The Dissertation Committee for Virag Shah
certifies that this is the approved version of the following dissertation:

Centralized Content Delivery Infrastructure Exploiting Resource Pools: Performance Models and Asymptotics

Committee:

Gustavo de Veciana, Supervisor

François Baccelli

Alex Dimakis

John Hasenbein

Sanjay Shakkottai

Centralized Content Delivery Infrastructure Exploiting Resource Pools: Performance Models and Asymptotics

by

Virag Shah, B.E.; M.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2015

Dedicated to fellow human beings.

Acknowledgments

gurur brahmā gurur viṣnuḥ
gururdevo maheśhwaraḥ
guruḥ sākṣāt parabrahma
tasmai śri gurave namaḥ

gurustōtram

The Sanskrit hymm roughly translates as follows: Guru symbolizes the universal principles of creation, existence, and evolution; salutations to that guru who embodies the very essence of reality. I am fortunate to find a guru in my advisor Prof. Gustavo de Veciana. I gladly acknowledge his contributions in development of this thesis. I have greatly benefited from his scientific acumen, clarity of thoughts, and extraordinary perception of what are the right questions. If not for his patience and constant support, I would likely not have persisted the ups and downs of PhD research and life.

I thank Prof. François Baccelli for being a mentor and a role model. His steadfast knowledge and mathematical perspective on engineering problems has influenced me greatly. Prof. Sanjay Shakkottai, for being a constant source of enthusiasm and for exposing me to the wonderful world of concentration inequalities. Professors Sujay Sanghavi, Alex Dimakis, John Hasenbein, Constantine Caramanis, Gordon Žitković, Mihai Sîrbu, and several other faculty members at UT who have influenced me in many ways through their excellent courses and several interesting discussions.

I would also like to thank Dr. Murali Kodialam and Dr. T. V. Lakshman for a wonderful summer internship at Bell Labs, New Jersey. Prof. George Kesidis at Penn State for being an inspiration and a great research collaborator. Prof. Neelesh B. Mehta at IISc Bangalore, Professors D. Manjunath and Bikash K. Dey at IIT Bomaby for providing me with solid research foundations prior to my PhD tenure and always being there as a source of guidance.

My stay at Austin has been exciting and fun thanks to my friends. Special thanks to Avik and Anamika for being all-weather friends, for organizing wonderful trips and movies, and for teaching me how to drive. I will remember with fondness the intense discussions with Ankit, Pravesh, Sarabjot, and Srinadh; they have something to debate on all topics under the sun and never seem to get tired. Jiaxiao Zheng has been a very helpful cubicle-mate and a patient listener for several technical discussions. Sreangsu and Suriya, thank you for taking away countless hours from my research time, the distractions were a pleasure and an added motivation to visit lab regularly. I have made so many friends at Austin, who are all so kind that I am sure they will forgive me for not listing them here.

Finally, I will always be indebted to my family: mom, dad, Shaili, Chirag, Nirali, and Niv for the constant love and support.

Centralized Content Delivery Infrastructure Exploiting Resource Pools: Performance Models and Asymptotics

Publication No. ________________

Virag Shah, Ph.D.
The University of Texas at Austin, 2015

Supervisor: Gustavo de Veciana

We consider a centralized content delivery infrastructure where a large number of storage-intensive files are replicated across several collocated servers. To achieve scalable delays in file downloads under stochastic loads, we allow multiple servers to work together as a pooled resource to meet individual download requests. In such systems basic questions include: How and where to replicate files? How significant are the gains of resource pooling over policies which use single server per request? What are the tradeoffs among conflicting metrics such as delays, reliability and recovery costs, and power? How robust is performance to heterogeneity and choice of fairness criterion? In this thesis we provide a simple performance model for large systems towards addressing these basic questions. For large systems where the overall system load is proportional to the number of servers, we establish scaling laws among delays, system load, number of file replicas, demand heterogeneity, power, and network capacity.

Table of Contents

List of Figures

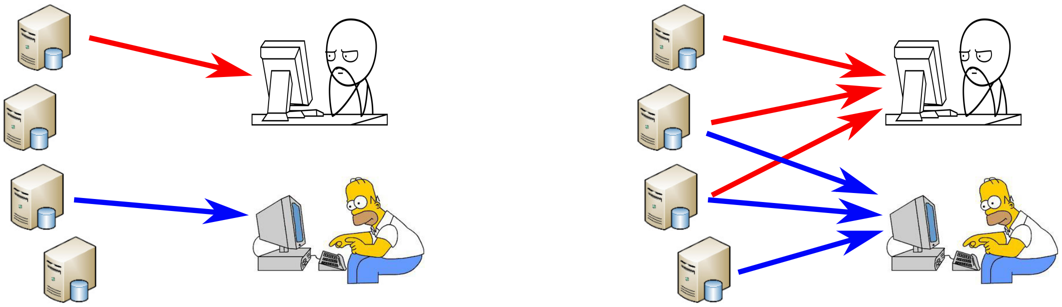
Chapter 1

Introduction

Suppose one is to design a centralized content delivery infrastructure with thousands of collocated servers with a goal to store and quickly deliver large files such as software updates, scientific datasets, 3D videos, etc. Suppose that these files are too large and diverse to be held in main memory and are thus stored on disk. Further, the demands for these files are high and dynamic so you replicate them across multiple servers at the cost of increasing storage space. In delivering these files to users, which of the following approaches would one rather choose?

(i) Single server allocation: Each file download request is served by a single server, see Fig. 1.1a. For example, among the servers which store the file, the request is routed to a server which is serving the least number of download requests. Alternatively, one could centralize and defer routing decisions to times when servers become idle.

(ii) Pooling of server resources: Each file download request is served by multiple servers in parallel thus pooling their resources, see Fig. 1.1b. In this setting different chunks of the file may be downloaded concurrently from different servers. If the server pools for different file download requests overlap



(a) Single server allocation

(b) Pooling of server resources

Figure 1.1: An illustration of two different content delivery approaches.

then the requests share the server resources according to an appropriate fairness criterion.

Intuitively, in an idealized situation where there is only one active request in the system and no network bottlenecks, resource pooling would provide much better download speed as compared to single server allocation. However, in a more realistic setting where download requests arrive dynamically and are served by overlapping pools of servers, the gains of resource pooling over single server allocation are not directly clear.

In this dissertation we investigate the gains of resource pooling in such a dynamic setting with a particular focus on a scaling regime where the system size (total number of servers) and the overall system load scale proportionally. In this setting, we show that resource pooling is indeed effective and outperforms single server allocation. This is not unexpected. However what is interesting is how these performance gains scale in the size of server pools and loads. For example, the delays in downloading files scale inversely with the size of the server pools. In addition, under appropriate load-balancing/fairness criterion across classes of download jobs, the delays are robust to limited heterogeneity in file demands and system components, and are also less sensitive to increases in overall system load. We also investigate the impact on delays of resources such as power, memory, and certain network bottlenecks. The overarching thesis for this dissertation is as follows:

Thesis Statement: *Pooling of server resources in a large-scale centralized content delivery systems can achieve scalable and robust performance in large file downloads.*

Below, we describe the key questions we explore in this dissertation and give a brief summary of our results towards addressing them.

Key Questions and Summary of Contributions:

We ask a sequence of questions, starting from a basic one which assesses relevance of pooling, and then gradually investigating more complex scenarios. Each question is followed up by a brief summary of our contribution towards addressing it.

Question 1. *How significant are the gains of resource pooling as compared to single server allocation? How do they fare when evaluated across conflicting metrics such as delays, reliability and recovery costs, and power?*

To address this question, we develop a simple performance model for systems using resource pooling in Chapter 2. We start by providing a system model which captures arbitrary static placement of files across servers, dynamic arrivals and service of file download requests, and a resource allocation policy employing pooling of servers. We develop an exact expression for mean delay for such a system. As one might expect, the expression is somewhat complex. However it simplifies under symmetry, and, perhaps surprisingly, it takes a clean and transparent form in an asymptotic regime where system size and load scale proportionally. We use this asymptotic expression to compare our policy with other known policies and to study system tradeoffs. We show that our resource allocation policy achieves blanket improvement over other policies. A version of this work is to appear in [48].

The above mentioned results are based on a specific resource allocation/fairness criterion across server pools, namely balanced fairness, which is amenable to mean delay analysis under stochastic loads. A natural question thus arises: how important is the choice of fairness criterion? Another natural question is: how robust are these results to symmetry assumptions? These are important questions, not only for conclusions regarding resource pooling, but also in practice. For example, achieving robustness to asymmetries in file demands and heterogeneities in system components in large systems via resource pooling implies a scalable approach towards addressing the delivery of popular content without requiring complex caching strategies. Also, for dynamic systems/networks, the basic problem of linking fairness in resource allocation to job delays has remained largely open in spite of significant research efforts.

Question 2. *What is the impact of heterogeneity and fairness criterion for* *resource allocation on job delays in large scale content delivery systems?*

We address this question in Chapter 3 where we provide new performance comparison results for following fairness criteria: α-fair (including max-min and proportional fair), Balanced fair, and Greedy. We also provide an explicit mean delay bound for large systems with heterogeneous servers and limited heterogeneity in file demands. Our results exhibit robustness of delays to limited types of heterogeneity. In the process, we establish the asymptotic symmetry of large randomly configured (random file placement) systems with heterogeneous components in an appropriate asymptotic regime. A version of this work appeared in [49].

Our next question pertains to the memory requirements per server. Note that to be able to pool c servers to serve a file download request we required replication of the file across at least c servers. This seemingly implies a tradeoff between job delays and memory requirement. However, one can perhaps do better; in particular, there may be a replication strategy such that scalable delays are achieved without scaling memory requirements. Further, the size of server pools may be constrained not only by the file-replication/memory but also by a parallelism constraint which limits the maximum number of servers one can use in parallel.

Question 3. *Given a constraint on the maximum number of servers that can serve a job in parallel, what are the tradeoffs between memory and job delays?*

We investigate this question in Chapter 4. Here we consider splitting

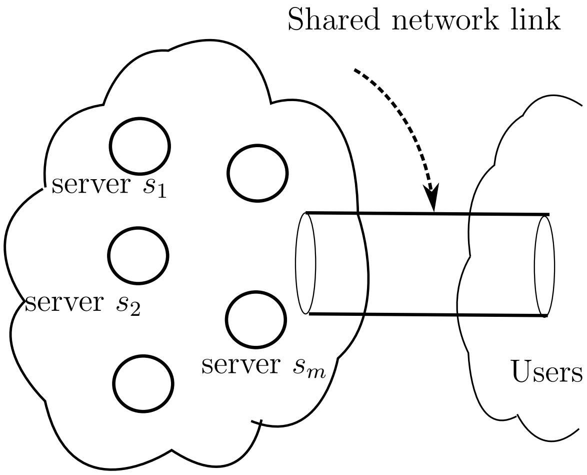


Figure 1.2: A centralized content delivery system where collocated servers are connected to users via a shared network link.

of a file into multiple blocks before replication. This allows us to reduce the memory requirement while effectively achieving larger server pools, but at a loss of ability to download a given chunk from any server in the pool. We provide a policy which mitigates the impact of this loss, thus achieving scalable delays without scaling memory requirement.

The final question addressed in this thesis concerns the impact of power capacity and network bottlenecks. Indeed the finite capacity of shared network links or a cap on overall power draw at the infrastructure may constrain the download speeds of jobs and potentially reduce the gains from server pooling. We consider impact of finite download capacity on the user side, and also the impact of a shared network link(s) on the content delivery system side, see Fig 1.2.

Question 4. *What is the impact on job delays of a network link and power* *capacity shared by the servers at the infrastructure? Also, how does the finite download capacity on users' end impact their performance?*

We address these questions in Chapter 5. The answer to the former question is perhaps surprising. For large scale systems, we show a concentration result in the number of active servers when the server pools are of limited size. Using this result we show that if the capacity of the shared network link is close to (and slightly larger than) the average traffic demand, its impact on user performance will be negligible as the system scales. Similarly, if the peak power capacity is close to average power consumption, the risk of overload with adverse impact on performance becomes low as system becomes large.

The impact of user's download capacity is a bit subtle, in that, it depends on overall system load. If the system is lightly loaded then the download capacity may become a dominant bottleneck and may drive the user's performance. However, if the overall system load increases beyond a threshold then the servers become a dominant bottleneck and the impact of download capacity becomes negligible.

In summary this thesis is devoted to the analysis of performance of a class of systems with resource pooling that might be suitable to meet future demands for large files in high capacity networks. Chapter 6 presents concluding remarks.

Chapter 2

Performance Model under Resource Pooling

We consider a centralized infrastructure which stores and delivers large files such that delay to serve a download request is scalable with traffic loads. Such centralized infrastructure could, for example, be part of a larger distributed content delivery network, where requests not currently available at distributed sites are forwarded to the centralized infrastructure which in turn delivers the files to the remote sites and/or users. Performance in such systems is the result of a complex interaction among requests that come and go dynamically and the pools of resources that are able to serve them. As traffic loads increase, one can make the following design choices to meet performance requirements: 1) dimensioning of system's server and network resources; 2) (possibly random) placement of data across servers; and 3) policy for routing/servicing requests. In this chapter we develop a robust large-scale performance model to enable system-level optimization with respect to these design choices.

We also aim to study tradeoffs among conflicting goals in such systems,

[1] A version of this work is to appear as "High Performance Centralized Content Delivery Infrastructure: Models and Asymptotics," in IEEE/ACM Transactions on Networking. This is a joint work with Prof. Gustavo de Veciana.

e.g., 1) service capacity available to end users and the resulting perceived performance; 2) reliability and recovery costs; and, 3) energy costs. For example, by increasing the total number of active servers, or scaling the speed of individual servers, one can tradeoff energy cost with performance. A more subtle example, discussed further in the sequel, involves spreading multiple copies of files across pools of servers so as to trade off the cost in recovery from large-scale server loss events, e.g., power outages [14], with performance.

Our contributions. The key challenge we tackle in this chapter is the performance evaluation of large scale storage systems wherein multiple file copies are placed across pools of servers and are subject to stochastic loads. We consider a system model where arriving file-requests/download-jobs can be collectively served by servers, i.e., different chunks of each file can be downloaded in parallel from servers currently storing the file – this is akin to peer-to-peer systems. Since each server can store multiple files, which are themselves replicated across sets of servers, the service capacities available to serve requests for different files are dynamically coupled. Indeed, as explained in the sequel, ongoing file requests can share server capacity subject to various possible 'fairness' objectives rendering performance evaluation quite challenging.

The main analytical contributions of this chapter can be summarized as follows. Firstly, we propose a file-server model and show that the overall service capacity set has polymatroid structure. We combine this structural result of an achievable capacity region with dynamic balanced fair rate allocations (described later) to develop an explicit expression for the mean file transfer delay experienced by file requests. Secondly, we prove a new asymptotic result for *symmetric* large-scale systems wherein the distribution of the number of waiting file requests concentrates at its mean. This result provides an easily computable approximation for the mean delay which is used to quantify system tradeoffs.

Finally, these analytical results are used to develop and quantify three key insights regarding large file-server systems:

a) We show how dynamic service capacity allocation across ongoing demands is impacted by the structure of overlapping resource pools (file placement) and quantify the substantial performance benefits over simpler load balancing strategies such as those assigning file requests at random or to least loaded servers.

b) We show that performance gains resulting from the overlapping of server pools, although significant, quickly saturate as one increases the overlap. This enables engineering of such systems to realize close to optimal performance while simultaneously achieving high reliability and thus low recovery costs.

c) For a simple speed scaling policy where the processor runs at low speed (or halts) when idle and a high but fixed speed when busy, we show that dynamic service capacity allocation can achieve up to 70% energy saving as compared to simpler policies.

2.1 Related work

There are several large-scale performance models applicable to content delivery systems. For example, the super-market queueing model studied in [12, 41, 43, 56] captures a policy where each arriving request is assigned to the least loaded server among those able to serve it. It is known to have better mean delay performance and tail decay for the distribution of the waiting jobs as compared to the policy of routing requests randomly among the possible servers. Alternatively, one can make centralized scheduling decisions as servers become available [37,60]. In [37] a greedy policy is shown to be optimal over all scheduling disciplines in a heavy-traffic regime. A centralized policy is studied in [60] and is shown to have robustness properties with respect to limited heterogeneity in loads across different file types. The key difference between these works and ours is that, rather than assigning a file request to a single server, we allow it to be served by multiple servers simultaneously. In the sequel, we evaluate the benefits of doing so.

Pooling of server resources is similar in spirit to multipath routing in wireline networks, see e.g. [24, 25, 27, 31, 59]. A multipath TCP architecture is proposed in [59] to achieve network wide resource pooling. Studies of the benefits of multipath routing have been previously carried out, e.g., in [31] the authors show the benefits of coordinating rate over multiple paths in terms of the worst case rate achieved by users in a static setting. For networks with stochastic loads, performance analysis under multipath transport is in general hard; [25, 27] study role of resource pooling in such a setting and provide performance bounds/approximations. Resource pooling in networks via multipath, and that in content delivery infrastructure via pooling of servers may eventually complement each other to achieve scalable performance gains.

There has also been previous work considering file placement across servers [33,34,45,65]. For example, [33] studies file placement across servers so as to minimize 'bandwidth inefficiency' when there is a fixed set of file transfer requests. Further, [34, 45] consider the problem of adaptive replication of files for a loss network model where each server can serve one file request at a time, thus avoiding queuing. The focus of these works is on caching popular files via distributed content delivery networks. In turn, they rely on a centralized infrastructure to handle cache misses and request denials arising when all associated servers are busy. Another line of work has focused on online packing/placement of dynamically arriving files/objects under constraints on available resources, e.g., [50]. By contrast with these works, we assume file placements across servers are fixed and we examine the performance impact of this when the system is subject to stochastic loads with no loss.

There are several works in the literature studying energy-performance tradeoffs, see e.g., [22, 35] and citations therein. In [22], the authors provide an approximation to the number of servers that should be active so as to optimize the energy-delay product. Similarly, [58] investigates speed scaling so as to optimize a weighted average of energy and mean delay for a single server system. In [35], the authors consider energy costs of switching servers on and off and provide an optimal online algorithm to optimize overall convex cost functions that can include performance and energy costs. In these works a server can handle any job request. By contrast in this chapter we are particularly interested in the situations where servers' capabilities are constrained (e.g., by the files they have available) and the coupling across server pools critically impacts energy-performance tradeoffs.

As will be discussed in more detail below this chapter draws on, and extends, previous work on bandwidth sharing models; in particular "balanced fair" allocations, see e.g., [6, 7, 11]. Such allocations are a useful device in that they are amenable to analysis, are provably insensitive to job size distribution, and yet serve to approximate various forms of 'fair' resource sharing policies considered in the literature and in practice [5, 6, 39].

Organization of the chapter. In Section 2.2 we develop our system model for file server systems under stochastic loads. In Section 2.3 provide an exact analysis for mean delay in file transfers under balanced fair resource allocation. In Section 2.4 we consider large scale systems and provide an asymptotic expression for the mean delay. In Section 2.5 we use our analysis to compare the performance of our policy with other resource allocation policies. In Section 2.6 we discuss system tradeoffs involving mean delay, recovery costs and energy consumption. Some of the proofs are provided in the Appendix.

2.2 System model

Consider a bipartite graph $G = (F \cup S; E)$ where F is a set of n files, S is a set of m servers, and each edge $e \in E$ connecting a file $i \in F$ and server

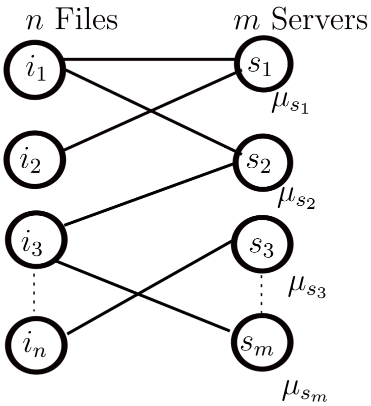


Figure 2.1: Graph $G = (F \cup S; E)$ modeling replication of n files across m finite capacity servers in a content delivery infrastructure.

$s \in S$ implies that a copy of file i is replicated at server s, see Fig 5.1. For each node $s \in S$, let N_s denote the set of neighbors of server s, i.e., the set of files it stores. Similarly, for each file $i \in F$ let S_i denote the set of servers that store file i. Further, for each $A \subset F$ let $S(A) = \cup_{i \in A} S_i$. Suppose that each server $s \in S$ has a peak service capacity of μ_s bits per second. For each $A \subset F$ let

$$\mu(A) = \sum_{s \in S(A)} \mu_s,$$

i.e., $\mu(A)$ is the sum rate at which requests for files in set A can be served.

Requests for file $i \in F$ arrive according to an independent Poisson process with rate λ_i. We shall use the terms request and job interchangeably. Similarly, we refer to each file $i \in F$ as a file or a job class interchangeably. Service requirements for jobs in class $i \in F$ are i.i.d with mean ν_i. Let $\boldsymbol{\rho} = (\rho_i : i \in F)$, where $\rho_i = \lambda_i \nu_i$ denotes the load associated with class i.

Jobs arrive to the system at total rate $\sum_{i \in F} \lambda_i$. Let $q_i(t)$ denote the *set* of ongoing jobs of class i at time t, i.e., jobs which have arrived but have not completed service, and $\mathbf{q}(t) = (q_i(t) : i \in F)$. For each $A \subset F$, let $q_F(t) = \cup_{i \in F} q_i(t)$, i.e., the set of all active jobs in the system. Let $\mathbf{x}(t) = (x_i(t) : i \in F)$, where $x_i(t) \triangleq |q_i(t)|$, i.e., $\mathbf{x}(t)$ captures the *number of ongoing jobs in each class.* Let $\mathbf{X}(t)$ correspond to the random vector describing the state of the system at time t.

For any $\mathbf{x}(t)$, let $A_{\mathbf{x}(t)}$ denote the set of active classes, i.e., classes with at least one ongoing job. Further, for each $s \in S$, let $Y_s(t) = \mathbf{1}_{\left\{s \in S(A_{\mathbf{X}(t)})\right\}}$. If $Y_s(t)$ is 1 we say that the server is active at time t.

For each $v \in q_i(t)$ and $s \in S$, let $b_{v,s}(t)$ be the rate at which server s serves job v at time t. Let $b_v(t)$ be the total rate at which job v is served at time t. If job v arrives at time t_v^a and has service requirement η_v, then it departs at time t_v^d such that $\eta_v = \int_{t_v^a}^{t_v^d} b_v(t)dt$.

Our service model is subject to the following assumption.

Assumption 1. *Sharing of system service capacity among ongoing jobs is such that:*

1. *A server s can concurrently serve multiple jobs as long as $\sum_v b_{v,s}(t) \leq \xi$ for all t.*

2. *Multiple servers can concurrently serve a job v at time t giving a total service rate $b_v(t) = \sum_s b_{v,s}(t)$.*

3. *The service rate $b_{v,s}(t)$ allocated to a job v at server s at time t depends only on its job's class and the numbers of ongoing jobs $\mathbf{x}(t)$. Thus, for each i, the jobs in $q_i(t)$ receive equal rate at time t which depends only on $\mathbf{x}(t)$.*

Allowing multiple servers to concurrently serve a job is reminiscent of service model in P2P systems [61, 65] which consists of a set of users/peers connected through the Internet, collectively sharing their files/resources. In this thesis, however, our focus is on modeling a centralized infrastructure aimed at quickly serving large files.

Let $r_i(\mathbf{x}')$ be the total rate at which class i jobs are served at time t when $\mathbf{x}(t) = \mathbf{x}'$, i.e., at any time t, $r_i(\mathbf{x}(t)) = \sum_{v \in q_i(t)} b_v(t)$. Let $\mathbf{r}(\mathbf{x}) = (r_i(\mathbf{x}) : i \in F)$. We call the vector function $\mathbf{r}(.)$ *the resource allocation.*

Under Assumption 1 we now show that the set of feasible service-rate allocations across classes, i.e., the *capacity region*, is a polymatroid. We say a polytope $\tilde{\mathcal{C}}$ is a *polymatroid* if there exists a set function $\tilde{\mu}$ on F such that

$$\tilde{\mathcal{C}} = \left\{ \mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq \tilde{\mu}(A), \ \forall A \subset F \right\},$$

and if $\tilde{\mu}$ satisfies the following properties:

1) Normalized: $\tilde{\mu}(\emptyset) = 0$.

2) Monotonic: if $A \subset B$, $\tilde{\mu}(A) \leq \tilde{\mu}(B)$.

3) Submodular: for all $A, B \subset F$,

$$\tilde{\mu}(A) + \tilde{\mu}(B) \geq \tilde{\mu}(A \cup B) + \tilde{\mu}(A \cap B).$$

A function $\tilde{\mu}$ satisfying the above properties is called a *rank function.* Polymatroids and submodular functions are well studied in the literature, see e.g., [46]. Each polymatroid $\tilde{\mathcal{C}}$ has a special property that for any $\mathbf{r} \in \tilde{\mathcal{C}}$, there exists $\mathbf{r}' \geq \mathbf{r}$ such that $\mathbf{r}' \in \tilde{\mathcal{D}} \triangleq \{\mathbf{r} \in \tilde{\mathcal{C}} : \sum_{i \in F} r_i = \tilde{\mu}(F)\}$ [21]. Also, as evident from the definition, for any $A \subset F$ the set $\{\mathbf{r} \in \tilde{\mathcal{C}} : r_i = 0, \forall i \notin A\}$ is a polymatroid, with a rank function which is the restriction of $\tilde{\mu}$ to subsets of A. A proof of the following theorem is provided in the Appendix.

