massif (M-MIMO) et la connexion E2E du point de vue de la couche transport.
Dans la première partie, nous considérons la formation de faisceaux focalisés avec un structure hiérarchique dans les réseaux sans fil. Pour un ensemble de flots donnée, nous proposons des algorithmes efficaces en terme de complexité pour une allocation avec α-équité. Nous proposons ensuite des formules exactes pour la performance au niveau du flot, à la fois pour le trafic élastique (avec une équité proportionnelle et équité max-min) et le trafic en continu. Nous validons les résultats analytiques par des simulations.
La seconde partie de la thèse vise à développer une fonction de réseau auto-organisant (SON) qui améliore la qualité d'expérience (QoE) des connexions en bout-en-bout. Nous considérons un service de type vidéo streaming et développons une fonctionnalité SON qui adapte la QoE de bout-en-bout entre le serveur vidéo et l'utilisateur. La mémoire-tampon reçoit les données d'un serveur avec une connexion E2E en suivant le protocole TCP. Nous proposons un modèle qui décrit ce comportement et nous comparons les formules analytiques obtenues avec les simulations. Enfin, nous proposons un SON qui donne la qualité vidéo de sorte que la probabilité de famine soit égale à une valeur cible fixée au préalable.

Title: Self-organizing mechanisms for end-to-end connections

Keywords: Queueing systems, performance evaluation, scheduling, Massive MIMO, QoE, E2E SON

Abstract: Fifth generation networks are being defined and their different components are beginning to emerge: new technologies for access to radio, fixed and mobile convergence of networks and virtualization.
End-to-end (E2E) control and management of the network have a particular importance for network performance. Having this in mind, we segment the work of the thesis in two parts: the radio access network (RAN) with a focus on Massive MIMO (M-MIMO) technology and the E2E connection from a point of view of the transport layer.
In the first part, we consider hierarchical beamforming in wireless networks. For a given population of flows, we propose computationally efficient algorithms for fair rate allocation. We next propose closed-form formulas for flow level performance, for both elastic (with either proportional fairness and max-min fairness) and streaming traffic. We further assess the performance of hierarchical beamforming using numerical experiments.
In the second part, we identify an application of SON namely the control of the starvation probability of video streaming service. The buffer receives data from a server with an E2E connection following the TCP protocol. We propose a model that describes the behavior of a buffer content and we compare the analytical formulas obtained with simulations. Finally, we propose a SON function that by adjusting the application video rate, achieves a target starvation probability.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France