Theorem 1. *Consider a content delivery system defined by graph $G = (F \cup S, E)$ where each server $s \in S$ has a peak service capacity of μ_s. Let*

$$\mathcal{C} \triangleq \{\mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq \mu(A),\ \forall A \subset F\}.$$

Then, the following hold
1) μ is a rank function.
2) Under Assumption 1, $\mathcal{C}$ is the polymatroid capacity region associated with the system.

We say that a polymatroid capacity region is *symmetric* if $\mu(A) = h(|A|)$ for any $A \subset F$ where $h : \mathbb{Z}_+ \to \mathbb{R}_+$ is a non-decreasing function, i.e., $\mu(A)$ depends on A only through $|A|$. Conversely, it is easy to show that if $\mu(A) = h(|A|)$ for some non-decreasing concave function $h : \mathbb{R}_+ \to \mathbb{R}_+$ with $h(0) = 0$, then the capacity region is a symmetric polymatroid.

We say a resource allocation $\mathbf{r}(.)$ is feasible if $\mathbf{r}(\mathbf{x}) \in \mathcal{C}$ for each $\mathbf{x}$. Different feasible resource allocations may potentially lead to different user performance as we will see in the sequel. In next section we focus on a particular resource allocation to leverage its analytical tractability.

Further, we let

$$\hat{\mathcal{C}} \triangleq \left\{ \boldsymbol{\rho}' \geq \mathbf{0} : \sum_{i \in A} \rho'_i < \mu(A),\ \forall A \subset F \right\}, \tag{2.1}$$

and will see, $\hat{\mathcal{C}}$ is a set of loads which are stabilizable for appropriate rate allocation policies.

<u>*Notation for scaling:*</u> Consider sequences of numbers $(f_n : n \in \mathbb{N})$ and $(g_n : n \in \mathbb{N})$. We say that $f_n = O(g_n)$ if there exists a constant $k > 0$ and an integer n_0 such that for each $n \geq n_0$, we have $f_n \leq k g_n$. We say that $f_n = \Omega(g_n)$ if there exists a constant $k > 0$ and an integer n_0 such that for each $n \geq n_0$, we have $f_n \geq k g_n$.

We say that $f_n = o(g_n)$ if $\lim_{n\to\infty} \frac{f_n}{g_n} = 0$. Similarly, we say that $f_n = \omega(g_n)$ if $\lim_{n\to\infty} \frac{g_n}{f_n} = 0$.

2.3 Mean delay analysis

In this section we provide an exact expression for mean delays of jobs in each class under balanced fair resource allocation policy. The balanced fair (BF) allocations were introduced in [7] to provide ‘insensitivity’ in bandwidth sharing networks. By insensitivity we mean that performance depends on service requirement distribution of each class only through its mean. BF is also known to be structurally close to proportional fair resource allocation policy [6,7,39]; in fact, we will compare BF with proportional fair and certain other resource allocation policies in Chapter 3 and develop a performance bound for them by using BF performance analysis developed below.

Balanced fair rate allocation [7] for a polymatroid capacity region $\mathcal{C}$ can be defined as the service rate allocation $\mathbf{r}(\mathbf{x})$, where for any $\mathbf{x}$,

$$r_i(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e}_i)}{\Phi(\mathbf{x})},\ \forall i \in F \tag{2.2}$$

where function Φ is called a balance function and is defined recursively as follows: $\Phi(\mathbf{0}) = 1$, and $\Phi(\mathbf{x}) = 0\ \forall \mathbf{x}$ s.t. $x_i < 0$ for some i, otherwise,

$$\Phi(\mathbf{x}) = \max_{A \subset F} \left\{ \frac{\sum_{i \in A} \Phi(\mathbf{x} - \mathbf{e}_i)}{\mu(A)} \right\}, \tag{2.3}$$

where $\mathbf{e}_i$ is a vector with 1 at i^{th} position and 0 elsewhere. As shown in [7], (5.1) ensures the important property of insensitivity, while (5.2) ensures that $\mathbf{r}(\mathbf{x})$ for each $\mathbf{x}$ lies in the capacity region, i.e., the constraints $\sum_{i \in A} r_i(\mathbf{x}) \leq \mu(A)$ are satisfied for each A. It also ensures that there exists a set $B \subset A_{\mathbf{x}}$ for which $\sum_{i \in B} r_i(\mathbf{x}) = \mu(B)$. In fact the BF allocation is the unique policy satisfying the above properties.

It was shown in [6, 7] that as long as the load vector $\boldsymbol{\rho}$ lies $\hat{\mathcal{C}}$, the random process $(\mathbf{X}(t) : t \in \mathbb{R})$ is stationary. Further, under this condition, its stationary distribution is given by

$$\pi(\mathbf{x}) = \frac{\Phi(\mathbf{x})}{G(\boldsymbol{\rho})} \prod_{i \in F} \rho_i^{x_i} \quad \text{where} \quad G(\boldsymbol{\rho}) = \sum_{\mathbf{x}'} \Phi(\mathbf{x}') \prod_{i \in F} \rho_i^{x'_i}.$$

A resource allocation is Pareto efficient if for any state $\mathbf{x}$, there does not exist an $\mathbf{r}' \in \mathcal{C}$ such that $r'_i \geq r_i(\mathbf{x}),\ \forall i \in A_{\mathbf{x}}$ with a strict inequality for at least one $i \in A_{\mathbf{x}}$. Pareto efficiency is a desirable property since it implies that the resource allocation is less wasteful. BF may not satisfy this property in general, e.g., see triangle networks studied in [7]. However, Theorem 2 below shows that BF is Pareto efficient when the capacity region is a polymatroid. For a polymatroid capacity $\mathcal{C}$, showing Pareto efficiency is equivalent to showing $\sum_{i \in A_{\mathbf{x}}} r_i(\mathbf{x}) = \mu(A_{\mathbf{x}})$. A proof of the following theorem is provided in the Appendix.

Theorem 2. *For balanced fair rate allocations on polymatroid capacity regions we have* $\sum_{i \in A_{\mathbf{x}}} r_i(\mathbf{x}) = \mu(A_{\mathbf{x}})$ *for all* $\mathbf{x}$.

A similar result was proved in [11] for the special case of wireline networks with tree topology. Theorem 2 below serves as a basis to obtain a recursive expression for the mean delays. In the expression below, $G_A(\boldsymbol{\rho})/G(\boldsymbol{\rho})$ is the stationary probability that the set of active classes is A.

Theorem 3. *Consider a system with polymatroid capacity region* $\mathcal{C}$, *with load* $\boldsymbol{\rho}$ *and under balanced fair resource allocation. Let* $\mu(.)$ *be the rank function function associated with the capacity region. The mean delay for requests/flows of class* i *is given by*

$$E\left[D_i\right] = \frac{\nu_i \frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho})}{G(\boldsymbol{\rho})} = \nu_i \frac{\partial}{\partial \rho_i} \log G(\boldsymbol{\rho}), \tag{2.4}$$

where $G(\boldsymbol{\rho})$ *is given by,*

$$G(\boldsymbol{\rho}) = \sum_{A \subset F} G_A(\boldsymbol{\rho}), \tag{2.5}$$

and where $G_\emptyset(\boldsymbol{\rho}) = 1$ *and* $G_A(\boldsymbol{\rho})$ *can be computed recursively as*

$$G_A(\boldsymbol{\rho}) = \frac{\sum_{i\in A} \rho_i G_{A\setminus\{i\}}(\boldsymbol{\rho})}{\mu(A) - \sum_{j\in A} \rho_j}. \tag{2.6}$$

Also, $\frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho})$ *can be recursively computed, without actually computing derivatives, as follows:*

$$\frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho}) = \sum_{A\subset F} \frac{\partial}{\partial \rho_i} G_A(\boldsymbol{\rho}), \tag{2.7}$$

where $\frac{\partial}{\partial \rho_i} G_\emptyset(\boldsymbol{\rho}) = 0$*, and,*

$$\frac{\partial}{\partial \rho_i} G_A(\boldsymbol{\rho}) = \frac{G_A(\boldsymbol{\rho}) + G_{A\setminus\{i\}}(\boldsymbol{\rho}) + \sum_{j\in A} \rho_j \frac{\partial}{\partial \rho_i} G_{A\setminus\{j\}}(\boldsymbol{\rho})}{\mu(A) - \sum_{j\in A} \rho_j}, \tag{2.8}$$

if $i \in A$ *and* 0 *otherwise.*

Proof. By Little's law, we have

$$E\left[D_i\right] = \frac{\sum_{\mathbf{x}} x_i \pi(\mathbf{x})}{\lambda_i} = \frac{\nu_i \frac{\partial}{\partial \rho_i} G(\boldsymbol{\rho})}{G(\boldsymbol{\rho})}. \tag{2.9}$$

Thus, to prove the result we only need to show (2.5). Equation (2.7) follows by taking derivative of (2.5) w.r.t. ρ_i. From Theorem 2 and (5.2) we have,

$$\Phi(\mathbf{x}) = \frac{\sum_{i\in A_{\mathbf{x}}} \Phi(\mathbf{x} - \mathbf{e}_i)}{\mu(A_{\mathbf{x}})}. \tag{2.10}$$

Since $G_A(\boldsymbol{\rho}) = \sum_{\mathbf{x}:A_{\mathbf{x}}=A} \Phi(\mathbf{x}) \prod_{i\in F} \rho_i^{x_i}$, we get , $G(\boldsymbol{\rho}) = \sum_{A\subset F} G_A(\boldsymbol{\rho})$ and

$$G_A(\boldsymbol{\rho}) = \sum_{\mathbf{x}:A_{\mathbf{x}}=A} \frac{\sum_{i\in A} \Phi(\mathbf{x} - \mathbf{e}_i)}{\mu(A)} \prod_{j\in F} \rho_j^{x_j},$$

$$= \frac{\sum_{i\in A}\sum_{\mathbf{x}:A_{\mathbf{x}}=A}\Phi(\mathbf{x}-\mathbf{e}_i)\prod_{j\in F}\rho_j^{x_j}}{\mu(A)},$$

Rearranging terms, we get,

$$\begin{aligned}\mu(A)G_A(\boldsymbol{\rho}) &= \sum_{i\in A}\rho_i \sum_{\mathbf{x}:A_{\mathbf{x}}=A\setminus\{i\}}\Phi(\mathbf{x})\prod_{j\in F}\rho_j^{x_j} + \sum_{i\in A}\rho_i\sum_{\mathbf{x}:A_{\mathbf{x}}=A}\Phi(\mathbf{x})\prod_{j\in F}\rho_j^{x_j},\\ &= \sum_{i\in A}\rho_i G_{A\setminus\{i\}}(\boldsymbol{\rho}) + G_A(\boldsymbol{\rho})\sum_{i\in A}\rho_i,\end{aligned}$$

further simplification of which gives the desired result. □

While the mean delay for systems with polymatroid capacity can be computed using (2.4) - (2.8), an exact computation has a complexity which grows exponentially in the number of files n. If, however, the capacity region is given by a symmetric polymatroid and the load vector $\boldsymbol{\rho}$ is homogenous, the complexity is linear in n. The following corollary details this result.

Corollary 1. *Consider a system with symmetric polymatroid capacity region $\mathcal{C}$ with homogenous load $\boldsymbol{\rho}$ and under balanced fair resource allocation, i.e., for each $A \subset F$, the rank function $\mu(A) = h(|A|)$ for some non-decreasing function $h : \mathbb{Z}_+ \to \mathbb{R}_+$ and for all $j \in F$ $\rho_j = \rho = \lambda\nu$. Then, the mean delay to serve the requests/flows of class i is given by,*

$$E\left[D_i\right] = \frac{\nu\hat{F}(\rho)}{F(\rho)}, \tag{2.11}$$

where, $F(\rho)$ and $\hat{F}(\rho)$ can be recursively obtained as follows:

$$F(\rho) = \sum_{k=0}^{n} F_k(\rho), \tag{2.12}$$

where, $F_0(\rho) = 1$, and for $k \geq 1$,

$$F_k(\rho) = \frac{(n-k+1)\rho F_{k-1}(\rho)}{h(k) - k\rho}. \tag{2.13}$$

Also,

$$\hat{F}(\rho) = \sum_{k=0}^{n} \frac{k}{n} \hat{F}_k(\rho), \tag{2.14}$$

where, $\hat{F}_0(\rho) = 0$, and for $k \geq 1$,

$$\hat{F}_k(\rho) = \frac{F_k(\rho) + \frac{n-k+1}{k} F_{k-1}(\rho) + \frac{(n-k+1)(k-1)}{k} \rho \hat{F}_{k-1}(\rho)}{h(k) - k\rho}. \tag{2.15}$$

Proof. From symmetry it follows that $G_A(\boldsymbol{\rho})$ depends on A only through $|A|$. For each $k \geq 0$, let $H_k(\rho) = G_A(\boldsymbol{\rho})$ for A such that $|A| = k$. Similarly, let $\hat{H}_k(\rho) = \frac{\partial}{\partial \rho_i} G_A(\boldsymbol{\rho})$ for A such that $|A| = k$ and $i \in A$.

Thus, from (2.5), we get

$$H_k(\rho) = \frac{k\rho H_{k-1}(\rho)}{h(k) - k\rho}.$$

Similarly, from (2.8), we get

$$\hat{H}_k(\rho) = \frac{H_k(\rho) + H_{k-1}(\rho) + (k-1)\rho \hat{H}_{k-1}(\rho)}{h(k) - k\rho}.$$

Then, the result follows from Theorem 4 by letting $F_k(\rho) = \binom{n}{k} H_k(\rho)$ and $\hat{F}_k(\rho) = \binom{n}{k} \hat{H}_k(\rho)$ and appropriate simplifications. $\square$

2.4 Performance asymptotics

In this section we consider asymptotics for large file-server systems wherein the number of files n and the number of servers m become large. Our focus is on systems where there is increased overall demand for increasingly diverse content, and thus one must scale server resources. The number of files in a content delivery infrastructure can be huge, e.g., a study in [64] estimated that Youtube had 5×10^8 videos in 2011, and the number has been steadily increasing since then.

Consider a system with a given m and n. Let each file be replicated across c different servers chosen at random. Let graph $G^{(m,n)} = (F^{(n)} \cup S^{(m)}, E^{(m,n)})$ represent a realization of such random file-server system. Further, let the $\mu_s^{(m,n)} = \xi$ for each server $s \in S^{(m)}$. Let the resulting capacity region realization be $\mathcal{C}^{(m,n)}$. Also, let the total request rate in the system be $m\lambda$, i.e., it grows linearly with m, resulting in a total traffic load $m\rho = m\lambda\nu$ where ν is the mean service requirement per request. For simplicity, let the traffic load across files be symmetric, and thus equal to $\rho_i^{(m,n)} = m\rho/n$ for each file $i \in F^{(n)}$.

Further, we assume the number of files n to be orders of magnitude larger than m. To model this, we first fix m, and consider a sequence of systems wherein the number of files n increases to infinity. Then, to model the fact that m itself can be large, we consider a sequence of such sequences where m itself increases to infinity. This is a good model towards approximating systems with say $m \sim 10^3$, but with $n \sim 10^7$ or greater. For a given m and

n, we let the total load on the system be ρm, with a fixed load per server ρ. Thus, for a given m, the load per file is equal to $\frac{\rho m}{n}$. As we will see in the sequel, this asymptotic regime is similar in spirit to that considered in the study of the super-market model [12, 43, 56].

For each realization, the service capacity is allocated dynamically according to balanced fair allocations over the associated capacity region, see Sec. 2.3. We shall refer to the file-server systems with resource allocation as described above as one with *Random Placement with Balanced Fairness (RP-BF)*.

2.4.1 Performance asymptotics for symmetric 'averaged' capacity region

For a given realization of the random file placement, the associated rank function $\mu^{(m,n)}$ need not be symmetric. Exact performance computations for such a system would require computation of the associated capacity region and evaluating the recursions developed in Sec. 2.3 both of which have exponential complexity in n. However, a key insight we develop below is that realizations of large RP-BF systems exhibit the same performance.

To that end consider the averaged RP-BF system having the "averaged capacity region". Let $M^{(m,n)}(.)$ denote the random rank function associated with an (m,n) RP-BF file placement. Given a set of files A where $|A| = k \leq n$ one can show that

$$\bar{\mu}^{(m,n)}(A) \triangleq E[M^{(m,n)}(A)] = \xi m(1-(1-c/m)^k).$$

Indeed the probability that none of the c copies of a file are stored on a given server is $(1 - c/m)$. Thus the probability that none of A' s k files is stored at the server is $(1 - c/m)^k$. So $m(1 - (1 - c/m)^k)$ is the mean number of servers that can serve *at least* one file in A, and the above is their associated service capacity. The averaged capacity region is thus given by a *symmetric* polymatroid with rank function $\bar{\mu}^{(m,n)}(A) = h^{(m,n)}(|A|)$ where

$$h^{(m,n)}(k) \triangleq \xi m(1 - (1 - c/m)^k) \text{ for } k = 0, 1, \ldots, n. \tag{2.16}$$

Below we let $\pi^{(m,n)}(\mathbf{x})$ denote the stationary distribution of the queue length process for the averaged RP-BF system, i.e., using balanced fair allocations over the average capacity region. Also, let $E[D^{(m,n)}]$ be the expected delay for a typical request in this system. The following result gives a simple expression for the expected delay in the asymptotic regime of interest. Its proof is provided in the Appendix.

Theorem 4. *Consider a sequence of (m, n) averaged RP-BF file-server systems with symmetric polymatroid capacity with the rank function $\bar{\mu}^{(m,n)}(\cdot)$ given above and symmetric traffic load $\rho_i^{(m,n)} = m\rho/n$ for each file i where $\rho = \lambda\nu < \xi$. For given (m, n), let $\pi_k^{(m,n)} = \sum_{\mathbf{x}:|A_{\mathbf{x}}|=k} \pi^{(m,n)}(\mathbf{x})$ for $k = 0, 1, 2, \ldots, n$, and let*

$$\alpha^* \triangleq \frac{1}{c} \log\left(\frac{1}{1 - \rho/\xi}\right). \tag{2.17}$$

Then, for each $\epsilon > 0$, we have:

$$\lim_{m\to\infty} \lim_{n\to\infty} \sum_{k=\lfloor\alpha^* m(1-\epsilon)\rfloor}^{\lfloor\alpha^* m(1+\epsilon)\rfloor} \pi_k^{(m,n)} = 1 \tag{2.18}$$

Also, under the same limits, the expected delay is given by

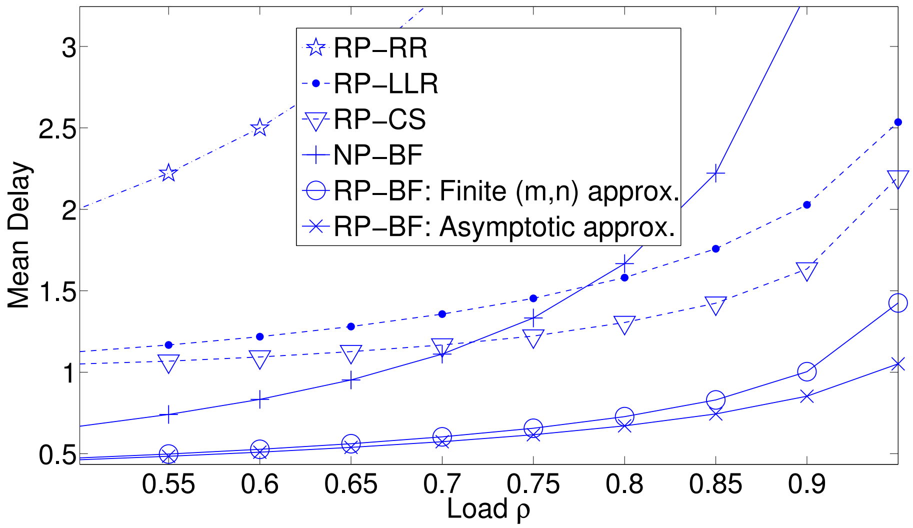
$$\lim_{m\to\infty}\lim_{n\to\infty} E[D^{(m,n)}] = \frac{\alpha^*}{\lambda} = \frac{1}{\lambda c}\log\left(\frac{1}{1-\rho/\xi}\right). \tag{2.19}$$

The intuition underlying this result is as follows. For large systems, the probability measure $\pi^{(m,n)}(\mathbf{x})$ concentrates on states $\mathbf{x}$ such that $h^{(m,n)}(|A_{\mathbf{x}}|) \approx \rho m$. From (4.1), for any $\alpha > 0$, we have $\lim_{m\to\infty}\lim_{n\to\infty}\frac{1}{m}h^{(m,n)}(\alpha m) = \xi(1-e^{-c\alpha})$, which is equal to ρ for $\alpha = \alpha^*$.

Fig. 2.2 exhibits plots for mean delay as a function of load for *averaged* RP-BF systems. The plot for the approximation for a finite (m, n) system was computed using Corollary 1. The closeness of asymptotic expression to that for finite (m, n) depends on the value of ρ. Suppose n is orders of magnitude larger than m. For ρ less than or equal to 0.8 the asymptotic expression is remarkably close even for m as small as 30. Although not shown in the figure, for $\rho = 0.9$ the expression is close for m equal to 60 or larger. In next section we discuss why these expressions are good approximations for the actual performance in RP-BF realizations.

2.4.2 Approximating the performance of RP-BF file-server system via 'averaged' RP-BF.

In this subsection, we argue that the expression for mean delay given in Theorem 4 based on the averaged RP-BF system can be used to approximate the performance of realization of a large RP-BF file server system. In fact, we conjecture that the mean delay expression given in Theorem 4 holds for *almost all* sequences of RP-BF file placement realizations.



(a) Mean delay comparison: $\nu_i = 1$ and $\rho_i = \rho m/n$ for each $i \in F$, $\xi = 1$, and $c = 3$. For finite (m, n) approximations: $m = 30$, $n = 2 \times 10^4$.

File placement options	Service policies
RP: Randomized Placement	RR: Randomized Routing
NP: Non-overlapping Pools	LLR: Least-loaded Routing
	CS: Centralized Scheduling
	BF: Balanced Fairness

(b) Abbreviations

	Pooling of servers	Better load-balancing
RP-RR	×	×
RP-LLR	×	✓
RP-CS	×	✓
NP-BF	✓	×
RP-BF	✓	✓

(c) Qualitative comparision

Figure 2.2: Comparision of different resource allocation policies.

Recall that $M^{(m,n)}(.)$ denotes random rank function for our (m,n) RP-BF system, and $\bar{\mu}^{(m,n)}(.)$ its mean over all random file placements, and $\mu^{(m,n)}(.)$ denotes a (likely asymmetric) realization of $M^{(m,n)}(.)$. Our informal argument involves two steps.

Step 1: For large set of files A such that $|A| \approx \alpha m$ (integer) we have that

$$\frac{1}{m}\mu^{(m,n)}(A) \approx \frac{1}{m}h_{\text{avg}}^{(m,n)}(|A|),$$

where

$$h_{\text{avg}}^{(m,n)}(|A|) \triangleq \frac{\sum_{B:|B|=\alpha m} \mu^{(m,n)}(B)}{\binom{n}{\alpha m}}$$

This results from a general concentration property for c-Lipschitz monotonic submodular functions [54].

Step 2: With high probability, for most sets A such that $|A| = \alpha m$, we have

$$\frac{1}{m}\mu^{(m,n)}(A) \approx \frac{1}{m}\bar{\mu}^{(m,n)}(A) = \frac{1}{m}h^{(m,n)}(\alpha m),$$

where $h^{(m,n)}(.)$ is given by (4.1). This can be shown as follows.

Recall that $M^{(m,n)}(A) = \xi \sum_{s \in S^{(m)}} \mathbf{1}_{\left\{s \in S^{(m,n)}(A)\right\}}$, where $S^{(m)}$ and $S^{(m,n)}(A)$ are respectively the set of m servers, and the (random) set of servers where a copy of at least one of the files in A is stored. Suppose, for each (m,n), a subset of files $A_{\alpha}^{(m,n)}$ is selected uniformly at random from all $A \subset F^{(n)}$ such that $|A| = \alpha m$. Suppose $S^{(m)} = \{s_1, s_2, \ldots, s_m\}$. Consider a random process

$$X^{(m,n)} = \left(X_1^{(m,n)}, X_2^{(m,n)}, \ldots, X_m^{(m,n)}\right)$$

where

$$X_i^{(m,n)} = \mathbf{1}_{\left\{s_i \in S^{(m,n)}\left(A_\alpha^{(m,n)}\right)\right\}}, \ \forall i \leq m.$$

Then,

$$M^{(m,n)}\left(A_\alpha^{(m,n)}\right) = \xi \sum_{i=1}^{m} X_i^{(m,n)}.$$

We now study $\lim_{m\to\infty} \lim_{n\to\infty} \frac{1}{m} M^{(m,n)}\left(A_\alpha^{(m,n)}\right)$.

It can be checked that for each n, $X^{(m,n)}$ is a process of m exchangeable Bernoulli$(1-(1-c/m)^{\alpha m})$ random variables, and so is $X^{(m,\infty)} \triangleq \lim_{n\to\infty} X^{(m,n)}$. Also, for any fixed set of l servers, say $\{s_1, s_2, \ldots, s_l\}$, $X_i^{(m,\infty)}$ for $i \in \{1, 2, \ldots, l\}$ can be shown to become independent in the limit as $m \to \infty$. As was shown in [1, 52], such asymptotic independence implies that a law of large numbers would hold for a sequence of exchangeable random processes which for our case implies that $\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} X_i^{(m,\infty)} = 1 - e^{-\alpha c}$ in probability. This shows that for most realizations, $\frac{1}{m}\mu^{(m,n)}\left(A_\alpha^{(m,n)}\right) \approx \frac{1}{m} h^{(m,n)}(\alpha m)$ for almost all sets A of size αm, thus showing the claim in Step 2.

Step 1 and Step 2 jointly imply that for each A such that $|A| \approx \alpha m$,

$$\frac{1}{m}\mu^{(m,n)}(A) \approx \frac{1}{m} h_{\text{avg}}^{(m,n)}(\alpha m) \approx \frac{1}{m} h^{(m,n)}(\alpha m),$$

which further suggests that Theorem 4 holds for almost all file placement realizations of RP-BF systems.

Note that, for a given realization, there might still be few sets A of large enough size such that $\mu^{(m,n)}(A)$ is not close to $h^{(m,n)}(|A|)$. For example, consider set A of size m/c where each file in A is stored in disjoint set of

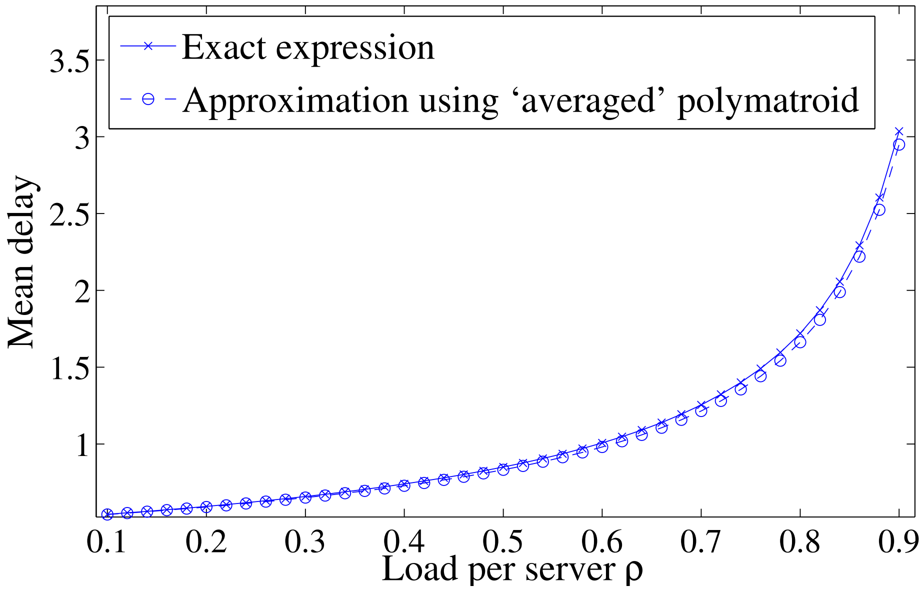


Figure 2.3: Approximating performance of a file server system by using the 'averaged' polymatroid capacity: $m = 4$, $n = 6$, service rate $\mu_s = 1$ for each server s, $\rho_i = m\rho/n$ and $\nu_i = 1$ for each class i.

servers. Here, $\mu(A) = m$ and is not close to $h^{(m,n)}(m/c)$. The above argument only shows that such outliers are small in number. A more rigorous argument is needed to show that the small number of outliers do not impact the overall performance a lot. We defer such analysis to a possible future work.

Let us numerically check the goodness of the approximation using an 'averaged' polymatroid capacity for a file-server system with $m = 4$ servers and $n = 6$ files, with each file stored on a distinct set of $c = 2$ servers. The mean delay in such a system can be shown to be equivalent to a system with $m = 4$ severs and number of files $n \to \infty$, as follows. A system with $m = 4$ servers has $\binom{m}{c} = 6$ distinct server-pools. For a given set of servers, one may view the group of files stored on each of them a distinct file-class. Since the files are distributed randomly, the load across these file-classes (equivalently server-pools) becomes homogeneous asymptotically.

Note, however, the rank function $\mu^{(4,6)}(.)$ is asymmetric. For example, $\mu^{(4,6)}(A)$ takes values 3 or 4 for different sets A of size 2, which is a difference of about 30%. We numerically compute $\mu^{(4,6)}(A)$ for each of the 2^6 subsets A of F, as well as an 'averaged' capacity region with the associated 'averaged' rank function $\mu^{(4,6)}_{\text{avg}}(A) = h^{(4,6)}_{\text{avg}}(|A|)$ for each $A \subset F$, where $h^{(4,6)}_{\text{avg}}(k) = \frac{\sum_{A:|A|=k} \mu^{(4,6)}(A)}{\binom{n}{k}}$ for $k = 0, 1, \ldots, 6$. Fig. 2.3 exhibits the exact performance for both capacity regions using Theorem 4 and Corollary 1. It can be seen that the exact and the averaged systems are remarkably close.

2.5 Comparison with routing and scheduling policies

We now compare RP-BF with several other resource allocation policies. For a given set of files and servers, the key components of a resource allocation policy that impact user-performance are the following:

1) **File placement**: Options include: (a) partitioning the set of servers and constraining each partition to store a distinct set of files, thus creating independent 'non-overlapping' pools of servers; (here, by pools of servers we mean the subsets of servers which can jointly serve file requests due to common files they store); and (b) randomly storing files across the servers, resulting into overlapping pools of servers. Option (a) was proposed in [14] as having a desirable property of higher reliability against correlated failures. We will

explore this further in Section 2.6.1 as well. Option (b), as we will see below, opens opportunities to better balance the load across servers and improve performance.

2) **Service policy**: A naive service policy is to route a file request randomly upon arrival to one of the servers that stores the corresponding file. The requests thus get queued at the servers and are served in, e.g., round-robin or processor sharing fashion. A simple modification to this policy which makes routing a function of the current load at servers, e.g., the number of queued requests at the servers, can provide significant performance improvement [12,56]. An even better approach is that considered in [37, 53] where the requests are queued centrally and their service is scheduled dynamically based upon the availability of the servers. In each of these policies, a request is constrained to be served by a single server. Our work departs from these approaches, in that we allow each request to be served jointly by a pool of servers. As explained in Section 2.2, we constrain service only through Assumption 1, or equivalently through capacity region $\mathcal{C}^{(m,n)}$. Under these constraints, we balance the load across servers through a fairness based rate allocation as explained in Section 2.3.

We now compare four different resource allocation policies with RP-BF, each of which is characterized by a choice of file placement and of service policy.

Randomized Placement with Random Routing (RP-RR): Files are stored uniformly at random in c servers as with RP-BF. Upon arrival of a file request, it is randomly routed to one of the c servers that stores the corresponding file. Each server serves its request in processor sharing fashion. As $n \to \infty$, the total load of ρm is eventually balanced across the m servers and the system is equivalent to m independent $M/GI/1$ systems with load ρ and service rate ξ.

Random Placement with Least-loaded Routing (RP-LLR): Files are stored uniformly at random. Upon arrival, requests are routed to a server with least number of ongoing jobs among c servers which store the corresponding file. Each server serves its request in a processor sharing fashion. In the limit as $n \to \infty$, this system is equivalent to the super-market model studied in [12, 56]. Let p_k be the fraction of servers having k waiting requests in equilibrium. When the service-requirement distribution for each request is exponential, it was shown in [56] that as the number of servers $m \to \infty$, the fraction p_k is given by

$$p_k = (\rho/\xi)^{\frac{c^k-1}{c-1}} - (\rho/\xi)^{\frac{c^{k+1}-1}{c-1}},$$

where ρ is the load per server. Thus, by Little's law, the mean delay for a typical request in the asymptotic regime of interest is given by,

$$E[D_{\text{RP-LLR}}] = \frac{1}{\lambda} \sum_{k=1}^{\infty} k p_k = \frac{1}{\lambda} \sum_{k=1}^{\infty} (\rho/\xi)^{\frac{c^k-1}{c-1}}. \tag{2.20}$$

Random Placement with Centralized Scheduling (RP-CS): Files are stored uniformly at random. Unlike the previous policies each server serves a maximum of one request at a time, and there is no service preemption.

Upon arrival of a request, if there exist idle servers which store a copy of the corresponding file, it is assigned and served by one of them at random, else, it is queued at a central queue. Upon completion of service of a request at a server, if there exists a waiting request which the server can serve, it gets assigned to that server. If there exist multiple such requests, the choice is made as follows. Among all the files which the available server stores, one of the files with maximum number of waiting requests is chosen at random. Among the waiting requests of the chosen file, a request is chosen at random for service.

Non-overlapping Pools with Balanced Fairness (NP-BF): The m servers are divided into m/c groups, each of size c. Each server group stores a mutually exclusive subset with nc/m files. Within a group, each server stores the same set of files. Each file is thus stored at c servers. Under balanced fairness, each group behaves as an independent pool of servers which serves its requests in processor sharing fashion. The system is equivalent to m/c independent $M/GI/1$ queues with load ρc and service rate ξc, with mean delay given by

$$E[D_{\text{NP-BF}}] = \frac{\nu}{c\xi(1-\rho/\xi)}. \tag{2.21}$$

Contrast this with Theorem 4 where the mean delay increase is logarithmic in $1/(1-\rho/\xi)$.

In Fig. 2.2, we compare the performance of these resource allocation policies. RP-BF's performance is plotted using the approximations described in Section 2.4. The performance of RP-RR, RP-LLR and NP-BF is plotted using corresponding asymptotic expressions for mean delay described above. For RP-CS, the service requirement distribution was assumed exponential and we built a simulator for the underlying Markov Chain. For each point in the plot, the average number of requests waiting in the queue or in service was measured over a period of time of up to 10^6 events and the mean delay was computed using Little's law.

All the above policies are stable for any value of ρ less than 1. As expected, RP-RR performs poorly as it does not exploit pooling or load dependent routing. RP-CS outperforms RP-LLR at higher loads since requests are queued centrally in the former and its service policy uses global state information. NP-BF outperforms both RP-CS and RP-LLR at lower loads since pooling of servers works to its advantage. However, due to creation of independent non-overlapping pools, its ability to balance the load across servers is limited and it performs significantly worse at higher loads.

RP-BF outperforms all of the policies since it enjoys the best of both worlds. At higher loads, one might expect that the gains of RP-BF over RP-LLR and RP-CS due to pooling may be limited since load balancing of the later policies would ensure that most of the servers are busy serving requests most of the time and are utilized well. However, even for $\rho = 0.9$, the mean delay for RP-LLR and RP-CS is over 2 and 1.6 times that of RP-BF for $c = 3$, respectively.

For larger values of c, the improvements are even greater. For any value of c, mean delay for RP-LLR and RP-CS is lower bounded by 1. However, from Theorem 4, mean delay for RP-BF is inversely proportional to c. The significant performance improvement by RP-BF shows that server pooling and fairness based resource allocation is worthwhile towards optimizing the performance of centralized content delivery systems.

2.6 Using model to study system tradeoffs

2.6.1 Recovery costs on correlated failure v/s performance

We consider the cost of recovering files when there are large-scale correlated failures such as those occurring after power outages, see [14] for an extensive discussion. It is not uncommon in datacenters that about 1% of servers fail to reboot after a power outage. The system then needs to recover data in these servers by retrieving copies from the servers that successfully rebooted. However, there might be some files for which no copy exists in the datacenter due to the failure of all servers in which it was stored. The probability of such an event occurring can be significant especially when the total number of files in the system is large.

When this occurs the system needs to locate and recover the lost files from 'cold' storage. Recovery of the files from cold storage may incur a high fixed cost but may not be greatly affected by the number of files lost. Thus in practice (as argued in [14]) it is desirable that the probability that one or more files are lost during power outage events be low. This can be achieved by constraining randomness in how files are copied across servers. The intuition from Section 2.5 suggests that randomly 'spreading' the files across the servers so that the server pools overlap improves the user perceived performance. However, this may increase the probability of a file loss. To study how these quantities are related, we consider a storage policy that divides m servers into independent groups of smaller size and restricts the copies of each file to be placed within a single group, as follows.

Fix an integer κ such that $c \leq \kappa \leq m$. Suppose, for now, that number of servers m is divisible by κ and that number of files n is divisible by m/κ. Divide the set S of m servers into m/κ number of groups each of size κ. Similarly, divide the set F of n files into disjoint m/κ groups of size $n\kappa/m$. Associate each group of files with a distinct group of servers. Then, for each file, independently store c copies by selecting c servers uniformly at random from the corresponding group.

Suppose that upon a power outage, each server fails to reboot with probability γ independently. Then, for a group of size κ, the probability that l servers fail is $\binom{\kappa}{l}\gamma^l(1-\gamma)^{\kappa-l}$, so the probability that one or more files are lost can be given by

$$P_{\text{loss}} = 1 - \left(\sum_{l=0}^{c-1} \binom{\kappa}{l} \gamma^l (1-\gamma)^{\kappa-l} + \sum_{l=c}^{\kappa} \binom{\kappa}{l} \gamma^l (1-\gamma)^{\kappa-l} \left(1 - \frac{\binom{l}{c}}{\binom{\kappa}{c}} \right)^{n\kappa/m} \right)^{m/\kappa}$$

For the general case where m is not divisible by κ or n is not divisible by m/κ, we can create non-uniform groups and compute the corresponding loss probability. We use the above expression as a simpler approximation by using $\lfloor m/\kappa \rfloor$ and $\lfloor n\kappa/m \rfloor$ appropriately. Also, the performance within each group

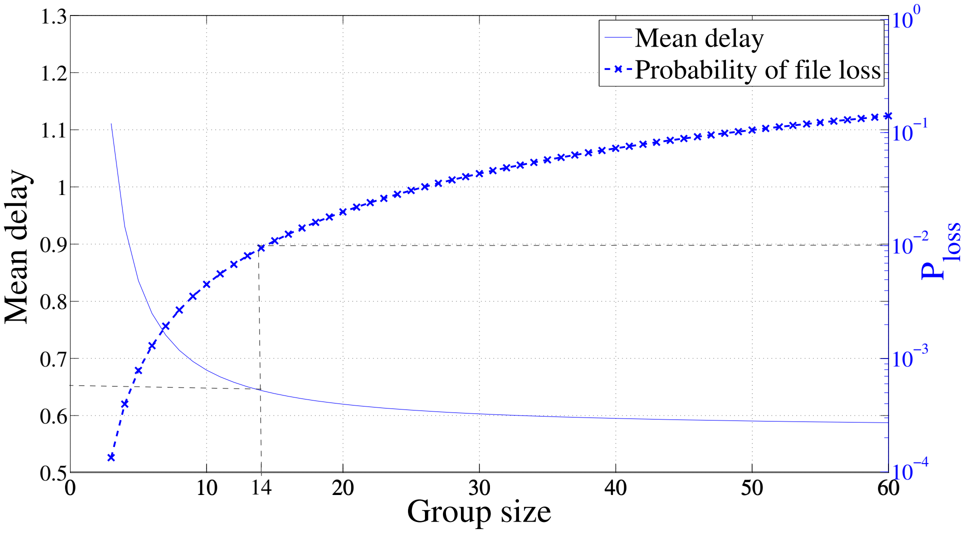


Figure 2.4: Delay v/s reliability $n = 2 \times 10^6$, $m = 400$, $c = 3$, $\gamma = 0.01$, $\rho = 0.7$, and $\nu = 1$.

can be computed using the expression of Corollary 1 for symmetric capacity systems, which gives a reasonable approximation as explained in Sec. 2.4.1.

Fig. 2.4 exhibits the mean delay and P_{loss} for $\gamma = 0.01$ for a system with $n = 2 \times 10^6$, $m = 400$, and $c = 3$ copies. The load per server is $\rho = 0.7$, i.e., the total load on the system is $m\rho = 280$ and is distributed uniformly across files. Also, $\nu_i = 1$ for all $i \in F$ and $\mu_s = 1$ for all $s \in S$. As can be seen, varying κ trades off performance with file loss probability. As κ increases mean delay decreases but quickly saturates at 0.57, which matches with the asymptotic limit as given by Theorem 4. At $\kappa = 14$, mean delay is 0.64 which is about 12% greater than the asymptotic value, while P_{loss} is less than 1%. Decreasing κ can further lower P_{loss} but at the cost of a significant increase in mean delay.

2.6.2 Energy-delay tradeoffs

We now consider RP-BF systems where for each server $s \in S$, we have $\mu_s = \xi$. Energy consumption per unit time by a server is fixed when it is busy and is denoted by e_b. Similarly, even when a server is idle, its energy consumption per unit time is fixed and denoted by e_i. If the system is stable, the sum of the fraction of time each server is busy is equal to $\frac{\sum_{i \in F} \rho_i}{\xi}$. Thus, the mean energy spent by the system per unit time is given by

$$E = e_b \frac{\sum_{i \in F} \rho_i}{\xi} + e_i \left(m - \frac{\sum_{i \in F} \rho_i}{\xi} \right).$$

Thus, one can trade of energy consumption for performance by varying m.

Fig. 2.5 exhibits the energy-delay curve for a system with 2×10^6 files with a fixed total load of 280, $e_b = 1$ units and $e_i = 0.5$ units. Points in the plot are obtained by varying m and computing the performance using Corollary 1. The figure also exhibits tradeoff for the case when the total number of servers are divided into smaller independent groups of size 10, as in Section 2.6.1. The tradeoff curve worsens in this case. For example, to obtain a mean delay of 0.8, it requires $m = 370$ servers while the former system that groups all the servers together requires 320 servers; the corresponding mean energy consumption being 325 units and 300 units, respectively. Thus, creating smaller independent groups of size 10 increases the energy consumption by about 8%.

Next, we consider RP-BF systems where servers' processing speed is

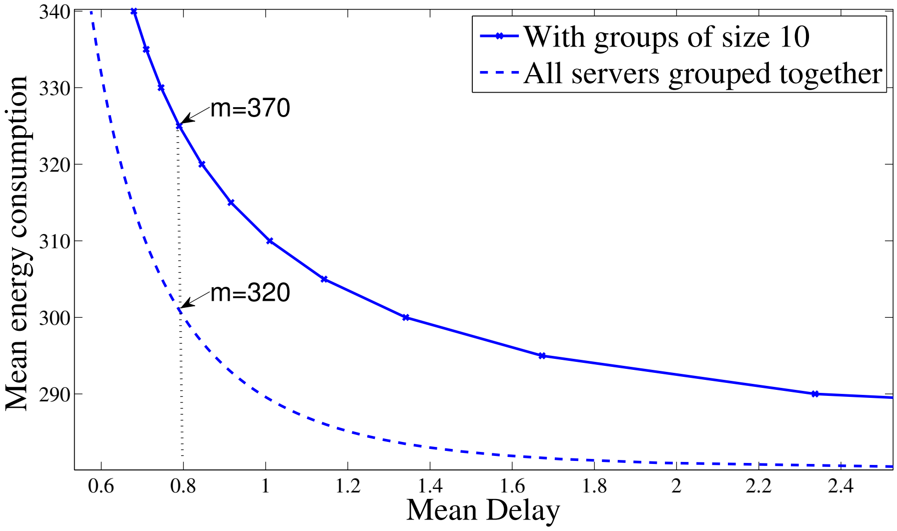


Figure 2.5: Energy-delay tradeoff for system with $n = 2 \times 10^6$ and varying m: $\nu = 1$, $c = 3$, $\xi = 1$, and total load $\rho m = 280$.

a bottleneck. The processing speed can be improved by increasing clock frequency and voltage supply, which in turn increases energy consumption. This dependence is typically modeled through a polynomial relationship of power with ξ, i.e., when the service rate of a server is ξ the power consumption is given by $f(\xi) = \xi^\alpha/\beta$ per unit time where $\alpha > 1$ and β is a positive constant [35]. In practice, even when ξ is set to 0, there is non-negligible leakage power consumption. Since our focus is on dynamic power, we ignore leakage power here. The choice of ξ trades off performance for energy consumption. Here, we consider a simple semi-static policy where each server operates at a fixed rate ξ when busy and rate 0 when idle, thus consuming negligible power when idle. For $M/GI/1$ queues, it was shown in [35] that such a simple policy, with ξ chosen judiciously, is close to an optimal policy for minimiz-

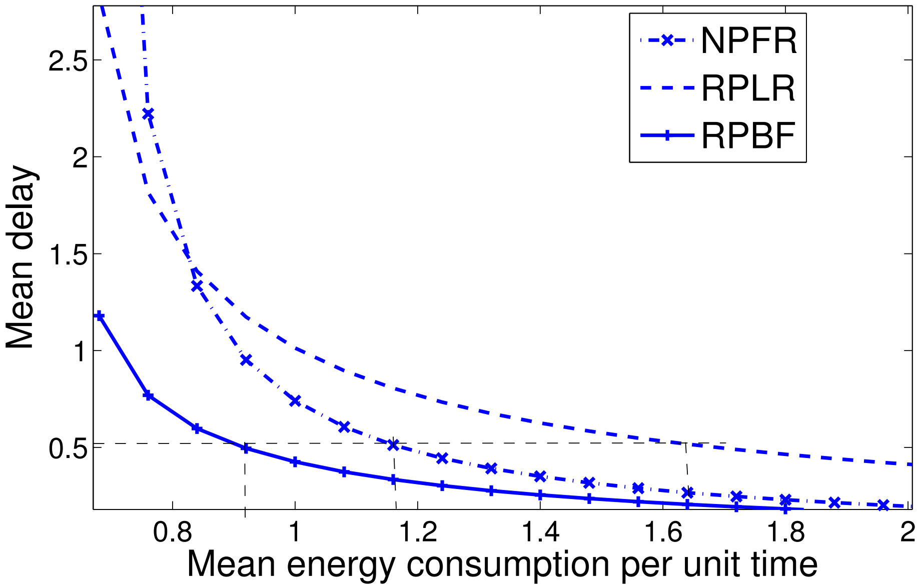


Figure 2.6: Energy-delay tradeoff with varying server speed ξ: load per server fixed at $\rho = 0.8$, $\nu = 1$, and $c = 3$.

ing a weighted average of the mean delay and energy consumption across all dynamic policies where ξ is allowed to vary with the queue state.

Fig. 2.6 compares the energy-performance tradeoff for NP-BF, RP-LLR, and RP-BF where the plots are obtained by varying values of ξ. For RP-BF, Theorem 4 is used to compute dependence of performance on ξ, whereas for NP-BF and RP-LLR, (2.21) and (2.20), respectively, are used. Also, we assume that the power consumption as a function of ξ is given by $f(\xi) = \xi^2$. Since the fraction of time a server is busy in each system is ρ/ξ, the mean energy consumption is given by $E = \rho\xi$. To obtain a mean delay of 0.5 for $\rho = 0.8$, the energy consumption for NP-BF and RP-LLR systems is 20% and

70% more than that for RP-BF, respectively.

2.7 Appendix

2.7.1 Proof of Theorem 1

We first show that μ is a rank function. By definition it is clear that $\mu(\emptyset) = 0$ and that μ is monotonic. To show that $\mu(.)$ is submodular we use the inclusion-exclusion principle to obtain

$$\begin{aligned}\mu(A) &= \sum_{s\in S(A)} \mu_s = \sum_{s\in S(A\cap B)\cup S(A\backslash B)} \mu_s \\ &= \sum_{s\in S(A\cap B)} \mu_s + \sum_{s\in S(A\backslash B)} \mu_s - \sum_{s\in S(A\cap B)\cap S(A\backslash B)} \mu_s.\end{aligned}$$

Similarly,

$$\mu(B) = \sum_{s\in S(B\cap A)} \mu_s + \sum_{s\in S(B\backslash A)} \mu_s - \sum_{s\in S(B\cap A)\cap S(B\backslash A)} \mu_s$$

Again using inclusion-exclusion principle, we further have,

$$\begin{aligned}\mu(A\cup B) &= \sum_{s\in S(A\cup B)} \mu_s = \sum_{s\in S(A\cap B)\cup S(A\backslash B)\cup S(B\backslash A)} \mu_s \\ &= \sum_{s\in S(A\cap B)} \mu_s + \sum_{s\in S(A\backslash B)} \mu_s + \sum_{s\in S(B\backslash A)} \mu_s \\ &- \sum_{s\in S(A\cap B)\cap S(A\backslash B)} \mu_s - \sum_{s\in S(B\cap A)\cap S(B\backslash A)} \mu_s \\ &- \sum_{s\in S(A\backslash B)\cap S(B\backslash A)} \mu_s + \sum_{s\in S(B\cap A)\cap S(A\backslash B)\cap S(B\backslash A)} \mu_s\end{aligned}$$

Also, $\mu(A\cap B) = \sum_{s\in S(A\cap B)} \mu_s$. Thus,

$$\mu(A) + \mu(B) - \mu(A\cup B) - \mu(A\cap B)$$

$$
\begin{aligned}
&= \sum_{s \in S(A\backslash B) \cap S(B \backslash A)} \mu_s - \sum_{s \in S(B \cap A) \cap S(A \backslash B) \cap S(B \backslash A)} \mu_s \\
&\geq 0
\end{aligned}
$$

which shows that μ is submodular.

We now show that $\mathcal{C}$ is the capacity region. We first show that if $\mathbf{r}$ is feasible then $\mathbf{r} \in \mathcal{C}$, and later show the converse.

Suppose $\mathbf{r} \notin \mathcal{C}$. Then, we show that $\mathbf{r}$ violates the capacity constraints in Assumption 1 for any set of active flows $\mathbf{q}$ such that for all i, $|q_i| > 0$ iff $r_i > 0$. By definition of $\mathcal{C}$, there exists $A \subset F$ such that $\sum_{i \in A} r_i > \mu(A)$. Now suppose $\sum_{v \in q_i, s \in S_i} b_{v,s} = r_i$ for all $i \in F$. Then, we get, $\sum_{i \in A} \sum_{v \in q_i, s \in S_i} b_{v,s} > \mu(A)$ which further gives $\sum_{s \in S(A)} \sum_{v \in \cup_{i \in A} q_i} b_{v,s} > \mu(A)$. Thus, there exists s such that $\sum_{v \in \cup_{i \in A} q_i} b_{v,s} > \mu_s$. Thus, $\mathbf{r}$ is not feasible.

We now show the converse, i.e., $\mathbf{r} \in \mathcal{C}$ implies that $\mathbf{r}$ is feasible. Recall that, for a polymatroid capacity $\mathcal{C}$, for all $\mathbf{r} \in \mathcal{C}$ there exists $\mathbf{r}' \geq \mathbf{r}$ such that $\mathbf{r}' \in \mathcal{D}$, where $\mathcal{D} = \{\mathbf{r} \in \mathcal{C} : \sum_{i \in F} r_i = \mu(F)\}$. Thus, it is sufficient to show that if $\mathbf{r} \in \mathcal{D}$, then $\mathbf{r}$ is feasible. Let P be set of all permutations on F. For each $p \in P$, let $\mathbf{r}^{(p)} = (r_i^{(p)} : i \in F)$ such that $r_{p(k)}^{(p)} = \mu(\{p(1), \ldots, p(k)\}) - \mu(\{p(1), \ldots, p(k-1)\})$, for all $k \in \{1, 2, \ldots, n\}$. It can be shown that $\{\mathbf{r}^{(p)} : p \in P\}$ is the set of all extreme points of $\mathcal{D}$, see [21]. Thus, it is sufficient to show that $\mathbf{r}^{(p)}$ for each $p \in P$ is feasible. Remaining points can be obtained using time sharing over arbitrarily smaller time scale. For each s, find the smallest k such that $s \in S_{p(k)}$ and set $b_{(v,s)} = \mu_s / |q_{p(k)}|$ if $v \in q_{p(k)}$ and 0 otherwise, thus satisfying Assumption 1. Then, for each k, $\sum_{s \in S_{p(k)}} b_{(v,s)} =$

$\mu(\{p(1),\ldots,p(k)\}) - \mu(\{p(1),\ldots,p(k-1)\}) = r^{(p)}_{p(k)}$. Thus, $\mathbf{r}^{(p)}$ is feasible.

2.7.2 Proof of Theorem 2

We prove this by induction on $|\mathbf{x}| \triangleq \sum_i x_i$. Clearly, the result is true when $|\mathbf{x}| = 1$. Lets assume that the claim is true for all $\mathbf{x}'$ such that $|\mathbf{x}'| < |\mathbf{x}|$ for a given $\mathbf{x}$. We show that it holds for $\mathbf{x}$ as well.

By definition of balanced fairness, i.e., by (5.1) and (5.2), there exists a B such that $\sum_{i\in B} r_i(\mathbf{x}) = \mu(B)$. Also, by monotonicity of $\mu(.)$, $B \subset A_{\mathbf{x}}$. If $B = A_{\mathbf{x}}$, then we are done. Suppose this is not the case. Then, from (5.1) and definition of B, we have

$$\Phi(\mathbf{x}) = \frac{\sum_{i\in B} \Phi(\mathbf{x}-\mathbf{e}_i)}{\mu(B)}. \tag{2.22}$$

Since the capacity condition $\sum_{i\in B} r_i(\mathbf{x}') \leq \mu(B)$ is satisfied for all states, we have $\sum_{i\in B} r_i(\mathbf{x}-\mathbf{e}_j) \leq \mu(B)$ for all $j \in A_{\mathbf{x}}\backslash B$. Using this in (2.22), we get

$$\Phi(\mathbf{x}) \leq \frac{\sum_{i\in B} \Phi(\mathbf{x}-\mathbf{e}_i)}{\sum_{i\in B} r_i(\mathbf{x}-\mathbf{e}_j)}, \forall j \in A_{\mathbf{x}}\backslash B. \tag{2.23}$$

We now use this bound to compute one on the sum of all rates as follows:

$$\begin{aligned}\sum_{i\in A_{\mathbf{x}}} r_i(\mathbf{x}) &= \sum_{i\in B} r_i(\mathbf{x}) + \sum_{j\in A_{\mathbf{x}}\backslash B} r_j(\mathbf{x}),\\ &= \mu(B) + \sum_{j\in A_{\mathbf{x}}\backslash B} \frac{\Phi(\mathbf{x}-\mathbf{e}_j)}{\Phi(\mathbf{x})},\\ &\geq \mu(B) + \sum_{j\in A_{\mathbf{x}}\backslash B} \frac{\sum_{i\in B} r_i(\mathbf{x}-\mathbf{e}_j)\Phi(\mathbf{x}-\mathbf{e}_j)}{\sum_{i\in B}\Phi(\mathbf{x}-\mathbf{e}_i)},\end{aligned}$$

$$
\begin{aligned}
&= \mu(B) + \sum_{j \in A_{\mathbf{x}} \setminus B} \frac{\sum_{i \in B} \Phi(\mathbf{x} - \mathbf{e}_j - \mathbf{e}_i)}{\sum_{i \in B} \Phi(\mathbf{x} - \mathbf{e}_i)}, \\
&= \mu(B) + \frac{\sum_{i \in B} \sum_{j \in A_{\mathbf{x}} \setminus B} \Phi(\mathbf{x} - \mathbf{e}_j - \mathbf{e}_i)}{\sum_{i \in B} \Phi(\mathbf{x} - \mathbf{e}_i)}, \\
&\geq \mu(B) + \frac{\sum_{j \in A_{\mathbf{x}} \setminus B} \Phi(\mathbf{x} - \mathbf{e}_j - \mathbf{e}_{i^*})}{\Phi(\mathbf{x} - \mathbf{e}_{i^*})}, \qquad (2.24)
\end{aligned}
$$

where $i^* = \arg\min_{i \in B} \left\{ \frac{\sum_{j \in A_{\mathbf{x}} \setminus B} \Phi(\mathbf{x} - \mathbf{e}_j - \mathbf{e}_i)}{\Phi(\mathbf{x} - \mathbf{e}_i)} \right\}$. In the last inequality (2.24), we have used the identity $\frac{a+b}{c+d} \geq \frac{a}{c}$ if $\frac{a}{c} \leq \frac{b}{d}$. Thus, we get the following inequality.

$$
\sum_{i \in A_{\mathbf{x}}} r_i(\mathbf{x}) \geq \mu(B) + \sum_{j \in A_{\mathbf{x}} \setminus B} r_j(\mathbf{x} - \mathbf{e}_{i^*}). \qquad (2.25)
$$

We now only need to show $\mu(B) + \sum_{j \in A_{\mathbf{x}} \setminus B} r_j(\mathbf{x} - \mathbf{e}_{i^*}) \geq \mu(A_{\mathbf{x}})$. The following two cases are possible for the given $\mathbf{x}$.

Case 1 $x_{i^*} = 1$: Then, in state $\mathbf{x} - \mathbf{e}_{i^*}$, only classes in $A_{\mathbf{x}} \setminus \{i^*\}$ are active. Thus, we have,

$$
\begin{aligned}
&\sum_{j \in A_{\mathbf{x}} \setminus B} r_j(\mathbf{x} - \mathbf{e}_{i^*}) + \mu(B) \\
&\qquad = \mu(A_{\mathbf{x}} \setminus \{i^*\}) - \sum_{k \in B \setminus \{i^*\}} r_k(\mathbf{x} - \mathbf{e}_{i^*}) + \mu(B), \\
&\qquad \geq \mu(A_{\mathbf{x}} \setminus \{i^*\}) - \mu(B \setminus \{i^*\}) + \mu(B), \\
&\qquad \geq \mu(A_{\mathbf{x}}),
\end{aligned}
$$

where the equality follows from induction hypothesis, the first inequality follows from the capacity constraint on set $B \setminus \{i^*\}$, and the last inequality follows from the submodularity of $\mu(.)$.

<u>**Case 2**</u> $x_{i^*} > 1$: Here, all the classes in $A_{\mathbf{x}}$ are active in state $\mathbf{x} - \mathbf{e}_{i^*}$ as well, i.e., $A_{\mathbf{x}} = A_{\mathbf{x}-e_{i^*}}$. Thus, we have,

$$\begin{aligned}\sum_{j\in A_{\mathbf{x}}\setminus B} r_j(\mathbf{x}-\mathbf{e}_{i^*}) + \mu(B) &\geq \sum_{i\in A_{\mathbf{x}}} r_i(\mathbf{x}-\mathbf{e}_{i^*}) \\ &= \mu(A_{\mathbf{x}}),\end{aligned}$$

where the inequality follows from the capacity constraint on set B, and the equality follows from induction hypothesis. Thus, the result holds for both the cases.

2.7.3 Proof of Theorem 4

We prove (2.18) first and then (4.2).

Proof of (2.18): We first prove the following lemma by finding an explicit expression for $\pi_k^{(m,n)}$ for each k for given m and n and then taking the limit as $n \to \infty$ for a fixed m. Let $\lim_{n\to\infty} \pi_k^{(m,n)} = \pi_k^{(m,\infty)}$. Also let $h^{(m,\infty)}(k) = \xi m(1-(1-c/m)^k)$ for $k = 0, 1, 2, \ldots, \infty$.

Lemma 1. *For any fixed integers k_1 and k_2 such that $k_1 > k_2$, we have*

$$\frac{\pi_{k_1}^{(m,\infty)}}{\pi_{k_2}^{(m,\infty)}} = \frac{(m\rho)^{k_1-k_2}}{\prod_{l=k_2+1}^{k_1} h^{(m,\infty)}(l)} \tag{2.26}$$

Proof. Fix m and n. From definition of $F_k(.)$ in the proof of Corollary 1 one can show that

$$\pi_k^{(m,n)} = \frac{F_k(m\rho/n)}{F(m\rho/n)} \text{ for } k = 1, \ldots, n \tag{2.27}$$

where $F_k(m\rho/n)$ and $F(m\rho/n)$ are given by recursive expressions in the statement of Corollary 1. Thus, from (2.13), we get $\pi_0^{(m,n)} = 1/F(m\rho/n)$ and

$$\pi_k^{(m,n)} = \frac{(n-k+1)\frac{m\rho}{n}\pi_{k-1}^{(m,n)}}{h^{(m,n)}(k) - k\frac{m\rho}{n}}, \text{ for } k = 1, \dots, n.$$

Thus, for any $k_1 > k_2$ we get

$$\begin{aligned}\frac{\pi_{k_1}^{(m,n)}}{\pi_{k_2}^{(m,n)}} &= \frac{(n-k_2)!(\frac{m\rho}{n})^{k_1-k_2}}{(n-k_1)!\prod_{l=k_2+1}^{k_1}(h^{(m,n)}(l) - l\frac{m\rho}{n})} \\ &\underset{n\to\infty}{\longrightarrow} \frac{(m\rho)^{k_1-k_2}}{\prod_{l=k_2+1}^{k_1} h^{(m,\infty)}(l)}\end{aligned}$$

□

Now let us study $h^{(m,\infty)}$ and $\pi_k^{(m,\infty)}$ in the limit as $m \to \infty$. For any $\alpha > 0$, we have

$$\lim_{m\to\infty} \frac{1}{m} h(\lfloor \alpha m \rfloor) = \xi(1 - e^{-\alpha c}).$$

Let $k^{(m)}$ be the largest k such that $h^{(m,\infty)}(k) \leq m\rho$. Thus, it is easy to show that $k^{(m)}/m \to \alpha^*$ as $m \to \infty$ where α^* is given by (2.17).

Now for some large enough γ, consider the following four cases: (1) $0 \leq k < (1-2\epsilon)k^{(m)}$, (2) $(1-2\epsilon)k^{(m)} \leq k \leq (1+2\epsilon)k^{(m)}$, (3) $(1-2\epsilon)k^{(m)} < k \leq \gamma m$, and (4) $k > \gamma m$. Our approach now onwards can be summarized as follows. We first consider the case (4) and show that by choosing γ large enough the tail probability $\sum_{l:l>\gamma m} \pi_l^{(m,\infty)}$ can be made arbitrarily small, independent of m. For the remaining three cases, we then show that $\pi_k^{(m,\infty)}$ concentrates on the second case as m increases to ∞.

Lemma 2. *For any $\delta > 0$, there exists a constant γ such that*

$$\sum_{l:l>\gamma m} \pi_l^{(m,\infty)} \leq \pi_{k^{(m)}}^{(m,\infty)} \delta$$

for all m.

Proof. Find the smallest α such that αm is an integer and $h^{(m,\infty)}(\alpha m) \geq m\rho(1+\epsilon')$ for some fixed $\epsilon' > 0$. Since $\alpha m \geq k^{(m)}$, we have $\pi_{\alpha m}^{(m,\infty)} \leq \pi_{k^{(m)}}^{(m,\infty)}$. Also, it is easy to check that α is $O(1)$, i.e., it does not scale with m. By monotonicity of h, $h^{(m,\infty)}(k) \geq m\rho(1+\epsilon')$ for each $k \geq \alpha m$. From (2.26), for each $k \geq \alpha m$, we get

$$\pi_k^{(m,\infty)} \leq \pi_{\alpha m}^{(m,\infty)} (\frac{1}{1+\epsilon'})^{k-\alpha m}.$$

Also, for each $k > \alpha m$,

$$\begin{aligned}
\sum_{l:l\geq k} \pi_l^{(m,\infty)} &\leq \pi_k^{(m,\infty)} \sum_{l=1}^{\infty} \left(\frac{1}{1+\epsilon'}\right)^l \\
&= \pi_k^{(m,\infty)} \frac{1}{1-1/(1+\epsilon')} \\
&\leq \pi_{\alpha m} (\frac{1}{1+\epsilon'})^{k-\alpha_m} \frac{1}{1-1/(1+\epsilon')} \\
&\leq \pi_{k^{(m)}}^{(m,\infty)} c' \left(\frac{1}{1+\epsilon'}\right)^{k-\alpha m},
\end{aligned}$$

for some constant c'. Putting $k = \gamma m$, we get,

$$\sum_{l:l\geq \gamma m} \pi_l^{(m,\infty)} \leq c' \pi_{k^{(m)}}^{(m,\infty)} \left(\frac{1}{1+\epsilon'}\right)^{(\gamma-\alpha)m}$$

Thus, for any $\delta > 0$, by choosing γ large enough one can ensure that $\sum_{l\geq \gamma m} \pi_l^{(m,\infty)} \leq \pi_{k^{(m)}}^{(m,\infty)} \delta$ for all m. □

We now prove the following lemma from which (2.18) follows since ϵ can be chosen arbitrarily and $k^{(m)}/m \to \alpha^*$ as $m \to \infty$.

Lemma 3. *For any $\epsilon > 0$, we have*

$$\lim_{m\to\infty} \frac{\sum_{k=0}^{\infty} \pi_k^{(m,\infty)}}{\sum_{k=(1-2\epsilon)k^{(m)}}^{(1+2\epsilon)k^{(m)}} \pi_k^{(m,\infty)}} = 1$$

Proof. By monotonicity of $h^{(m,\infty)}$, $h^{(m,\infty)}(k) \le h^{(m,\infty)}((1-2\epsilon)k^{(m)})$ for all $k \le (1-2\epsilon)k^{(m)}$. Using (2.26) with $k_1 = (1-\epsilon)k^{(m)}$ and with any $k_2 \le (1-2\epsilon)k^{(m)}$, we get,

$$\begin{aligned}
\frac{\pi^{(m,\infty)}_{(1-\epsilon)k^{(m)}}}{\pi^{(m,\infty)}_{k_2}} &= \frac{(m\rho)^{(1-\epsilon)k^{(m)}-k_2}}{\prod_{l=k_2+1}^{(1-\epsilon)k^{(m)}} h^{(m,\infty)}(l)} \\
&\ge \left(\frac{m\rho}{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}\right)^{(1-\epsilon)k^{(m)}-k_2} \\
&\ge \left(\frac{m\rho}{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}\right)^{(1-\epsilon)k^{(m)}-(1-2\epsilon)k^{(m)}} \\
&\ge \left(\frac{m\rho}{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}\right)^{\epsilon k^{(m)}}
\end{aligned}$$

Similarly, $h^{(m,\infty)}(k) \ge h^{(m,\infty)}((1+2\epsilon)k^{(m)})$ for all $k \ge (1+2\epsilon)k^{(m)}$. Using (2.26) with any $k_1 \ge (1+2\epsilon)k^{(m)}$ and with $k_2 = (1+\epsilon)k^{(m)}$, we get,

$$\begin{aligned}
\frac{\pi^{(m,\infty)}_{k_1}}{\pi^{(m,\infty)}_{(1+\epsilon)k^{(m)}}} &= \frac{(m\rho)^{k_1-(1+\epsilon)k^{(m)}}}{\prod_{(1+\epsilon)k^{(m)}}^{k_1} h^{(m,\infty)}(l)} \\
&\le \left(\frac{m\rho}{h^{(m,\infty)}((1+2\epsilon)k^{(m)})}\right)^{k_1-(1+\epsilon)k^{(m)}} \\
&\le \left(\frac{m\rho}{h^{(m,\infty)}((1+2\epsilon)k^{(m)})}\right)^{\epsilon k^{(m)}}
\end{aligned}$$

Thus, we get,

$$
\begin{aligned}
&\frac{\sum_{k=0}^{\infty} \pi_k^{(m,\infty)}}{\sum_{k=(1-2\epsilon)k^{(m)}}^{(1+2\epsilon)k^{(m)}} \pi_k^{(m,\infty)}} \\
=&\left(\sum_{k=(1-2\epsilon)k^{(m)}}^{(1+2\epsilon)k^{(m)}} \pi_k^{(m,\infty)}\right)^{-1} \left(\sum_{k<(1-2\epsilon)k^{(m)}} + \sum_{k=(1-2\epsilon)k^{(m)}}^{(1+2\epsilon)k^{(m)}} + \sum_{k=(1+2\epsilon)k^{(m)}+1}^{\gamma m} + \sum_{k>\gamma m}\right) \pi_k^{(m,\infty)} \\
\leq&\frac{\sum_{k<(1-2\epsilon)k^{(m)}} \pi_k^{(m,\infty)}}{\pi_{(1-\epsilon)k^{(m)}}^{(m,\infty)}} + 1 + \frac{\sum_{k=(1+2\epsilon)k^{(m)}+1}^{\gamma m} \pi_k^{(m,\infty)}}{\pi_{(1+\epsilon)k^{(m)}}^{(m,\infty)}} + \delta \\
\leq&(1-2\epsilon)k^{(m)} \left(\frac{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}{m\rho}\right)^{\epsilon k^{(m)}} + 1 \\
&+ (\gamma m - (1+2\epsilon)k^{(m)}) \left(\frac{m\rho}{h^{(m,\infty)}((1+2\epsilon)k^{(m)})}\right)^{\epsilon k^{(m)}} + \delta \\
\xrightarrow[m\to\infty]{}& \ 0 + 1 + 0 + \delta
\end{aligned}
$$

Where the last limit can be shown to hold as follows. Using $k^{(m)}/m \to \alpha^*$ as $m \to \infty$, one can show that $\lim_{m\to\infty} h^{(m,\infty)}((1-2\epsilon)k^{(m)})/(\rho m) = \xi(1 - e^{-(1-2\epsilon)\alpha^* c}) < \xi(1 - e^{-\alpha^* c}) = 1$. Thus, there exists $c_1 < 1$ and $m' > 0$ such that the inequality $\frac{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}{m\rho} < c_1$ holds for all $m > m'$. Similarly, there exists $c_2 < 1$ and $m'' > 0$ such that the inequality $\frac{m\rho}{h^{(m,\infty)}((1+2\epsilon)k^{(m)})} < c_2$ holds for all $m > m'$. Thus, terms $\left(\frac{h^{(m,\infty)}((1-2\epsilon)k^{(m)})}{m\rho}\right)^{\epsilon k^{(m)}}$ and $\left(\frac{m\rho}{h^{(m,\infty)}((1+2\epsilon)k^{(m)})}\right)^{\epsilon k^{(m)}}$ tend to 0 geometrically fast. Since $\epsilon > 0$ and $\delta > 0$ where chosen arbitrarily, the lemma holds. □

Proof of (4.2): To find mean delay, we cannot use Little's law just yet, since we have shown concentration in $\pi_k^{(m,n)}$ which is the probability measure for number of active classes and not number of waiting requests. However, intuitively, by increasing n while keeping ρ fixed, we are thinning the arrival process of each class so that the probability of having more than one waiting job for any given class at any given point in time goes to 0. By taking the limit as $n \to \infty$, $\pi_k^{(m,n)}$ then becomes a proxy for the number of waiting jobs. To prove the result formally, we use expression for mean delay in Corollary 1. Define

$$\tau_k^{(m,n)} = \frac{\hat{F}_k(m\rho/n)}{nF(m\rho/n)}.$$

Then, using (2.11) and (2.14) from Corollary 1 and using $\nu_i = \nu$ for all i, the mean delay for a given n and m is given by

$$E\left[D^{(m,n)}\right] = \nu \sum_{k=0}^{n} k\tau_k^{(m,n)}. \tag{2.28}$$

Let $\lim_{n\to\infty} \tau_k^{(m,n)} = \tau_k^{(m,\infty)}$. We now prove the following lemma by induction on k.

Lemma 4.

$$\tau_k^{(m,\infty)} = \frac{\pi_k^{(m,\infty)}}{m\rho} \textit{ for } k = 1, 2, \ldots$$

Proof. For a given n, from (2.13), (2.14) and (2.27) we get

$$\tau_k^{(m,n)} = \frac{\frac{1}{n}\pi_k^{(m,n)} + \frac{n-k+1}{nk}\pi_{k-1}^{(m,n)} + \frac{(n-k+1)(k-1)m\rho}{nk}\tau_{k-1}^{(m,n)}}{h^{(m,n)}(k) - km\rho/n}$$

for $k = 1, 2, \ldots, n$ and $\tau_0^{(m,n)} = 0$. By taking limits as $n \to \infty$, we get

$$\tau_k^{(m,\infty)} = \frac{\frac{1}{k}\pi_{k-1}^{(m,\infty)} + \frac{(k-1)m\rho}{k}\tau_{k-1}^{(m,\infty)}}{h^{(m,\infty)}(k)},$$

for any $k \geq 1$, and $\tau_0^{(m,\infty)} = 0$ Now we prove the lemma by induction using the above recursion. First, we prove the result for the base case of $k = 1$. By direct substitution we get,

$$\begin{aligned}\tau_1^{(m,\infty)} &= \frac{\pi_0^{(m,\infty)} + 0}{h(1)} \\ &= \frac{\pi_1^{(m,\infty)} \frac{h(1)}{m\rho}}{h(1)},\end{aligned}$$

where the last equality follows from (2.26). Thus, we get $\tau_1^{(m,\infty)} = \pi_1^{(m,\infty)}/(m\rho)$. Now, assume the result is true for $\tau_{k-1}^{(m,\infty)}$. Thus we get,

$$\begin{aligned}\tau_k^{(m,\infty)} &= \frac{\frac{1}{k}\pi_{k-1}^{(m,\infty)} + \frac{(k-1)}{k}\pi_{k-1}^{(m,\infty)}}{h^{(m,\infty)}(k)} \\ &= \frac{\pi_{k-1}^{(m,\infty)}}{h^{(m,\infty)}(k)} \\ &= \frac{\pi_k^{(m,\infty)}}{m\rho},\end{aligned}$$

where the last equality again follows from (2.26). □

Thus from (2.28), we get,

$$\lim_{n\to\infty} E\left[D^{(m,n)}\right] = \frac{\sum_{k=1}^{\infty} k\pi_k^{(m,\infty)}}{\lambda m}.$$

Proofs of Lemma 2 and 3 show that the probability $\pi_k^{(m,\infty)}$ for $k < (1-2\epsilon)k^{(m)}$ or $k > (1+2\epsilon)k^{(m)}$ decreases to 0 geometrically fast with m. Thus, proceeding along similar lines, one can show that

$$\lim_{m\to\infty} \lim_{n\to\infty} E\left[D^{(m,n)}\right] \in \lambda^{-1}[\alpha^* - 2\epsilon, \alpha^* + 2\epsilon].$$

for any $\epsilon > 0$. Hence, the result.

Chapter 3

Impact of Fairness and Heterogeneity on Delays

In many shared network systems service rate is allocated to ongoing jobs based on a fairness criterion, e.g., α-fair (αF) (including max-min and proportional fair) as well as Balanced fair (BF), and other Greedy criteria [62]. When the network loads are stochastic a key open question is how the choice of fairness and network design will impact user perceived performance, e.g., job delays, as well as the sensitivity of performance to heterogeneity in network resources and traffic loads. Motivated by this challenge in this chapter we take a step towards understanding these issues by investigating performance bounds for an interesting class of stochastic networks with symmetric polymatroid capacity under various fairness criteria.

The second question driving this chapter is whether large scale systems can be designed to be inherently robust to heterogeneity and at what cost? Specifically we consider content delivery systems where a large collection servers deliver a proportionally large number of files. There has been

[1] A version of this work appeared as "Impact of Fairness and Heterogeneity on Delays in Large-scale Content Delivery Networks," in Proceedings of ACM Sigmetrics 2015 . This is a joint work with Prof. Gustavo de Veciana.

substantial recent interest in understanding basic design questions for these systems including, see e.g. [34, 45, 53] and references therein: How should the number of file copies scale with the demand? What kinds of hierarchical caching policies are most suitable? How to best optimize storage/backhaul costs for unpredictable time-varying demands? Our focus is on content delivery systems that permit parallel file downloads from multiple servers – akin to peer-to-peer systems. In principle with an appropriate degree of storage redundancy, one can achieve much better peak service rates, exploit diversity in service paths, produce robustness to failures, and provide better sharing of pooled server resources. Intuitively when such content delivery systems have sufficient redundancy they will exhibit performance which is robust to limited heterogeneity in demands and server capacity, as well as to the fairness criterion driving resource allocation. Such systems might also circumvent the need for, and overheads (such as backhaul, state update, etc) associated with, dynamic caching. If this is the case, content delivery systems enabling parallel servicing of individual download requests could be more scalable and robust to serving popular content.

3.0.4 Our Contributions and Organization

The contributions of this chapter are threefold, each of independent interest, and collectively, providing a significant step forward over what is known in the current literature.

a.) *Performance bounds:* In Sections 3.2-3.3. we consider a class of systems with symmetric polymatroid capacity for which we develop several resource allocation monotonicity properties which translate to performance comparisons amongst fairness policies, and eventually give explicit bounds on mean delays. Specifically we show that under homogeneous loads the mean delay achieved by Greedy and αF resource allocations are bounded by that of BF allocation which is computable. We then extend this upper bound to the case when the load is heterogeneous but 'majorized by a symmetric load.'

b.) *Uniform symmetry in large systems:* In Section 3.4 we consider a bipartite graph where nodes represent n job classes (files) and m servers with potentially heterogenous service capacity. The graph edges capture the ability of servers to serve the jobs in the given classes. If jobs can be concurrently served by multiple servers the system's service capacity region is polymatroid. We show that for appropriately scaled large system where the edge set is chosen at random (random file placement) the capacity region is uniformly close to a *symmetric* polymatroid.

c.) *Performance robustness of large systems:* Combining these two results, in Section 3.5 we provide a simple performance bound for large-scale systems. The bound exhibits performance robustness in such systems with respect to variations in total system load, heterogeneity in load across the classes, heterogeneity in server capacities, for α-fair based resource allocation. Specifically it establishes a clear link between the degree of content replication and permissible demand heterogeneity while ensuring performance scalability.

We have have deferred some technical results to the appendix.

3.1 Related work

There is a substantial amount of related work. Yet the link between fairness in resource allocation and job delays in stochastic networks is poorly understood. The only fairness criterion for which explicit expressions or bounds are known is the Balanced Fair resource allocation [7] which generalizes the notion of 'insensitivity' of the processor sharing discipline in $M/G/1$ queuing system. Under balanced fairness, an explicit expression for mean delay was obtained in [10,11] for a class of wireline networks, namely, those with line and tree topologies. Also, a performance bound for arbitrary polytope capacity region and arbitrary load was provided in [4]. Similarly [25] developed bounds for stochastic networks where flows can be split over multiple paths. These bounds and expressions are either too specific or too loose. Recently, [47] developed an expression for the mean delay for systems with polymatroid capacity and arbitrary loads under Balanced Fair resource allocations. Unfortunately the result has exponential computational complexity in general. However the symmetric case has low complexity, a fact we use in the sequel.

Balanced fair resource allocation is defined recursively and is difficult to implement. α-fair resource allocations [32, 44] which are based on maximizing a concave sum utility function over the system's capacity region – this includes proportional and max-min fair allocations, are more amenable to implementation [30,36]. However, the only known explicit performance results for stochastic networks under such fairness criteria are for systems where proportional fair is equivalent to balanced fair [7,40]. In [6], performance relationship under balanced and proportional fairness for several systems where they are not equivalent was studied through numerical computations, and were found to be relatively close in several scenarios.

In this chapter we focus on a class of stochastic networks that can be characterized by a polymatroid capacity region. Such systems have also been considered in [62]. For example, the work in [62] shows that when such systems are symmetric with respect to load and capacity, a greedy resource allocation is delay optimal. However, the result is brittle to asymmetries. aWe provide more details on greedy and other resource allocations in Section 3.2.

In summary when it comes to fairness criteria and stochastic network performance there is a gap between what is implementable and what is analyzable. One of the goals of this chapter is to provide comparison results which address this gap, with particular focus on addressing user-performance in large-scale systems prevalent today.

In terms of robustness to heterogeneities, the work that is closest to this chapter is [53,60], where it is shown that if the graph is chosen at random and scaled appropriately then user-performance is robust to load heterogeneity. In [60] a service model is considered where each request can be served by a single server – recall we consider systems allowing parallel downloads. The resource pooling in our service model leads to a significantly improved mean delay bound and the resulting robustness.

We assume the same service model as in previous chapter, but is different in several respects. First, we focused on mean delay for CDNs only under Balanced fair resource allocation whereas we directly study the impact of fairness criteria on users delays. Second, the system was by design symmetric in asymptotic regime in previous chapter whereas here we establish the asymptotic symmetry. Thirdly, in this chapter we establish new results on robustness to limited heterogeneity in file demands, server capacity and α-fairness criteria by providing a uniform bound on delays.

3.2 Resource allocation policies: a background

There are several possible resource allocation policies, each resulting in potentially different user-perceived delays. In this chapter, we introduce three different policies studied in literature, each with its own merits. In comparing them, we will rely on notation for ordering and majorization which we introduce below, some of which are borrowed from [38] and [62].

Notation for ordering and majorization:

Let I be a finite arbitrary index set. Consider an arbitrary vector $\mathbf{z} = (z_i : i \in I)$. We let $z_{[1]} \geq z_{[2]} \geq \ldots, z_{[|I|]}$ denote the components of $\mathbf{z}$ in decreasing order. We let $|\mathbf{z}|$ denote $\sum_{i \in I} |z_i|$. We let $\mathbf{e}_i$ denote a vector with

1 at the i^{th} coordinate and 0 elsewhere.

For vectors $\mathbf{z}$ and $\mathbf{z}'$ such that $z_i \leq z'_i$ for each $i \in I$, we write $\mathbf{z} \leq \mathbf{z}'$ and say that $\mathbf{z}$ is *dominated* by $\mathbf{z}'$.

Below we define *majorization* ($\prec$) which describes how 'balanced' a vector is as compared to another vector. In words, by $\mathbf{z} \prec \mathbf{z}'$ we mean that $\mathbf{z}$ is 'more balanced' than $\mathbf{z}'$ but they have the same sum. By $\mathbf{z} \prec_w \mathbf{z}'$ we mean that $\mathbf{z}$ is 'more balanced' and has lower sum than $\mathbf{z}'$. Similarly, by $\mathbf{z} \prec^w \mathbf{z}'$ we mean that $\mathbf{z}$ is 'more balanced' and has larger sum than $\mathbf{z}'$.

Definition 1. *For vectors* $\mathbf{z}$ *and* $\mathbf{z}'$ *such that* $|\mathbf{z}| = |\mathbf{z}'|$ *and* $\sum_{l=1}^{k} z_{[l]} \leq \sum_{l=1}^{k} z'_{[l]}$ *for each* $k \in \{1, 2, \ldots, |I|\}$*, we say* $\mathbf{z}$ *is* majorized *by* $\mathbf{z}'$*, and denote this as* $\mathbf{z} \prec \mathbf{z}'$*.*

If we have $\sum_{l=1}^{k} z_{[l]} \leq \sum_{l=1}^{k} z'_{[l]}$ *for each* $k \in \{1, 2, \ldots, |I|\}$*, we say* $\mathbf{z}$ *is* weak-majorized from below *by* $\mathbf{z}'$*, and denote this as* $\mathbf{z} \prec_w \mathbf{z}'$*.*

Similarly, if we have $\sum_{l=0}^{k} z_{[|I|-l]} \geq \sum_{l=1}^{k} z'_{[|I|-l]}$ *for each* $k \in \{0, 1, \ldots, |I|-1\}$*, we say* $\mathbf{z}$ *is* weak-majorized from above *by* $\mathbf{z}'$*, and denote this as* $\mathbf{z} \prec^w \mathbf{z}'$*.*

The dominance and majorization have an associated stochastic version, defined below.

Definition 2. *Consider random vectors* $\mathbf{Z}$ *and* $\mathbf{Z}'$*. If there exist random vectors* $\tilde{\mathbf{Z}}$ *and* $\tilde{\mathbf{Z}}'$ *such that* $\mathbf{Z}$ *and* $\tilde{\mathbf{Z}}$ *are identically distributed,* $\mathbf{Z}'$ *and* $\tilde{\mathbf{Z}}'$ *are identically distributed, and* $\tilde{\mathbf{Z}}' \leq \tilde{\mathbf{Z}}'$ *almost surely, then we say that* $\mathbf{Z}$ *is* stochastically dominated *by* $\mathbf{Z}'$*, and denote this as* $\tilde{\mathbf{Z}} \leq^{st} \tilde{\mathbf{Z}}'$*.*

Instead, if $\tilde{\mathbf{Z}}' \prec_w \tilde{\mathbf{Z}}'$*, then we say that* $\mathbf{Z}$ stochastically weak-majorized from below *by* $\mathbf{Z}'$*, and denote this as* $\tilde{\mathbf{Z}} \prec_w^{st} \tilde{\mathbf{Z}}'$*.*

In the sequel, it will be useful to introduce following notation. Recall, $\mathbf{r}(\mathbf{x}) = (r_i(\mathbf{x}) : i \in F)$ is the vector of rates allocated to various classes. We define $r_{(k)}(.)$ for each $k \in \{1, \dots, n\}$ as follows: For a given state $\mathbf{x}$, let i_k be the class corresponding to $x_{[k]}$. Then, $r_{(k)}(\mathbf{x}) = r_{i_k}(\mathbf{x})$. In words, $r_{(k)}(\mathbf{x})$ is the rate allocated to the class with the k^{th} largest number of ongoing jobs.

Below provide a brief description of the resource allocation policies considered in this chapter.

1) Greedy resource allocation: Roughly, the Greedy resource allocation policy on a polymatroid capacity region $\mathcal{C}$ assigns the maximum possible rate to the largest queues subject to the capacity constraints. We denote the Greedy resource allocation by $\mathbf{r}^G(.)$ and define it as follows: for each state $\mathbf{x}$, we let

$$\begin{aligned} r^G_{(k)}(\mathbf{x}) &= \mu\left(\{[1], [2], \dots, [k]\}\right) - \mu\left(\{[1], [2], \dots, [k-1]\right) \\ &\qquad\qquad\qquad\qquad \text{if } k \in \{1, 2, \dots, |A_{\mathbf{x}}|\}, \\ &= 0 \text{ otherwise.} \end{aligned}$$

Equivalently, the sum rate assigned to the k largest queues, namely $\sum_{l=1}^{k} r^G_{(l)}(\mathbf{x})$, is equal to $\mu\left(\{[1], [2], \dots, [k]\}\right)$. The Greedy resource allocation for symmetric polymatroid capacity regions was first studied in [62] where the following result was shown.

Proposition 1. ([62]) *Suppose the capacity region $\mathcal{C}$ is a symmetric polymatroid and the load $\boldsymbol{\rho} \in \hat{\mathcal{C}}$ is homogeneous, i.e., $\rho_i = \rho$ for each $i \in F$. Further suppose that service requirement of jobs for each class are exponential with mean ν. Then the following statements hold:*

1. *Let $(\mathbf{X}^G(t) : t \geq 0)$ and $(\tilde{\mathbf{X}}(t) : t \geq 0)$ be state processes under Greedy and an arbitrary feasible resource allocation, respectively. If $\mathbf{X}^G(0) \prec_w^{st} \tilde{\mathbf{X}}(0)$ then $\mathbf{X}^G(t) \prec_w^{st} \tilde{\mathbf{X}}(t)$ for each $t \geq 0$.*

2. *The mean job delay under Greedy resource allocation is less than or equal to that under any feasible resource allocation.*

Unfortunately, this optimality result for symmetric systems does not provide any explicit performance characterization or bound. Further, the result is brittle to heterogeneity in load or capacity.

2) *α-fair resource allocation*: As introduced in [44], this policy allocates rates based on maximizing a concave sum utility function subject to the system's capacity region. Formally, for a given $\alpha > 0$, the α-fair (αF) resource allocation $\mathbf{r}^\alpha(.)$, can be defined as follows: for each state $\mathbf{x}$, let

$$\mathbf{r}^\alpha(\mathbf{x}) = \begin{cases} \arg\max_{\hat{\mathbf{r}} \in \mathcal{C}} \sum_{i \in F} \frac{x_i^\alpha \, \hat{r}_i^{1-\alpha}}{1-\alpha} & \text{for } \alpha \in (0, \infty) \backslash \{1\}, \\ \arg\max_{\hat{\mathbf{r}} \in \mathcal{C}} \sum_{i \in F} x_i \log(\hat{r}_i) & \text{for } \alpha = 1. \end{cases} \tag{3.1}$$

This generalizes various notions of fairness across jobs, e.g., proportional fair and max-min fair allocations are equivalent to the α-fair policy for $\alpha = 1$ and $\alpha \to \infty$, respectively [44]. However, for polymatroid capacity regions we establish the following result. For its proof, see 3.6.1

Proposition 2. *All α-fair resource allocations are equivalent for polymatroid capacity regions.*

This is a generalization of equivalence of αF policies for a single server system where they reduce to equal share. Such an equivalence is also known for tree networks [7] which form a special case to our system. Further, the stability results in [17, 39] implies that the αF resource allocation results in a stationary process $(\mathbf{X}(t) : t \in \mathbb{R})$ when $\boldsymbol{\rho} \in \hat{\mathcal{C}}$. The α-fair resource allocation is attractive in that it is amenable to distributed implementation [30, 36] and satisfies natural axioms for fairness [32]. Unfortunately, little is known regarding their performance under stochastic loads. What has been shown is that for α-fair allocations, the performance is *sensitive* to the distribution of service requirements [7]. Thus, it will be hard to make general claims. This leads us to the Balanced fair resource allocation below.

3) Balanced fair resource allocation: We described balanced fairness in Section 2.3. For completeness, we briefly reiterate its definition:

$$r_i^B(\mathbf{x}) = \frac{\Phi(\mathbf{x} - \mathbf{e}_i)}{\Phi(\mathbf{x})},\ \forall i \in F \tag{3.2}$$

where the function Φ is called a balance function and is defined recursively as follows: $\Phi(\mathbf{0}) = 1$, and $\Phi(\mathbf{x}) = 0\ \forall \mathbf{x}$ s.t. $x_i < 0$ for some i, otherwise,

$$\Phi(\mathbf{x}) = \max_{A \subset F} \left\{ \frac{\sum_{i \in A} \Phi(\mathbf{x} - \mathbf{e}_i)}{\mu(A)} \right\}. \tag{3.3}$$

It was shown in [6,7] that if $\boldsymbol{\rho} \in \hat{\mathcal{C}}$, the process $(\mathbf{X}^B(t) : t \in \mathbb{R})$ is asymptotically stationary. Further, its stationary distribution is given by

$$\pi(\mathbf{x}) = \frac{\Phi(\mathbf{x})}{G(\boldsymbol{\rho})} \prod_{i \in A_{\mathbf{x}}} \rho_i^{x_i} \quad \text{where} \quad G(\boldsymbol{\rho}) = \sum_{\mathbf{x}'} \Phi(\mathbf{x}') \prod_{i \in A_{x'}} \rho_i^{x'_i}.$$

The existence of such an expression for stationary distribution makes balanced fairness amenable for time-averaged performance analysis, a property we will use extensively in the sequel. In Section 2.3, we used this distribution to develop an exact expression for mean delays for system with polymatroid capacity region. In fact, Corollary 1 provides an easily computable expression for mean delays, with complexity $O(n)$, under symmetry in load and capacity region.

Further, we use several other properties of these resource allocation policies in the sequel, some of which are given in Section 3.6.2.

3.3 Performance comparison and bounds

In this section, we provide a comparison result for Greedy, αF, and BF resource allocation policy and and develop explicit and easily computable bounds on the mean delay of jobs in systems with Greedy or αF resource allocation under potentially heterogeneous load $\boldsymbol{\rho}$ within a subset of the stability region $\hat{\mathcal{C}}$. We will make the following assumption for the remainder of this section.

Recall that for each resource allocation policy considered in Section 3.2, namely Greedy, αF, and BF, the underlying state process is asymptotically stationary if the load $\boldsymbol{\rho} \in \hat{\mathcal{C}}$. Thus the corresponding mean delays of the system's jobs are finite. In this section, we assume that the *capacity region $\mathcal{C}$ is symmetric*, and develop explicit and easily computable bounds on the mean delay of jobs in systems with Greedy or αF resource allocation under potentially heterogeneous load $\boldsymbol{\rho}$ within a subset of the stability region $\hat{\mathcal{C}}$.

Our goal here is to enable performance analysis for a general enough class of systems so as to allow us to develop quantitative and qualitative insights for large-scale systems prevalent today. For example, the bounds developed below will enable us to later characterize user-performance in downloading files from heterogeneous (in loads and service capacities) large-scale CDNs supporting parallel servicing of downloads.

Below we develop performance bounds for the following three cases:

(i) *Homogeneous loads:* We provide an upper bound for mean delay for loads $\boldsymbol{\rho} \in \hat{\mathcal{C}}$ which are *homogeneous across classes with non-zero entries*, i.e., if A is the set of classes such that $\rho_i > 0$ for each $i \in A$, then $\rho_i = \rho_j$ for each $i, j \in A$.

(ii) *Dominance bound:* Consider loads $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \hat{\mathcal{C}}$ such that $\boldsymbol{\rho} \leq \boldsymbol{\rho}'$ and $\boldsymbol{\rho}'$ is homogeneous across non-zero entries as described above. Then, we show that the system with load $\boldsymbol{\rho}$ has lower mean delay than that with load $\boldsymbol{\rho}'$, even if $\boldsymbol{\rho}$ is heterogeneous.

(iii) *Majorization bound:* Consider loads $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \hat{\mathcal{C}}$ such that $\boldsymbol{\rho} \prec \boldsymbol{\rho}'$. Further, suppose that $\boldsymbol{\rho}'$ is homogeneous across non-zero entries as described above.

Then, we show that the system with load $\boldsymbol{\rho}$ has lower mean delay than that with load $\boldsymbol{\rho}'$.

Throughout this section, we will assume that the mean service requirements for jobs ν is same for each system. Using the above majorization bound, we can bound mean delay for a larger subset of heterogeneous loads as compared to the dominance bound. For example, consider $\boldsymbol{\rho} = (\rho, \frac{1}{2}\rho, \frac{1}{2}\rho)$. Recall, for symmetric rank functions we have $\mu(A) = h(|A|)$ for each $A \subset F$, where $h(.)$ is concave. Now, if $\frac{1}{3}h(3) < \rho < \frac{1}{2}h(2)$, then $\boldsymbol{\rho}' = (\rho, \rho, 0)$ is in $\hat{\mathcal{C}}$ but $\boldsymbol{\rho}'' = (\rho, \rho, \rho)$ is not. Then the majorization bound holds for $\boldsymbol{\rho}$ but the dominance bound does not. Further, even if $\boldsymbol{\rho}''$ is in $\hat{\mathcal{C}}$, the bound obtained through $\boldsymbol{\rho}'$ may be tighter.

The bounds for each case will be obtained through coupling arguments on the corresponding state processes, followed by an application of Little's law.

3.3.1 Homogeneous Loads

Consider the following set of loads:

$$\mathcal{B}_H \triangleq \{\boldsymbol{\rho} \in \hat{\mathcal{C}} : \exists A \subset F \text{ s.t. } \rho_i = \rho_j \ \forall i, j \in A \text{ and } \rho_i = 0 \ \ \forall i \in F \backslash A\}.$$

Since by Proposition 1 the Greedy resource allocation is delay optimal for homogeneous loads, for each $\boldsymbol{\rho} \in \mathcal{B}_H$ one can immediately conclude that the performance of BF as obtained in Corollary 1 is an upper bound for Greedy. Below we show that this performance upper bound via BF also holds for αF resource allocation.

To that end we show a coupling result for systems under αF and BF resource allocations. In the process, we prove and use the property that αF is more greedy than BF in the following sense: if the state process corresponding to αF is same as or more balanced than that of BF, then αF assigns larger rate to bigger queues than BF. This in turn keeps the state process for αF more balanced in the future. For a proof of the theorem below see Section 3.3.4.

Theorem 5. *Consider a system with symmetric polymatroid capacity region and load $\boldsymbol{\rho} \in \mathcal{B}_H$, i.e., $\boldsymbol{\rho}$ is homogeneous across classes with non-zero entries, and that service requirement of jobs are exponentially distributed with mean ν. Then the following statements hold:*

1. *Let $(\mathbf{X}^{\alpha}(t) : t \geq 0)$ and $(\mathbf{X}^{B}(t) : t \geq 0)$ be state processes under αF and BF resource allocation. If $\mathbf{X}^{\alpha}(0) \prec_w \mathbf{X}^{B}(0)$ then we have $\mathbf{X}^{\alpha}(t) \prec_w^{st} \mathbf{X}^{B}(t)$ for each $t \geq 0$.*

2. *The mean delays for systems with αF and BF resource allocation for load $\boldsymbol{\rho} \in \mathcal{B}_H$ satisfy the following:*

$$E[D_{\boldsymbol{\rho}}^{\alpha}] \leq E[D_{\boldsymbol{\rho}}^{B}].$$

3.3.2 Dominance Bound

Consider the following resource allocation property. Recall, $\frac{r_i(\mathbf{x})}{x_i}$ is the rate allocated to each job in class i when the system is in state $\mathbf{x}$.

Definition 3 (*Per-job rate monotonicity*). *We say that a resource allocation $\mathbf{r}(.)$ satisfies per-job rate monotonicity if the following holds for all states $\mathbf{x}$*

and $\mathbf{x}'$ such that $\mathbf{x} \geq \mathbf{x}'$: for each class i, we have $\frac{r_i(\mathbf{x})}{x_i} \leq \frac{r_i(\mathbf{x}')}{x_i'}$. In words, adding jobs into the system only decreases the rate allocated to each job.

From the definition of αF, one can check that αF resource allocation satisfies per-job rate monotonicity. This property was used in [9] to provide a comparison result for systems where the resource allocation in one system dominates that in another system for each state $\mathbf{x}$. In contrast, we provide below a comparison result for systems with same resource allocation policy and capacity region, but with different loads. For such systems, we show that the larger loads result into worse delays if the resource allocation satisfies per-job rate monotonicity. For a proof of the theorem below see Section 3.3.4.

Theorem 6. *Consider a system with symmetric polymatroid capacity region $\mathcal{C}$. Suppose that the resource allocation $\mathbf{r}(.)$ satisfies per-job rate monotonicity. Let $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \hat{\mathcal{C}}$ (recall, $\hat{\mathcal{C}}$ is stability region) be such that $\boldsymbol{\rho} \leq \boldsymbol{\rho}'$. Then the following statements hold:*

1. *Let $(\mathbf{X}(t) : t \geq 0)$ and $(\mathbf{X}'(t) : t \geq 0)$ be state processes under loads $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$. If $\mathbf{X}(0) \leq \mathbf{X}'(0)$, then we have $\mathbf{X}(t) \leq^{st} \mathbf{X}'(t)$ for each $t \geq 0$.*

2. *For systems with loads $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$, the mean delays for jobs for each class $i \in F$ satisfy the following:*

$$E[D_i^{(\boldsymbol{\rho})}] \leq E[D_i^{(\boldsymbol{\rho}')}]$$

The above result holds for αF since it satisfies per-job rate monotonicity. However, one can check that the Greedy resource allocation does not satisfy per-job rate monotonicity in general. Thus, we get the following corollary.

Corollary 2. *Consider a system with symmetric polymatroid capacity region and load $\boldsymbol{\rho} \in \mathcal{B}_D$. Let $\rho' = \max_i \rho_i$. Let $\boldsymbol{\rho}'$ be such that for each $i \in F$ we have $\rho'_i = \rho'$ if $\rho_i > 0$ and $\rho'_i = 0$ if $\rho_i = 0$. Then, mean delays for systems with αF resource allocations for load $\boldsymbol{\rho}$ satisfy the following:*

$$E[D^{\alpha}_{\boldsymbol{\rho}}] \leq E[D^{B}_{\boldsymbol{\rho}'}].$$

3.3.3 Majorization Bound

The theorem below generalizes the Dominance bound to provide a mean delay bound for a system with load $\boldsymbol{\rho}$ such that there exists $\boldsymbol{\rho}' \in \mathcal{B}_H$ which satisfies $\boldsymbol{\rho} \prec \boldsymbol{\rho}'$.

Its proof is similar to that of Theorem 5, where instead of relative greediness between resource allocations, we use the following balancing property satisfied by both αF and Greedy: if state $\mathbf{x}$ is more balanced than state $\mathbf{x}'$, then the resource allocation $\mathbf{r}(.)$ would provide larger rates to longer queues in state $\mathbf{x}$ as compared to $\mathbf{x}'$, and thus balancing it even further. For a proof of the theorem below see Section 3.3.4.

Theorem 7. *Consider a system with symmetric polymatroid capacity region $\mathcal{C}$. The resource allocation $\mathbf{r}(.)$ is either αF or Greedy. Let $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \hat{\mathcal{C}}$ be such that $\boldsymbol{\rho} \prec \boldsymbol{\rho}'$ and $\boldsymbol{\rho}' \in \mathcal{B}_H$, i.e., $\boldsymbol{\rho}'$ is homogeneous across classes with non-zero entries. Then the following statements hold:*

1. *Let* $(\mathbf{X}(t) : t \geq 0)$ *and* $(\mathbf{X}'(t) : t \geq 0)$ *be state processes under loads* $\boldsymbol{\rho}$ *and* $\boldsymbol{\rho}'$*. If* $\mathbf{X}(0) \prec_w \mathbf{X}'(0)$*, then we have* $\mathbf{X}(t) \prec_w^{st} \mathbf{X}'(t)$ *for each* $t \geq 0$*.*

2. *The mean delays for systems with loads* $\boldsymbol{\rho}$ *and* $\boldsymbol{\rho}'$ *satisfy the following:*

$$E[D_{\boldsymbol{\rho}}] \leq E[D_{\boldsymbol{\rho}'}]$$

Theorem 7 above is stronger than Theorem 6 in the sense that it only requires the condition $\boldsymbol{\rho} \prec_w \boldsymbol{\rho}'$ instead of $\boldsymbol{\rho} \leq \boldsymbol{\rho}'$. However, it is weaker in the sense that it requires $\boldsymbol{\rho}'$ to be in $\mathcal{B}_H$ and that it gives stochastic weak-majorization of the corresponding state processes instead of stochastic dominance.

For both $\mathbf{r}^G(.)$ and $\mathbf{r}^\alpha(.)$, Theorem 7, along with Theorem 5 and Proposition 1, allows us to bound the mean delay for any load in the following region:

$$\mathcal{B}_M \triangleq \{\boldsymbol{\rho} \in \hat{\mathcal{C}} : \exists \boldsymbol{\rho}' \in \mathcal{B}_H \text{ s.t. } \boldsymbol{\rho} \prec \boldsymbol{\rho}'\},$$

or equivalently,

$$\mathcal{B}_M \triangleq \left\{\boldsymbol{\rho} \in \hat{\mathcal{C}} : \exists k \leq n \text{ s.t. } \max_i \rho_i < \frac{h(k)}{k} \text{ and } |\boldsymbol{\rho}| < h(k)\right\}.$$

Theorem 7 implies that for αF and Greedy resource allocation, the mean delay for each load $\boldsymbol{\rho} \in \mathcal{B}_M$ can be bounded by that for a corresponding load $\boldsymbol{\rho}' \in \mathcal{B}_H$, which in turn has an easily computable bound through Theorem 5. Thus, we get the following corollary.

Corollary 3. *Consider a system with symmetric polymatroid capacity region and load* $\boldsymbol{\rho} \in \mathcal{B}_M$*. Let* $\rho' = \max_{i \in F} \rho_i$*. Let* $k = \min\{l : \rho' \leq \frac{h(l)}{l}$ *and* $|\boldsymbol{\rho}| \leq$

$h(l)\}$. Let A be an arbitrary subset of F of size k and $\boldsymbol{\rho}'$ be such that $\rho'_i = \rho' \ \ \forall i \in A$ and $\rho'_i = 0$ otherwise. Then, the mean delays for systems with Greedy and αF resource allocations for load $\boldsymbol{\rho}$ satisfy the following:

$$E[D^G_{\boldsymbol{\rho}}] \leq E[D^B_{\boldsymbol{\rho}'}], \ \textit{and}\ E[D^\alpha_{\boldsymbol{\rho}}] \leq E[D^B_{\boldsymbol{\rho}'}].$$

It is easy to check that for each $\boldsymbol{\rho} \in \mathcal{B}_M$ the computation of the mean delay upper bound as given by Corollary 3 has complexity $O(n)$ when computed using Corollary 1.

3.3.4 Proofs of Coupling Results

Proof of Theorem 5: Consider the following lemma regarding relative greediness of αF and BF.

Lemma 5. *Consider states $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{x} \prec_w \mathbf{y}$. For each k such that $\sum_{l=1}^k x_{[l]} = \sum_{l=1}^k y_{[l]}$, we have $\sum_{l=1}^k r^\alpha_{(l)}(\mathbf{x}) \geq \sum_{l=1}^k r^B_{(l)}(\mathbf{y})$.*

Roughly, it asserts that if state $\mathbf{x}$ is same or more balanced than state $\mathbf{y}$, then the sum rate assigned to larger queues by αF to state $\mathbf{x}$ is greater than that by BF to state $\mathbf{y}$. Proof of this lemma is given in Section 3.6.2. Below, we provide a detailed coupling argument showing stochastic weak-majorization using this lemma.

Coupling Argument: Without loss of generality, assume $\nu = 1$. Suppose $\mathbf{X}^\alpha(0) \prec_w \mathbf{X}^B(0)$. Below, we couple the arrivals and departures of processes $(\mathbf{X}^\alpha(t) : t \geq 0)$ and $(\mathbf{X}^B(t) : t \geq 0)$ such that their marginal distributions remain intact and $\mathbf{X}^\alpha(t) \prec_w \mathbf{X}^B(t)$ almost surely for each $t \geq 0$.

Let Π_a be a Poisson point process with rate $\sum_{i\in F}\lambda_i$, and let Π_d be Poisson point process with rate $\mu(F)$. The points in these processes are the times of 'potential events' in $(\mathbf{X}^B(t) : t \geq 0)$ and $(\mathbf{X}^\alpha(t) : t \geq 0)$. We use Π_a to couple arrivals and Π_d to couple departures. For each time t' when a potential event occurs, let $\epsilon_{t'}$ be a small enough number such that no potential event occurred in the time interval of $[t' - \epsilon_{t'}, t')$.

Coupling of arrivals: For each point t' in Π_a, do the following: Choose a random variable $Z_{t'}$ independently and uniformly from $\{1, \ldots, n\}$. Let an arrival occur in $(\mathbf{X}^\alpha(t) : t \geq 0)$ at time t' in the $Z_{t'}^{\text{th}}$ largest queue of $\mathbf{X}^\alpha(t' - \epsilon_{t'})$. Ties are broken uniformly at random. Similarly, let an arrival occur in $(\mathbf{X}^\alpha(t) : t \geq 0)$ at time t' in the $Z_{t'}^{\text{th}}$ largest queue of $\mathbf{X}^\alpha(t' - \epsilon_{t'})$. Again, ties are broken uniformly at random.

Coupling of departures: For each point t' of increment in Π_d, do the following: Choose a random variable $Z_{t'}$ independently and uniformly from interval $(0, \mu(F)]$. For k such that

$$Z_{t'} \in \left(\sum_{l=1}^{k-1} r^\alpha_{(l)}(X^\alpha(t' - \epsilon_{t'})), \sum_{l=1}^{k} r^\alpha_{(l)}(X^\alpha(t' - \epsilon_{t'})) \right],$$

let a departure occur in $(\mathbf{X}^\alpha(t) : t \geq 0)$ at time t' in the k^{th} largest queue of $\mathbf{X}^\alpha(t' - \epsilon_{t'})$, with ties broken uniformly and independently at random.

Similarly, for k such that

$$Z_{t'} \in \left(\sum_{l=1}^{k-1} r^B_{(l)}(\mathbf{X}^B(t' - \epsilon_{t'})), \sum_{l=1}^{k} r^B_{(l)}(\mathbf{X}^B(t' - \epsilon_{t'})) \right],$$

let a departure occur in $(\mathbf{X}^B(t) : t \geq 0)$ at time t' in the k^{th} largest queue of $\mathbf{X}^B(t' - \epsilon_{t'})$, with ties broken uniformly and independently at random. Note that in both cases it is possible that no such k exists since some classes may not be active and the total service rate may be less than $\mu(F)$. In that case, no departure occurs.

It can be checked that the marginal distributions of $(\mathbf{X}^\alpha(t) : t \geq 0)$ and $(\mathbf{X}^B(t) : t \geq 0)$ remain intact. We now show that $\mathbf{X}^\alpha(t) \prec_w \mathbf{X}^B(t)$ almost surely for each t.

It is easy to check that if an arrival occurred at time t' and if $\mathbf{X}^\alpha(t) \prec_w \mathbf{X}^B(t)$ for each $t < t'$, then $\mathbf{X}^\alpha(t') \prec_w \mathbf{X}^B(t')$ as well. We now show that the same holds for points of Π_d as well.

Suppose a potential departure occurred at t', and $\mathbf{X}^\alpha(t) \prec_w \mathbf{X}^B(t)$ for each $t < t'$. We show below that $\sum_{l=1}^k X_{[l]}^\alpha(t') \leq \sum_{l=1}^k X_{[l]}^B(t')$ for each k. Here, we use Lemma 5. Following two cases arise.

Case 1: $\sum_{l=1}^k X_{[l]}^\alpha(t' - \epsilon_{t'}) < \sum_{l=1}^k X_{[l]}^B(t' - \epsilon_{t'})$. Since a maximum of one departure occurs at time t' in either processes, we have $\sum_{l=1}^k X_{[l]}^\alpha(t') \leq \sum_{l=1}^k X_{[l]}^B(t')$.

Case 2: $\sum_{l=1}^k X_{[l]}^\alpha(t' - \epsilon_{t'}) = \sum_{l=1}^k X_{[l]}^B(t' - \epsilon_{t'})$. By using $\mathbf{X}^\alpha(t - \epsilon_{t'}) \prec_w \mathbf{X}^B(t - \epsilon_{t'})$ in Lemma 5 and from the definition of the coupling at time t', it can be shown that if a departure occurs from any of the k largest queues in $\mathbf{X}^B(t' - \epsilon_{t'})$, then it also occurs in one of the k largest queues in $\mathbf{X}^\alpha(t' - \epsilon_{t'})$. Thus, $\sum_{l=1}^k X_{[l]}^\alpha(t') \leq \sum_{l=1}^k X_{[l]}^B(t')$.

Hence the first part of the theorem follows. Second part follows by application of Little's law on $(|\mathbf{X}^{\alpha}(t)| : t \geq 0)$ and $(|\mathbf{X}^{B}(t)| : t \geq 0)$. □

Proof of Theorem 6: Suppose $\mathbf{X}(0) \leq \mathbf{X}'(0)$. Below, we couple the arrivals and departures of jobs in $(\mathbf{X}(t) : t \geq 0)$ and $(\mathbf{X}'(t) : t \geq 0)$ such that their marginal distributions remain intact and $\mathbf{X}(t) \leq \mathbf{X}'(t)$ almost surely for each $t \geq 0$.

Since mean service requirement of jobs ν is same for both the systems, the corresponding arrival rates satisfy $\boldsymbol{\lambda} \leq \boldsymbol{\lambda}'$. For each i let Π_i and Π_i' be the Poisson arrival processes for class i in the respective systems. Let Π_i be obtained by sampling Π_i'. For each class i, the arrivals in $(\mathbf{X}'(t) : t \geq 0)$ at the sampled points, i.e., points in Π_i, see the average delay which is equal to the overall average delay of jobs in Π_i' for this system. Thus, the theorem follows if we couple the departures of jobs in both the systems such that for each point in Π_i, the corresponding job departure in $(\mathbf{X}(t) : t \geq 0)$ is no later than that in $(\mathbf{X}'(t) : t \geq 0)$. By using per-flow rate monotonicity property, one can couple the service rate of these jobs at each time t so that if such a job departs from $(\mathbf{X}'(t) : t \geq 0)$ than the corresponding job departs from $(\mathbf{X}(t) : t \geq 0)$ as well, if it hasn't already. □

Proof of Theorem 7: The theorem can be proved in a fashion similar to that of Theorem 5, except for the following changes. For notational convenience, for each time t let $\lambda_{(k)}(t)$ and $\lambda'_{(k)}(t)$ be the arrival rates of k^{th} largest queues in $\mathbf{X}(t)$ and $\mathbf{X}'(t)$ respectively, with ties broken arbitrarily.

1. *Coupling of arrivals:* For each point t' in Π_a, we choose a random variable $Z_{t'}$ independently and uniformly from interval $(0, |\boldsymbol{\lambda}|]$. For each k such that
$$Z_{t'} \in \left(\sum_{l=1}^{k-1} \lambda_{(l)}(t' - \epsilon_{t'}), \sum_{l=1}^{k} \lambda_{(l)}(t' - \epsilon_{t'}) \right],$$
let an arrival occur in $(\mathbf{X}(t) : t \geq 0)$ at time t' in the k^{th} largest queue of $\mathbf{X}(t' - \epsilon_{t'})$. Similarly, for each k such that
$$Z_{t'} \in \left(\sum_{l=1}^{k-1} \lambda'_{(l)}(t' - \epsilon_{t'}), \sum_{l=1}^{k} \lambda'_{(l)}(t' - \epsilon_{t'}) \right],$$
let an arrival occur in $(\mathbf{X}'(t) : t \geq 0)$ at time t' in the k^{th} largest queue of $\mathbf{X}'(t' - \epsilon_{t'})$.

2. *Coupling of departures:* Similar to that of Theorem 5, except that instead of Lemma 5 for a proof of weak-majorization upon a potential departure, we use the following lemma which asserts that αF and Greedy provide larger rate to longer queues in more balanced states.

Lemma 6. *Consider states* $\mathbf{x}$ *and* $\mathbf{y}$ *such that* $\mathbf{x} \prec_w \mathbf{y}$. *For each* k *such that* $\sum_{l=1}^{k} x_{[l]} = \sum_{l=1}^{k} y_{[l]}$, *we have* $\sum_{l=1}^{k} r^{\alpha}_{(l)}(\mathbf{x}) \geq \sum_{l=1}^{k} r^{\alpha}_{(l)}(\mathbf{y})$, *and* $\sum_{l=1}^{k} r^{G}_{(l)}(\mathbf{x}) \geq \sum_{l=1}^{k} r^{G}_{(l)}(\mathbf{y})$.

For $\mathbf{r}^G(.)$, is easy to check that the lemma holds. For $\mathbf{r}^{\alpha}(.)$, it follows from Lemma 15 in Section 3.6.2.

Hence the result. □

3.4 Asymptotic symmetry in large systems

Large content delivery systems, where servers can jointly serve file-download requests, not only have polymatroid capacity but under appropriate assumptions become approximately symmetric.

Consider a sequence of bipartite graphs $G^{(n)} = (F^{(n)} \cup S^{(n)}; E^{(n)})$ where $F^{(n)}$ is a set of n files, $S^{(n)}$ is a set of $m = \lceil bn \rceil$ servers for some constant b, and each edge $e \in E^{(n)}$ connecting a file $i \in F^{(n)}$ and server $s \in S^{(n)}$ implies that a copy of file i is available at server s. For each node $s \in S^{(n)}$, let $N_s^{(n)}$ denote the set of neighbors of server s, i.e., the set of files it stores and can serve. Henceforth, wherever possible, we will avoid the use of ceil and floor notations to avoid clutter.

We associate each file in $F^{(n)}$ with a class of job arrivals each corresponding to a file download request. The arrival processes and service requirements are as described in Section 2.2, with $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\rho}^{(n)}$ representing the corresponding arrival rates and loads. Further, we let the service capacity of each server $s \in S^{(n)}$ be μ_s bits per second.

We allow each server $s \in S^{(n)}$ to concurrently serve the jobs with classes $N_s^{(n)}$ as long as the total service rate does not exceed μ_s. The service rate for each job is the sum of the rates it receives from different servers. For any $A \subset F^{(n)}$, let $\mu^{(n)}(A)$ be the maximum sum rate at which jobs with file-class in A could be served, i.e.,

$$\mu^{(n)}(A) = \sum_{s \in S^{(n)}} \mathbf{1}_{\left\{A \cap N_s^{(n)} \neq \emptyset\right\}} \mu_s.$$

Clearly any rate allocation $\mathbf{r}(.)$ for such a system must satisfy the following constraints for each state $\mathbf{x}$: $\forall A \subset F^{(n)}$,

$$\sum_{i \in A} r_i(\mathbf{x}) \leq \mu^{(n)}(A).$$

We showed in Section 2.2 that $\mu^{(n)}(.)$ is submodular and that the corresponding polymatroid

$$\mathcal{C}^{(n)} = \left\{ \mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq \mu^{(n)}(A),\ \forall A \subset F^{(n)} \right\}$$

is indeed the capacity region for such a system, i.e., each $\mathbf{r} \in \mathcal{C}^{(n)}$ is achievable.

Note that $\mathcal{C}^{(n)}$ will in general be an asymmetric polymatroid depending upon edges $E^{(n)}$ and service capacities μ_s for each $s \in S^{(n)}$. However, we show below that if copies of files are stored across servers at random and scaled appropriately with n then, as n increases, a uniform law of large numbers hold where $\mathcal{C}^{(n)}$ gets uniformly close to a symmetric polymatroid, subject to the following assumptions:

Assumption 2 (Heterogeneous server capacities)**.** *$S^{(n)}$ is partitioned into a finite number of groups where each group has $\Omega(n)$ number of servers. Within each group, the server capacities are homogeneous. The server capacities across groups may be heterogeneous such that average of service capacity across servers*

$$\xi \triangleq \frac{1}{m} \sum_{s \in S^{(n)}} \mu_s$$

is independent of n.

Assumption 3 (<u>Randomized file placement</u>)**.** *Let* $(c_n : n \in \mathbb{N})$ *be a sequence such that*

$$c_n = \omega(\log n).$$

For each file $i \in F^{(n)}$*, store a copy in* c_n *different servers chosen uniformly and independently at random.*

A randomized placement of file copies implies a random system configuration, i.e., a random graph. Let $\mathcal{E}^{(n)}$ denote the random set of edges resulting Assumption 3. Similarly, for each $s \in S^{(n)}$, let $\mathcal{N}_s^{(n)}$ denote the random set of neighbors of s, i.e., the random set of files stored in server s. Let $M^{(n)}(.)$ denote the corresponding random rank function, and $\mu^{(n)}(.)$ a possible realization. Then, for each $A \subset F^{(n)}$, we have

$$M^{(n)}(A) = \sum_{s \in S^{(n)}} \mathbf{1}_{\left\{A \cap \mathcal{N}_s^{(n)} \neq \emptyset\right\}} \mu_s,$$

where $\mathbf{1}_{\left\{A \cap \mathcal{N}_s^{(n)} \neq \emptyset\right\}}$ is now a Bernoulli random variable indicating if a copy of at least one of the files in A is placed in s. In fact, for each $A \subset F^{(n)}$ such that $|A| = k$, the set $\left\{\mathbf{1}_{\left\{A \cap \mathcal{N}_s^{(n)} \neq \emptyset\right\}} : s \in S^{(n)}\right\}$ is a set of m negatively associated Bernoulli($p_k^{(n)}$) random variables [20] where $p_k^{(n)}$ is the probability that a given server is assigned at least one of the kc_n copies of files in A and is given by

$$p_k^{(n)} \triangleq 1 - \left(1 - \frac{1}{m}\right)^{kc_n} \quad \forall k = 0, 1, \dots, n.$$

By linearity of expectation, for each $A \subset F^{(n)}$, we have

$$\bar{\mu}^{(n)}(A) \triangleq E[M^{(n)}(A)] = \xi m p_{|A|}^{(n)}.$$

Note, $\bar{\mu}^{(n)}(A)$ depends on A only through $|A|$ and is thus symmetric. The theorem below shows that with high probability we can bound the random rank function $M^{(n)}(.)$ uniformly over all $A \subset F^{(n)}$, from above as well as from below, with a symmetric rank function which is close to $\bar{\mu}^{(n)}(A)$. See Section 3.4.1 for a proof.

Theorem 8. *Fix ϵ independent of n such that $0 < \epsilon < 1$. Consider a sequence of systems with n files and $m = \lceil bn \rceil$ servers, where $b > 0$ is a constant. Under Assumptions 2 and 3, let $M^{(n)}(.)$ be the corresponding random rank function. Then, there exists a sequence $(g_n : n \in \mathbb{N})$ such that $g_n = \omega(\log n)$, and*

$$P\left(\exists A \subset F^{(n)} \ s.t. \ M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-g_n},$$

and

$$P\left(\exists A \subset F^{(n)} \ s.t. \ M^{(n)}(A) \geq (1+\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-g_n}.$$

This result gives us following corollary on the random capacity region associated with $M^{(n)}(.)$ generated by random file placement. Recall, $\bar{\mu}^{(n)}(A) = E[M^{(n)}(A)]$ for all $A \subset F^{(n)}$, and let

$$\bar{\mathcal{C}}^{(n)} \triangleq \left\{ \mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq \bar{\mu}^{(n)}(A), \ \forall A \subset F^{(n)} \right\}.$$

Thus $\bar{\mathcal{C}}^{(n)}$ is the (symmetric) capacity region associated with the average rank function $\bar{\mu}(.)$. Then, the following holds:

Corollary 4. *Fix ϵ independent of n such that $0 < \epsilon < 1$. Under Assumptions 2 and 3, the random capacity region associated with randomized file placement is a subset of $(1+\epsilon)\bar{\mathcal{C}}^{(n)}$ and a superset of $(1-\epsilon)\bar{\mathcal{C}}^{(n)}$ with high probability.*

Further, under Assumption 2, there exists a deterministic file placement where $c_n = \omega(\log n)$ copies of each file are stored across servers such that the corresponding capacity region $\mathcal{C}^{(n)}$ is a subset of $(1+\epsilon)\bar{\mathcal{C}}^{(n)}$ and a superset of $(1-\epsilon)\bar{\mathcal{C}}^{(n)}$.

3.4.1 Proof of Theorem 8

Here, we will only show

$$P\left(\exists A \subset F^{(n)} \text{ s.t. } M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-g_n},$$

The other bound follows in similar fashion.

For now, suppose $\mu_s = \xi$ for each $s \in S^{(n)}$. We relax this assumption later.

We first provide a bound for $P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right)$ for each $A \subset F^{(n)}$. Then, for each $k = 1, 2, \ldots, n$, we use union bound to obtain a uniform bound over all sets $A \subset F^{(n)}$ such that $|A| = k$. The bound we provide for $P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right)$ is small enough so that the above union bound is small too. Then, yet another use of the union bound would give us the uniform result over all sets $A \subset F^{(n)}$.

Now, if the random variables $\left\{\mathbf{1}_{\left\{A \cap \mathcal{N}_s^{(n)} \neq \emptyset\right\}} : s \in S^{(n)}\right\}$ were independent Bernoulli($p_k^{(n)}$), then the following two concentration results would hold [42]: Fix $k \in \{1, \ldots, n\}$. For each set $A \subset F^{(n)}$ such that $|A| = k$, we have

$$P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-\frac{\epsilon^2}{2} m p_k^{(n)}}, \tag{3.4}$$

and,

$$P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-mH\left(p_k^{(n)}(1-\epsilon)||p_k^{(n)}\right)}, \tag{3.5}$$

where $H(p||q)$ is the KL divergence between Bernoulli(p) and Bernoulli(q) random variables, given by

$$H(p||q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right).$$

However, in reality, since $\left\{\mathbf{1}_{\left\{A\cap\mathcal{N}_s^{(n)}\neq\emptyset\right\}} : s \in S^{(n)}\right\}$ are negatively associated Bernoulli($p_k^{(n)}$) random variables, the above Chernoff bounds still apply [20].

In the sequel, we will use the following two technical lemmas. Their proofs are provided in the Section 3.6.4.

Lemma 7. *Let a sequence* $(g_n : n \in \mathbb{N})$ *be such that* $g_n = o(c_n)$*. Let* δ_1 *be a positive constant independent of* n *such that* $\delta_1 < 1$*. Then, for large enough* n*, we have*

$$p_k^{(n)} \geq \frac{\delta_1 g_n}{n}k \quad \forall k \in \left\{0, 1, \ldots, \left\lfloor\frac{n}{g_n}\right\rfloor\right\}.$$

Lemma 8. *There exists a positive constant* δ *such that* $H\left(p_k^{(n)}(1-\epsilon)||p_k^{(n)}\right) \geq -\delta + \epsilon\frac{kc_n}{m}$.

Now, let $(g_n : n \in \mathbb{N})$ be a sequence such that $g_n \triangleq (c_n \log n)^{1/2}$ for each n. The following properties of g_n can be easily checked:

$$g_n = \omega(\log n) \text{ and } g_n = o(c_n). \tag{3.6}$$

We now provide a uniform bound over all sets $A \subset F^{(n)}$ such that $|A| = k$ for each $k \in \{1, \ldots, n\}$, under following two cases.

<u>*Case 1:*</u> $0 \leq k \leq \frac{n}{g_n}$: From Lemma 7, for each k we have

$$p_k^{(n)} \geq \delta_1 \frac{k g_n}{n},$$

for a suitably chosen positive constant δ_1 independent of n. In the sequel, δ_i for any $i \geq 1$ will be a suitably chosen positive constant independent of n.

Using the concentration result (3.4), for $|A| = k$ we get

$$P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-\frac{\epsilon^2}{2}\delta_1 b k g_n},$$

and using the union bound, we get

$$\begin{aligned} P\Big(\exists A \subset F^{(n)} \text{ s.t. } |A| = k \text{ and } M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\Big) \\ \leq e^{-\frac{\epsilon^2}{2}\delta_1 b k g_n}\binom{n}{k} \leq e^{-\frac{\epsilon^2}{2}\delta_1 b k g_n + k\log n} \leq e^{-\delta_2 k g_n}. \end{aligned}$$

<u>*Case 2:*</u> $\frac{n}{g_n} < k \leq n$: In this case, we use the concentration result (3.5). From Lemma 8, we get

$$P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{(\delta_6 m - \epsilon k c_n)}.$$

Since $g_n = o(c_n)$, for n large enough we get $\delta_6 m \leq (\epsilon/2)\frac{n c_n}{g_n}$. Also, for each $k > \frac{n}{g_n}$, we have $(\epsilon/2)\frac{n c_n}{g_n} \leq (\epsilon/2) k c_n$. Thus, for large enough n, $\delta_6 m - \epsilon k c_n \leq -(\epsilon/2) k c_n$ for each k such that $\frac{n}{g_n} < k \leq n$, and consequently,

$$P\left(M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-\delta_7 k c_n}$$

By using the union bound, for large enough n, we get

$$P\left(\exists A \subset F^{(n)} \text{ s.t. } |A| = k \text{ and } M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \\ \leq e^{-\delta_7 k c_n} \binom{n}{k} \leq e^{-\delta_7 k c_n + k \log n} \leq e^{-\delta_8 k c_n}.$$

Combining the above two cases, we can show that for large enough n there exists a positive constant δ_9 such that for each $k \in \{1, \ldots, n\}$ we have

$$P\left(\exists A \subset F^{(n)} \text{ s.t. } |A| = k \text{ and } M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq e^{-\delta_9 g_n}.$$

Using the union bound again, we get

$$P\left(\exists A \subset F^{(n)} \text{ s.t. } M^{(n)}(A) \leq (1-\epsilon)\bar{\mu}^{(n)}(A)\right) \leq n e^{-\delta_9 g_n} \leq e^{-\delta_9 g_n + \log n} \leq e^{-\delta_{10} g_n}.$$

Now, we relax the assumption $\mu_s = \xi$ for each $s \in S^{(n)}$ with Assumption 2. The above proof can then be used to show a similar concentration result for individual groups. The overall result follows by linearity of expectation and yet another use of the union bound. □

3.5 Delay scaling and robustness

We now combine results from Section 3.3 and Section 3.4 to exhibit performance robustness in large content delivery systems. In Section 3.4, we showed that large content delivery systems support symmetric polymatroid capacity regions. This allows us to apply the performance bounds developed in Section 3.3 for symmetric polymatroid capacity regions.

However, there is one more hurdle to overcome before we can apply our bounds from Section 3.3. Recall, from Corollary 4, under Assumptions 2 and 3 the random capacity region for content delivery systems *contain* and are *contained by* symmetric polymatroids with high probability. The realizations of the random capacity region, themselves, may still not be symmetric. We thus need to show that if the capacity region is bigger then the corresponding mean delay is smaller when subject to the same load.

Intuitively, larger capacity regions may imply larger service rates for each class, and may thus provide better performance. Although intuitively obvious, such results are not always straightforward. We show below that such a comparison result indeed holds under certain monotonicity conditions for rate allocations. Consider the following monotonicity condition.

Definition 4 (*Monotonicity w.r.t. capacity region*)**.** *We say that a rate allocation satisfies monotonicity w.r.t. capacity region if for any state* $\mathbf{x}$*, the rate allocation per class for a system with a larger capacity region dominates that with a smaller one.*

Further, recall per-job rate monotonicity defined in Section 3.3.2, where the rate allocated to each job (viz., $\frac{r_i(\mathbf{x})}{x_i}$ for jobs in class i) only decreases when an additional job is added into the system. The following lemma can be shown to hold through a simple coupling argument across jobs for arbitrary polymatroid capacity regions.

Lemma 9. *Consider systems with arbitrary polymatroid capacity regions* $\mathcal{C}$ *and* $\tilde{\mathcal{C}}$ *such that* $\mathcal{C} \subset \tilde{\mathcal{C}}$*. Consider a rate allocation which satisfies monotonicity* *w.r.t. capacity region as well as per-job rate monotonicity. Then, the mean delay for capacity region $\mathcal{C}$ under arbitrary load $\boldsymbol{\rho}$ upper bounds that for capacity region $\tilde{\mathcal{C}}$ under the same load.*

It is easy to check that α-fair rate allocation satisfies per-job rate monotonicity as well as monotonicity w.r.t. capacity region. Thus, Lemma 9 holds for α-fair rate allocation. However, one can show that Greedy rate allocation may not satisfy either property for arbitrary polymatroid capacity regions. This further highlights the brittleness of Greedy rate allocation to asymmetries. Even for Balanced fair rate allocation it is not directly clear if the lemma holds. Thus, henceforth we will only consider α-fair rate allocation.

Now we are indeed ready with all the tools required to exhibit robustness in large scale systems.

Assumption 4 (Load Heterogeneity)**.** *We consider a sequence of systems where load $\boldsymbol{\rho}^{(n)}$ for each n is allowed to be within a set $\mathcal{B}^{(n)}$ defined as follows: Consider a sequence $(\theta_n : n \in \mathbb{N})$ such that $\theta_n = \omega(1)$, $\theta_n = o(\frac{n}{\log n})$, and $\theta_n = o(c_n)$. Also, fix a constant $\gamma < 1$ independent of n. For each n:*

$$\mathcal{B}^{(n)} \triangleq \left\{ \boldsymbol{\rho} : \max_{i \in F^{(n)}} \rho_i \leq \theta_n \textit{ and } |\boldsymbol{\rho}| \leq \gamma \xi m \right\}.$$

The condition $|\boldsymbol{\rho}| \leq \gamma\xi m$ implies that we allow load to increase linearly with system size. Also, since $\theta_n = \omega(1)$, the condition $\max_i \rho_i \leq \theta_n$ implies that we allow load across servers to be increasingly heterogeneous. The condition $\theta_n = o(\frac{n}{\log n})$ limits the heterogeneity allowed in the system. Further, the condition $\theta_n = o(c_n)$ would allow us to claim stability, and to show that the mean delay of the system tends to 0 as n increases.

The following is the main result of this section.

Theorem 9. *Consider a sequence of systems with n files $F^{(n)}$ and $m = \lceil bn \rceil$ servers $S^{(n)}$, where b is a constant. For each n, let the total service capacity of servers be ξm, where ξ is independent of n. $S^{(n)}$ is partitioned into a finite number of heterogeneous groups, each with $\Omega(n)$ servers and equal per-server capacity. Let $(c_n : n \in \mathbb{N})$ be a sequence such that $c_n = \omega(\log n)$. We allow c_n copies of each file to be placed across the servers.*

Let the service requirement of jobs be exponentially distributed with mean ν, where ν is independent of n. Let $(\theta_n : n \in \mathbb{N})$ be a sequence such that $\theta_n = \omega(1)$, but $o\left(\min(\frac{n}{\log n}, c_n)\right)$. Fix a constant $\gamma < 1$. Let $\mathcal{B}^{(n)} = \{\boldsymbol{\rho} : \max_i \rho_i \leq \theta_n \text{ and } |\boldsymbol{\rho}| \leq \gamma \xi m\}$. For each n, let load across file classes be $\boldsymbol{\rho}^{(n)} \in \mathcal{B}^{(n)}$.

Fix a constant $\delta > 1$. Then, there exists an integer n_δ such that for each $n \geq n_\delta$ the following holds: there exists at least one file placement policy such that the mean delay for file-download jobs with α-fair rate allocation satisfies the following bound:

$$E[D^{(n)}] \leq \delta \frac{\nu \theta_n}{\xi c_n} \frac{1}{\gamma} \log \left(\frac{1}{1-\gamma} \right).$$

Further, for each $n \geq n_\delta$, if the c_n copies of each file are stored uniformly at random across servers, then the above bound holds with high probability.

3.5.1 Proof of Theorem 9

We first show the existence of a file placement policy such that the mean delay bound is satisfied. Without loss of generality, assume $\delta < \frac{1}{\gamma}$.

From Corollary 4, and definitions of $\bar{\mathcal{C}}^{(n)}$ and $\bar{\mu}^{(n)}(.)$, for large enough n there exists a file placement such that the corresponding capacity region contains the following symmetric polymatroid:

$$\tilde{\mathcal{C}}^{(n)} \triangleq \left\{ \mathbf{r} \geq \mathbf{0} : \sum_{i \in A} r_i \leq h^{(n)}(|A|),\ \forall A \subset F^{(n)} \right\},$$

where

$$h^{(n)}(k) \triangleq (1/\delta)\xi m \left(1 - e^{-\frac{kc_n}{m}}\right) \quad \forall k = 0, 1, \dots, n.$$

Thus, from Lemma 9, for α-fair rate allocations it is sufficient to consider $\tilde{\mathcal{C}}^{(n)}$. Further, since $\tilde{\mathcal{C}}^{(n)}$ is monotonic in c_n, it is sufficient to assume that $c_n = o(\frac{n}{\log n})$ since, if it is not, we can set c_n to be equal to $\sqrt{\frac{n}{\log n}\theta_n}$ and all the assumptions still hold. Thus, henceforth we assume that

$$c_n = o(\frac{n}{\log n}).$$

Let $\xi' \triangleq \xi/\delta$. Thus, we get

$$h^{(n)}(k) = \xi' m \left(1 - e^{-\frac{kc_n}{m}}\right) \quad \forall k = 0, 1, \dots, n.$$

Since $\gamma\xi m < \xi' m$ and $\theta_n = o(c_n)$, one can check that $B^{(n)}$ is a subset of $\tilde{\mathcal{C}}^{(n)}$ for large enough n, and we get stability.

Let $t_n \triangleq \left\lceil \frac{\gamma\xi' m}{\theta_n} \right\rceil$. Let $A^{(n)}$ be an arbitrary subset of $F^{(n)}$ such that $|A^{(n)}| = t_n$. Let $\hat{\boldsymbol{\rho}}^{(n)} = (\hat{\rho}_i^{(n)} : i \in F^{(n)})$ where $\hat{\rho}_i^{(n)} = \theta_n$ if $i \in A^{(n)}$ and 0 otherwise. Then, it is easy to show that for each n, we have

$$B^{(n)} \subset \left\{ \boldsymbol{\rho} : \boldsymbol{\rho} \prec_w \hat{\boldsymbol{\rho}}^{(n)} \right\}.$$

Thus, from Theorem 7, it is sufficient to show that the bound on mean delay holds for balanced fair rate allocation under load $\boldsymbol{\rho}^{(n)} = \hat{\boldsymbol{\rho}}^{(n)}$.

Henceforth, we assume BF rate allocation and let load $\boldsymbol{\rho}^{(n)} = \hat{\boldsymbol{\rho}}^{(n)}$. For each n, we invoke Corollary 1 with ρ replaced by θ_n and n replaced by t_n, where for each $k = 0, 1, \ldots, t_n$ we let[1]

$$\pi_k^{(n)} \triangleq \frac{F_k(\theta_n)}{F(\theta_n)}, \text{ and } \tau_k^{(n)} \triangleq \frac{\hat{F}_k(\theta_n)}{F(\theta_n)}.$$

Then, we have

$$E[D^{(n)}] = \nu \sum_{k=1}^{t_n} \frac{k}{t_n} \tau_k^{(n)}. \tag{3.7}$$

Also, we have $\tau_0^{(n)} = 0$, $\pi_0^{(n)} = 1/F(\theta_n)$, and for each $k = 1, \ldots, t_n$ we have

$$\pi_k^{(n)} = \frac{(t_n - k + 1)\theta_n}{h^{(n)}(k) - k\theta_n} \pi_{k-1}^{(n)}, \tag{3.8}$$

and

$$\tau_k^{(n)} = \frac{\pi_k^{(n)} + \frac{t_n - k + 1}{k} \pi_{k-1}^{(n)} + \frac{(t_n - k + 1)(k-1)}{k} \theta_n \tau_{k-1}^{(n)}}{h^{(n)}(k) - k\theta_n}. \tag{3.9}$$

First, we show the following result.

[1] If $\pi^{(n)}(\mathbf{x})$ stationary distribution of the queue length process for the n^{th} system, then $\pi_k^{(n)}$ has the following interpretation: $\pi_k^{(n)} = \sum_{\mathbf{x}:|A_{\mathbf{x}}|=k} \pi^{(n)}(\mathbf{x})$ for $k = 1, \ldots, t_n$.

Theorem 10. *For any positive constants $\epsilon > 1$ and $\epsilon' < 1$ independent of n, there exists a constant $\delta' < 1$ such that for large enough n we have*

$$\sum_{k=\epsilon' b \log(\frac{1}{1-\gamma})\frac{n}{c_n}}^{\epsilon b \log(\frac{1}{1-\gamma})\frac{n}{c_n}} \pi_k^{(n)} \geq 1 - \delta'^{\frac{m}{c_n}}. \tag{3.10}$$

Proof. Fix a constant δ_{11} independent of n such that $0 < \delta_{11} < 1$. Let

$$k_\downarrow^{(n)} = \frac{m}{c_n} \log\left(\frac{1}{1-\gamma\delta_{11}}\right).$$

Then, we have $h^{(n)}(k_\downarrow) = \gamma\delta_{11}\xi' m$. In fact, we have $h^{(n)}(k) \leq \gamma\delta_{11}\xi' m, \;\; \forall k \leq k_\downarrow^{(n)}$. Using (3.8), for each $k \leq k_\downarrow^{(n)}$, we have

$$\pi_k^{(n)} \geq \frac{(t_n - k + 1)\theta_n}{\gamma\delta_{11}\xi' m - k\theta_n}\pi_{k-1}^{(n)} \geq \frac{t_n\theta_n - (k_\downarrow^{(n)} - 1)\theta_n}{\gamma\delta_{11}\xi' m}\pi_{k-1}^{(n)} = \frac{\gamma\xi' m - o(n)}{\gamma\delta_{11}\xi' m}\pi_{k-1}^{(n)} \geq \frac{1}{\delta_{12}}\pi_{k-1}^{(n)},$$

for a positive constant δ_{12} such that $\delta_{11} < \delta_{12} < 1$, and large enough n. Equivalently, $\pi_k^{(n)} \leq \delta_{12}\pi_{k+1}^{(n)} \;\; \forall k < k_\downarrow^{(n)}$. Fix a positive constant $\epsilon_1 < 1$. Then, for all $k < \epsilon_1 k_\downarrow^{(n)}$, we have

$$\pi_k^{(n)} \leq \delta_{12}^{(1-\epsilon_1)k_\downarrow^{(n)}} \pi_{k_\downarrow^{(n)}}^{(n)}.$$

Now, fix a constant δ_{13} independent of n such that $\gamma < \delta_{13} < 1$ and let

$$k_\uparrow^{(n)} = \frac{m}{c_n} \log\left(\frac{1}{1-\gamma/\delta_{13}}\right).$$

Then, for all $k \geq k_\uparrow^{(n)}$, we have $h^{(n)}(k) \geq \gamma\xi' m/\delta_{13}$. Now, for large enough n, $\gamma\xi' m/\delta_{13} \geq \gamma\xi' m + \theta_n \geq (t_n + 1)\theta_n$. Thus, for large enough n, we have $h^{(n)}(k) - k\theta_n \geq (t_n - k + 1)\theta_n \;\; \forall k \geq k_\uparrow^{(n)}$, or equivalently from (3.8),

$$\pi_k^{(n)} \leq \pi_{k-1}^{(n)} \;\; \forall k \geq k_\uparrow^{(n)}. \tag{3.11}$$

In fact, for a fixed positive constant $\epsilon_2 > 1$, for all k such that $k_{\uparrow}^{(n)} \leq k \leq \epsilon_2 k_{\uparrow}^{(n)}$ we have

$$\pi_k^{(n)} \leq \frac{(t_n - k + 1)\theta_n}{\gamma\xi' m/\delta_{13} - k\theta_n}\pi_{k-1}^{(n)} \leq \frac{t_n\theta_n}{\gamma\xi' m/\delta_{13} - \epsilon_2 k_{\uparrow}^{(n)}\theta_n}\pi_{k-1}^{(n)} \leq \frac{\gamma\xi' m}{\gamma\xi' m/\delta_{13} - o(n)}\pi_{k-1}^{(n)}$$
$$\leq \delta_{14}\pi_{k-1}^{(n)},$$

for a positive constant δ_{14} such that $\delta_{13} < \delta_{14} < 1$, and large enough n. Thus, $\pi_{\epsilon_2 k_{\uparrow}^{(n)}}^{(n)} \leq \delta_{14}^{(\epsilon_2 - 1)k_{\uparrow}^{(n)}} \pi_{k_{\uparrow}^{(n)}}^{(n)}$ for large enough n. Further, using (3.11) we get

$$\pi_k^{(n)} \leq \delta_{14}^{(\epsilon_2 - 1)k_{\uparrow}^{(n)}} \pi_{k_{\uparrow}^{(n)}}^{(n)} \quad \forall k > \epsilon_2 k_{\uparrow}^{(n)}$$

Thus, we get

$$\begin{aligned}
1 = \sum_{k=0}^{t_n} \pi_k^{(n)} &= \sum_{k=0}^{\epsilon_1 k_{\downarrow}^{(n)} - 1} \pi_k + \sum_{k=\epsilon_1 k_{\downarrow}^{(n)}}^{\epsilon_2 k_{\uparrow}^{(n)}} \pi_k^{(n)} + \sum_{\epsilon_2 k_{\uparrow}^{(n)} + 1}^{t_n} \pi_k^{(n)} \\
&\leq (\epsilon_1 k_{\downarrow}^{(n)}) \delta_{12}^{(1-\epsilon_1)k_{\downarrow}^{(n)}} + \sum_{k=\epsilon_1 k_{\downarrow}^{(n)}}^{\epsilon_2 k_{\uparrow}^{(n)}} \pi_k^{(n)} + \left(t_n - \epsilon_2 k_{\uparrow}^{(n)}\right) \delta_{14}^{(\epsilon_2 - 1)k_{\uparrow}^{(n)}} \\
&\leq n\delta_{12}^{(1-\epsilon_1)k_{\downarrow}^{(n)}} + n\delta_{14}^{(\epsilon_2 - 1)k_{\uparrow}^{(n)}} + \sum_{k=\epsilon_1 k_{\downarrow}^{(n)}}^{\epsilon_2 k_{\uparrow}^{(n)}} \pi_k^{(n)} \\
&= \delta_{12}^{\delta_{15}\frac{m}{c_n} - \log_{\delta_{12}} n} + \delta_{14}^{\delta_{17}\frac{m}{c_n} - \log_{\delta_{14}} n} + \sum_{k=\epsilon_1 k_{\downarrow}^{(n)}}^{\epsilon_2 k_{\uparrow}^{(n)}} \pi_k^{(n)},
\end{aligned}$$

for suitably chosen positive constants δ_{15}, and δ_{17}. Thus, the theorem follows by noting that $\epsilon_1, \epsilon_2, \delta_{11}$, and δ_{13} can be chosen arbitrarily close to 1. □

We now use (3.9) to provide a slightly simpler bound on $\tau_k^{(n)}$.

Lemma 10. *For large enough n, we get,*

$$\tau_k^{(n)} \leq \frac{(t_n - k + 1)\theta_n}{h^{(n)}(k) - k\theta_n} \left(\frac{1}{k}\pi_{k-1}^{(n)} + \frac{k-1}{k}\tau_{k-1}^{(n)} \right),$$

for each $k = 1, \ldots, t_n$.

Proof. Using (3.8) in (3.9), we get

$$\begin{aligned}
\tau_k^{(n)} &= \frac{\left(\begin{array}{r} \frac{(t_n - k + 1)\theta_n}{h^{(n)}(k) - k\theta_n}\pi_{k-1}^{(n)} + \frac{t_n - k + 1}{k}\pi_{k-1}^{(n)} \\ + \frac{(t_n - k + 1)(k-1)}{k}\theta_n\tau_{k-1}^{(n)} \end{array} \right)}{h^{(n)}(k) - k\theta_n} \\
&= \frac{(t_n - k + 1)\theta_n}{h^{(n)}(k) - k\theta_n}\left(\left(\frac{1}{h^{(n)}(k) - k\theta_n} + \frac{1}{k\theta_n} \right)\pi_{k-1}^{(n)} + \frac{k-1}{k}\tau_{k-1}^{(n)} \right).
\end{aligned}$$

Now, we have the lemma if we show that for large enough n the following holds: $\left(\frac{1}{h^{(n)}(k) - k\theta_n} + \frac{1}{k\theta_n}\right) \leq \frac{1}{k}$ for each $k = 1, \ldots, t_n$. This can be shown as follows.

One can show that Lemma 7 holds even when $p_k^{(n)} = 1 - e^{-\frac{kc_n}{m}}$. Using $g_n = \frac{\theta_n}{\gamma\xi' b}$, we get $h^{(n)}(k) = \xi' bnp_k^{(n)} \geq \frac{\delta_{20}}{\gamma}k\theta_n$ for large enough n and some constant δ_{20} such that $\gamma < \delta_{20} < 1$. Thus, $(h^{(n)}(k) - k\theta_n) \geq (\frac{\delta_{20}}{\gamma} - 1)k\theta_n$. For large enough n, $(\frac{\delta_{20}}{\gamma} - 1)\theta_n \geq 2$, and thus, $(h^{(n)}(k) - k\theta_n) \geq 2k$. Similarly, for large enough n, $k\theta_n \geq 2k$. Hence the lemma. □

Following lemma provides an even simpler bound on $\tau_k^{(n)}$.

Lemma 11. *For large enough n, we get,*

$$\tau_k^{(n)} \leq \pi_k^{(n)}$$

for each $k \in \{1, \dots, t_n\}$.

Proof. Fix n large enough such that the bound in Lemma 10 holds. We prove the result using induction on k. Consider the base case of $k = 1$. From Lemma 10 and (3.8) we have

$$\tau_1^{(n)} \leq \frac{(t_n - 1 + 1)\theta_n}{h^{(n)}(1) - \theta_n} \pi_0^{(n)} = \pi_1^{(n)}.$$

Now, let us assume that the lemma holds for $k = k' - 1$, i.e., $\tau_{k'-1}^{(n)} \leq \pi_{k'-1}^{(n)}$. Using this, we show below that the lemma holds for $k = k'$ as well.

From Lemma 10 and induction hypothesis we have

$$\tau_{k'}^{(n)} \leq \frac{(t_n - k' + 1)\theta_n}{h^{(n)}(k') - k'\theta_n} \left(\frac{1}{k'} \pi_{k'-1}^{(n)} + \frac{k' - 1}{k'} \pi_{k'-1}^{(n)} \right) = \pi_{k'}^{(n)},$$

where the last equality follows from (3.8). Hence, the lemma. □

Using above lemma and (3.7), we get

$$E[D^{(n)}] \leq \nu \sum_{k=1}^{t_n} \frac{k}{t_n} \pi_k^{(n)}.$$

Or equivalently,

$$\frac{1}{\nu} E[D^{(n)}] = \sum_{k=1}^{\epsilon b \log(\frac{1}{1-\gamma}) \frac{n}{c_n}} \frac{k}{t_n} \pi_k^{(n)} + \sum_{k=\epsilon' b \log(\frac{1}{1-\gamma}) \frac{n}{c_n} + 1}^{t_n} \frac{k}{t_n} \pi_k^{(n)}.$$

We now use Theorem 10 to prove the main result. From Theorem 10, we have

$$\frac{1}{\nu} E[D^{(n)}] \leq \sum_{k=1}^{\epsilon b \log(\frac{1}{1-\gamma}) \frac{n}{c_n}} \frac{k}{t_n} \pi_k^{(n)} + \delta'^{\frac{m}{c_n}}$$

$$
\begin{aligned}
&\leq \epsilon b \log\left(\frac{1}{1-\gamma}\right)\frac{n}{c_n t_n}\sum_{k=1}^{\epsilon b\log(\frac{1}{1-\gamma})\frac{n}{c_n}} \pi_k^{(n)} + \delta'^{\frac{m}{c_n}} \\
&\leq \epsilon \log\left(\frac{1}{1-\gamma}\right)\frac{\theta_n}{\gamma\xi' c_n} + \delta'^{\frac{m}{c_n}},
\end{aligned}
$$

where in last inequality we used definition of t_n. The first part of the theorem thus follows from definition of ξ', and the fact that ϵ and δ' where chosen arbitrarily.

Further, from Corollary 4, upon randomly placing c_n copies of each file, the associated random capacity region contains $\widetilde{\mathcal{C}}^{(n)}$ with high probability. Hence, the second part follows as well. □

3.6 Appendix

3.6.1 Equivalence of α-fair resource allocation policies

Clearly, for any α, α-fair resource allocations $\mathbf{r}(\mathbf{x})$ are Pareto efficient, i.e., for any state $\mathbf{x}$, there does not exist an $\mathbf{r}' \in \mathcal{C}$ such that $r_i' \geq r_i(\mathbf{x}),\ \forall i \in A_{\mathbf{x}}$ with a strict inequality for at least one $i \in A_{\mathbf{x}}$. Due to the existence of dominant face $\mathcal{D} = \{\mathbf{r} \in \mathcal{C} : \sum_{i\in F} r_i = \mu(F)\}$, α-fair resource allocation over capacity region $\mathcal{C}$ is equivalent to that over region $\mathcal{D}$.

We will show that α-fair resource allocations for any $\alpha \in (0,\infty)\backslash\{1\}$ are equivalent to Max-Min Fair (MMF) resource allocations. The result then follows immediately for $\alpha = 1$ as well since it is equivalent to the limiting α-fair allocation as $\alpha \to 1$.

Fix an $\alpha \in (0,\infty)\backslash\{1\}$. Without loss of generality, consider a state $\mathbf{x}$

such that $A_{\mathbf{x}} = F$. Consider the corresponding set of flows q_F. It is easy to show that an α-fair resource allocation over $\mathcal{D}$ is equivalent to assigning rates $(b_u : u \in q_F)$ as given by the unique solution to the following optimization problem:

$$\begin{aligned}
\text{maximize} \quad & \operatorname{sign}(1-\alpha) \sum_{u \in q_F} \hat{b}_u^{1-\alpha} \\
\text{subject to} \quad & \sum_{u \in q_A} \hat{b}_u \leq \mu(A),\ \forall A \subset F \\
& \sum_{u \in q_F} \hat{b}_u = \mu(F) \\
& \hat{b}_u \geq 0,\ \forall u \in q_F
\end{aligned}$$

The objective function for the above problem is strictly concave, and thus Schur-concave, in $(\hat{b}_u : u \in q_F)$ [32,38].

Now, suppose $(b_u : u \in q_F)$ is not max-min fair. Then, there exist flows u and v and a constant $\epsilon > 0$ such that $b_v \geq b_u$ and by increasing the rate of the flow u by ϵ and decreasing that of flow v by ϵ the feasibility for the above problem is not lost. However, due to Schur-concavity, this operation only increases the value of the objective function which contradicts with optimality and uniqueness of $(b_u : u \in q_F)$. Thus, $(b_u : u \in q_F)$ is max-min fair, and α-fair policy is equivalent to MMF.

3.6.2 Relative greediness and other resource allocation properties

Below, we provide a proof of Lemma 5 which asserts that αF is more greedy than BF. Along the way, we develop several other properties of the resource allocation policies.

Proof of Lemma 5 stems from the Properties (1) and (2) below on per-job rate assignment for αF and BF.

1.) *αF gives the most balanced per-job resource allocation:* This property follows from the fact that αF is equivalent to max-min fair resource allocation, see Proposition 2. Formally,

Lemma 12. *Let $\mathbf{b}^{\alpha}$ represent a vector of rates assigned to a set of flows under αF resource allocation. Let $\tilde{\mathbf{b}}$ be the rates assigned to the same set of flows under any other feasible resource allocation. Then, $\mathbf{b}^{\alpha} \prec^{w} \tilde{\mathbf{b}}$, i.e., weak majorized from above.*

Proof. Let the set of flows be $q_{A_{\mathbf{x}}}$. It is easy to show that $\mathbf{b}^{\alpha}$ is the unique solution to the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \operatorname{sign}(1-\alpha) \sum_{u \in q_{A_{\mathbf{x}}}} \hat{b}_u^{1-\alpha} \\ \text{subject to} \quad & \sum_{u \in q_A} \hat{b}_u \leq \mu(A), \ \forall A \subset A_{\mathbf{x}} \\ & \hat{b}_u \geq 0, \ \forall u \in q_F \end{aligned}$$

Also, since $\tilde{\mathbf{b}}$ is feasible, it satisfies the constraints of the above problem. The result then follows by noting that the objective function of the above problem is monotonic and Schur-Concave in $(\hat{b}_u : u \in q_{A_{\mathbf{x}}})$ [32,38]. □

2.) *In αF and BF, longest queues have smallest per-job rates:* For αF, this property again follows from the fact that it is equivalent to max-min fair, and that the capacity region is convex and symmetric. For BF, the proof for this property is technical and we omit its discussion here for brevity. Formally,

Lemma 13. *αF and BF resource allocations satisfy the following property for any state $\mathbf{x}$: if $x_i > x_j$ then $\frac{r_i(\mathbf{x})}{x_i} \leq \frac{r_j(\mathbf{x})}{x_j}$.*

Proof. Below, we prove the lemma for αF resource allocation. For a proof of this lemma for BF resource allocation, see Section 3.6.3.

Let $\mathbf{b}^\alpha = (b_u^\alpha : u \in q_{A_\mathbf{x}})$ represent the rates assigned to ongoing flows under αF resource allocation in state $\mathbf{x}$. Suppose $x_i > x_j$, but $\frac{r_i^\alpha(\mathbf{x})}{x_i} > \frac{r_j^\alpha(\mathbf{x})}{x_j}$. Then, then for each $u' \in q_i$ and $v' \in q_j$, we have $b_{u'}^\alpha > b_{v'}^\alpha$. Let $\tilde{\mathbf{b}} = (\tilde{b}_u : u \in q_{A_\mathbf{x}})$ where $\tilde{b}_u = b_u^\alpha$ for each $u \in q_{A_\mathbf{x} \setminus \{i,j\}}$ and $\tilde{b}_u = \frac{r_i^\alpha(\mathbf{x}) + r_j^\alpha(\mathbf{x})}{x_i + x_j}$ for each $u \in q_{\{i,j\}}$. It can be checked that $\tilde{b}_u$ is feasible and that $\tilde{\mathbf{b}} \prec^w \mathbf{b}^\alpha$. This contradicts Lemma 12. Hence the result. □

Now, let us study what the above properties imply for per-class resource allocation. Consider a state $\mathbf{x}$. Lemma 13 above implies that the most disadvantaged jobs are the ones which belong to longest queues for both, BF and αF. This, along with Lemma 13, implies that αF provides larger rate to longest queues. Thus we get the following property.

3.) *αF provides larger rate to longest queues compared to BF:* Formally, this property can be stated as follows:

Lemma 14. *For any state* $\mathbf{x}$*,* $\sum_{l=1}^{k} r_{(l)}^{\alpha}(\mathbf{x}) \geq \sum_{l=1}^{k} r_{(l)}^{B}(\mathbf{x})$ *for each* $k \in \{1, 2, \ldots, n\}$*.*

Proof. Let $u_1, u_2, \ldots, u_{x_{[1]}}$ be the flows in the class corresponding to $x_{[1]}$. Similarly, for each $k \in \{2, \ldots, n\}$, let $u_{\sum_{l=1}^{k-1} x_{[l]}+1}, \ldots, u_{\sum_{l=1}^{k} x_{[l]}}$ be the flows in the class corresponding to $x_{[k]}$. From Lemma 13, under both BF and αF resource allocation we have $b_{u_1} \leq b_{u_2} \leq \ldots \leq b_{u_{|x|}}$. Thus, it is enough to show that $\mathbf{b}^{\alpha} \prec^{w} \mathbf{b}^{B}$. However, this follows from Lemma 12. □

Now, we focus on αF and study how it allocates rates across classes for states $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{x} \prec \mathbf{y}$. Intuitively, jobs in longer queues in state $\mathbf{y}$ are more constrained than those in $\mathbf{x}$. Again using the fact that αF is equivalent to max-min fair, the most constrained jobs in state $\mathbf{y}$ have smaller rate than those in state $\mathbf{x}$. By monotonicity of αF, this holds even when $\mathbf{x} \prec_{w} \mathbf{y}$. When translated to per-class resource allocation in states $\mathbf{x}$ and $\mathbf{y}$, this argument leads us to the following property:

4.) *αF provides larger rate to longer queues in more balanced states:* Formally, this property can be stated as follows:

Lemma 15. *Consider states* $\mathbf{x}$ *and* $\mathbf{y}$ *such that* $\mathbf{x} \prec_{w} \mathbf{y}$*. For each* k *such that* $\sum_{l=1}^{k} x_{[l]} = \sum_{l=1}^{k} y_{[l]}$*, we have* $\sum_{l=1}^{k} r_{(l)}^{\alpha}(\mathbf{x}) \geq \sum_{l=1}^{k} r_{(l)}^{\alpha}(\mathbf{y})$*.*

Proof. Due to monotonicity of $\mathbf{r}^{\alpha}(\mathbf{y})$ with respect to components of $\mathbf{y}$, it is enough to show the result for the case where $\mathbf{x} \prec \mathbf{y}$. Assume, $\mathbf{x} \prec \mathbf{y}$.

Let $u_1, u_2, \ldots, u_{x_{[1]}}$ be the flows in the class corresponding to $x_{[1]}$. Similarly, let $u_{\sum_{l=1}^{k-1} x_{[l]}+1}, \ldots, u_{\sum_{l=1}^{k} x_{[l]}}$ be the flows in the class corresponding to $x_{[k]}$ for each $k \in \{2, \ldots, n\}$. Let the corresponding rates assigned to flows under αF resource allocation be given by $\mathbf{b}^{(\mathbf{x})}$. Using Lemma 13, we have $b_{u_1} \leq b_{u_2} \leq \ldots \leq b_{u_{|\mathbf{x}|}}$. Similarly, let $v_1, v_2, \ldots, v_{|\mathbf{y}|}$ be the flows corresponding to state $\mathbf{y}$ and construct the corresponding $\mathbf{b}^{(\mathbf{y})}$.

One can check that $\tilde{\mathbf{b}}^{(\mathbf{x})} = (\tilde{b}^{(\mathbf{x})}_{u_k} : k \in \{1, 2, \ldots, |\mathbf{x}|)$, where $\tilde{b}^{(\mathbf{x})}_{u_k} = b^{(\mathbf{y})}_{v_k}$ for each $k \leq |\mathbf{x}|$, is feasible under state $\mathbf{x}$ as well. Thus, from Lemma 12, we have $\mathbf{b}^{(\mathbf{x})} \prec^{w} \tilde{\mathbf{b}}^{(\mathbf{x})}$. From this, the result follows. □

Finally, we are ready to study relative greediness of αF and BF.

5.) *αF is more greedy than BF:* We now prove Lemma 5. Consider states $\mathbf{x}$ and $\mathbf{y}$ such that $\mathbf{x} \prec_w \mathbf{y}$. From Lemma 15 we have $\sum_{l=1}^{k} r^{\alpha}_{(l)}(\mathbf{x}) \geq \sum_{l=1}^{k} r^{\alpha}_{(l)}(\mathbf{y})$, and from Lemma 14 we have $\sum_{l=1}^{k} r^{\alpha}_{(l)}(\mathbf{y}) \geq \sum_{l=1}^{k} r^{B}_{(l)}(\mathbf{y})$. Hence, Lemma 5 holds.

3.6.3 In BF, longest queues have smallest per-job rates

Lemma 16. *For any state $\mathbf{x}$, if $x_i > x_j$ then $\frac{r_i^B(\mathbf{x})}{x_i} \leq \frac{r_j^B(\mathbf{x})}{x_j}$.*

Proof. Using definition of balanced fairness, we have $\frac{r_i^B(\mathbf{x})}{r_j^B(\mathbf{x})} = \frac{\Phi(\mathbf{x}-\mathbf{e}_i)}{\Phi(\mathbf{x}-\mathbf{e}_j)}$. Thus, we need to show that $\frac{\Phi(\mathbf{x}-\mathbf{e}_i)}{\Phi(\mathbf{x}-\mathbf{e}_j)} \leq \frac{x_i}{x_j}$. It is thus sufficient to prove that $\frac{\Phi(\mathbf{x}+\mathbf{e}_i)}{\Phi(\mathbf{x}+\mathbf{e}_j)} \geq \frac{x_j+1}{x_i+1}$ holds for each $\mathbf{x}$ since the result follows when $\mathbf{x}$ is replaced with $\mathbf{x} - \mathbf{e}_i - \mathbf{e}_j$.

We show below that $\frac{\Phi(\mathbf{x}+\mathbf{e}_i)}{\Phi(\mathbf{x}+\mathbf{e}_j)} \geq \frac{x_j+1}{x_i+1}$ holds for each $\mathbf{x}$.

Fix $i, j \in F$. By symmetry of balanced fairness and the capacity region, the result holds for each $\mathbf{x}$ such that $x_i = x_j$. We show that the result holds for each $\mathbf{x}$ such that $x_i \geq x_j$ using induction on $|\mathbf{x}|$. We will use the following recursive expression for $\Phi(.)$ which we get from definition of balanced fair and Proposition 2: For each state $\mathbf{x}$ we have,

$$\Phi(\mathbf{x}) = \frac{\sum_{i' \in A_{\mathbf{x}}} \Phi(\mathbf{x} - \mathbf{e}_{i'})}{\mu(A_{\mathbf{x}})}. \tag{3.12}$$

The result clearly holds for the base case of $|\mathbf{x}| = 0$. Assume that the result holds for all states $\mathbf{x}'$ such that $|\mathbf{x}'| < |\mathbf{x}|$. We prove that the result holds for the state $\mathbf{x}$ under each of the following two possible cases for $\mathbf{x}$:

<u>*Case 1*</u> $A_{\mathbf{x}+\mathbf{e}_i} \subsetneq A_{\mathbf{x}+\mathbf{e}_j}$: This case is possible only if $x_i > 0$ and $x_j = 0$. Thus, $\mu(A_{\mathbf{x}+\mathbf{e}_i}) \leq \mu(A_{\mathbf{x}+\mathbf{e}_j})$. Using (3.12), we get

$$\frac{\Phi(\mathbf{x}+\mathbf{e}_i)}{\Phi(\mathbf{x}+\mathbf{e}_j)} \geq \frac{\Phi(\mathbf{x}) + \sum_{i' \in A_{\mathbf{x}} \backslash \{i\}} \Phi(\mathbf{x}+\mathbf{e}_i - \mathbf{e}_{i'})}{\Phi(\mathbf{x}) + \Phi(\mathbf{x}+\mathbf{e}_j - \mathbf{e}_i) + \sum_{i' \in A_{\mathbf{x}} \backslash \{i\}} \Phi(\mathbf{x}+\mathbf{e}_j - \mathbf{e}_{i'})}.$$

Using induction hypothesis, we have $\frac{\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_{i'})}{\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_{i'})} \geq \frac{x_j+1}{x_i+1}$ for each $i' \in A_{\mathbf{x}} \backslash \{i\}$. Thus, using the fact that $\frac{a_1+a_2}{b_1+b_2} \geq \frac{x}{y}$ if $\frac{a_k}{b_k} \geq \frac{x}{y}$ for each $k \in \{1, 2\}$, the result follows if we show that $\frac{\Phi(\mathbf{x})}{\Phi(\mathbf{x})+\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)} \geq \frac{x_j+1}{x_i+1}$. This in turn follows since $x_j = 0$ and $\frac{\Phi(\mathbf{x})}{\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)} \geq \frac{1}{x_i}$ holds by induction hypothesis.

<u>*Case 2*</u> $A_{\mathbf{x}+\mathbf{e}_i} = A_{\mathbf{x}+\mathbf{e}_j}$: Again using (3.12), we get

$$\frac{\Phi(\mathbf{x}+\mathbf{e}_i)}{\Phi(\mathbf{x}+\mathbf{e}_j)} = \frac{\Phi(\mathbf{x}) + \Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_j) + \sum_{i' \in A_{\mathbf{x}} \backslash \{i,j\}} \Phi(\mathbf{x}+\mathbf{e}_i - \mathbf{e}_{i'})}{\Phi(\mathbf{x}) + \Phi(\mathbf{x}+\mathbf{e}_j - \mathbf{e}_i) + \sum_{i' \in A_{\mathbf{x}} \backslash \{i,j\}} \Phi(\mathbf{x}+\mathbf{e}_j - \mathbf{e}_{i'})}.$$

Again, using induction hypothesis we $\frac{\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_{i'})}{\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_{i'})} \geq \frac{x_j+1}{x_i+1}$ for each $i' \in A_{\mathbf{x}} \backslash \{i, j\}$. Thus, we only need to show that $\frac{\Phi(\mathbf{x})+\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_j)}{\Phi(\mathbf{x})+\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)} \geq \frac{x_j+1}{x_i+1}$. We show this below.

By induction hypothesis, we have $\frac{\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_j)}{\Phi(\mathbf{x})} \geq \frac{x_j}{x_i+1}$ and $\frac{\Phi(\mathbf{x})}{\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)} \geq \frac{x_j+1}{x_i}$. Thus, we get

$$\frac{\Phi(\mathbf{x})+\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_j)}{\Phi(\mathbf{x})+\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)} = \frac{1+\frac{\Phi(\mathbf{x}+\mathbf{e}_i-\mathbf{e}_j)}{\Phi(\mathbf{x})}}{1+\frac{\Phi(\mathbf{x}+\mathbf{e}_j-\mathbf{e}_i)}{\Phi(\mathbf{x})}} \geq \frac{1+\frac{x_j}{x_i+1}}{1+\frac{x_j+1}{x_i}} = \frac{x_j+1}{x_i+1}.$$

Hence, the result. □

3.6.4 Technical Lemmas for proof of Theorem 8

Lemma 7. Let a sequence $(g_n : n \in \mathbb{N})$ be such that $g_n = o(c_n)$. Let δ_1 be a positive constant independent of n such that $\delta_1 < 1$. Then, for large enough n, we have

$$p_k^{(n)} \geq \frac{\delta_1 g_n}{n} k \quad \forall k \in \left\{0, 1, \ldots, \left\lfloor \frac{n}{g_n} \right\rfloor \right\}.$$

Proof. Consider a sequence of functions $\left(f^{(n)}(.)\right)_{n\geq 1}$ where for each n, $f^{(n)}(t) = 1-(1-1/(bn))^{tc_n}$ for each $t \in \mathbb{R}_+$. Then,

$$f^{(n)}\left(n/g_n\right) = 1-(1-1/(bn))^{\frac{nc_n}{g_n}} \overset{n\to\infty}{\longrightarrow} 1.$$

Thus, there exists an integer n' such that $f^{(n)}\left(n/g_n\right) \geq \delta_1$ for all $n \geq n'$. Also, $f^{(n)}(0) = 0$ for each n. Using concavity of $f^{(n)}(.)$, for each $n \geq n'$ we have

$$f^{(n)}\left(t\right) \geq \frac{f^{(n)}\left(n/g_n\right)}{\left(n/g_n\right)} t, \quad \forall t \text{ s.t. } 0 \leq t \leq n/g_n.$$

Hence, the lemma. □

Lemma 8. There exists a positive constant δ such that $H\left(p_k^{(n)}(1-\epsilon)||p_k^{(n)}\right) \geq -\delta + \epsilon\frac{kc_n}{m}$.

Proof. From definition,

$$H\left(p_k^{(n)}(1-\epsilon)||p_k^{(n)}\right) = p_k^{(n)}(1-\epsilon)\log(1-\epsilon)+(1-p_k^{(n)}(1-\epsilon))\log\left(\frac{1-p_k^{(n)}(1-\epsilon)}{1-p_k^{(n)}}\right)$$

Here, the term $p_k^{(n)}(1-\epsilon)\log(1-\epsilon)$, while negative, is greater than $(1-\epsilon)\log(1-\epsilon)$, a constant. Similarly, the term $(1-p_k^{(n)}(1-\epsilon))\log\left(1-p_k^{(n)}(1-\epsilon)\right)$ is negative, but can be upper-bounded by a constant as follows:

$$(1-p_k^{(n)}(1-\epsilon))\log\left(1-p_k^{(n)}(1-\epsilon)\right) \geq \log\left(1-p_k^{(n)}(1-\epsilon)\right) \geq \log(1-(1-\epsilon)) = \log\epsilon$$

Thus, we have

$$H\left(p_k^{(n)}(1-\epsilon)||p_k^{(n)}\right) \geq -\delta + (1-p_k^{(n)}(1-\epsilon))\log\left(\frac{1}{1-p_k^{(n)}}\right)$$
$$\geq -\delta + (1-(1-\epsilon))\log\left(\frac{1}{1-p_k^{(n)}}\right) = -\delta + \epsilon\log\left(\frac{1}{1-p_k^{(n)}}\right) \geq -\delta + \epsilon\frac{kc_n}{m},$$

where in the last inequality we used the fact that $1-p_k^{(n)} \leq e^{-\frac{kc_n}{m}}$. □

Chapter 4

Impact of Splitting Files and Parallelism Constraints

In our previous chapters, we assumed the following: if c servers are pooled to serve a file-download request, each of those c servers already has a replica of the entire file. Thus, while delays scale as $1/c$, the memory requirement scales linearly with c. Can one improve this tradeoff? As we will see below, it is indeed possible to consider increasingly larger server pools without scaling memory requirement, but the maximum server pool size may be limited by system imposed parallelism constraints. This is achieved via splitting files in smaller blocks before replicating them on different servers. The goal of this chapter is to exposit the role of memory and parallelism constraints on delays.

4.1 Related work

Splitting files before replication is a common technique used in distributed content delivery systems such as P2P networks [15, 23, 61, 65, 66]. In P2P systems users collectively share content via the Internet. As mentioned earlier, our focus is on a centralized infrastructure aimed at serving large files very quickly. Unlike P2P networks, our system consists of dedicated and collocated servers which operate at all times, thus files are always available. Further, while user arrivals and departures are dynamic in both systems, the file-server association is static in our system. In this chapter we use splitting as an approach to reduce memory requirements over replication of whole files, while still achieving resource pooling benefits in centralized systems.

4.2 Memory vs. performance tradeoff

Let us first recall some of the key insights developed from performance comparison results in Section 2.5. Using NP-BF resource allocation policy, where non-overlapping server pools of size c where created, we observed that the delays scale as $1/c$. Then, by allowing server pools to overlap and using an appropriate load balancing strategy, as in RP-BF policy, we observed that the mean delay can be significantly reduced while maintaining the inverse relationship of delays with respect to c.

For simplicity, let us first explore the tradeoff between memory and delays for the simpler NP-BF policy. For completeness, let us summarize the NP-BF policy considered in Section 2.5. The system consists of n files and m servers, with peak service rate for each server being ξ. We divide the server set into m/c groups, each of size c. Each file is replicated across c servers belonging to a single group, where each group stores nc/m number of files. Thus, we create m/c independent server pools.

If arrival rates are symmetric ($\lambda_i = \lambda m/n$ for each class i) and so are mean service requirements ($\nu_i = \nu$ for each i), then the dynamics for each server pool can be modeled as an independent $M/GI/1$ queue with load $\rho c = \lambda \nu c$ and service capacity ξc. If the jobs per server pool are served via processor sharing discipline (equivalently, Balanced Fairness for $M/GI/1$ queues), then the mean delay in serving a job is given as:

$$E[D] = \frac{\nu}{c\xi(1 - \rho/\xi)}.$$

Thus, delays scale as $1/c$. Further, suppose that each file requires one unit of storage at a server where it is replicated. Then, the memory requirement per server is nc/m. Thus, memory requirement scales linearly with c.

Now, consider the following modification to NP-BF:

NP-BF-Split: Split each file into c equal blocks. Each file is associated with a single group of c servers as in NP-BF, but the file-placement is modified as follows: each block of the file is stored on a unique server within the group. Thus, each server stores $1/c$th fraction of a file. The memory requirement per server reduces to n/m. Further, since the server pools are independent, processor sharing of jobs is feasible and the mean delay in serving a job is same as that for NP-BF. Thus, job delays scale as $1/c$ but memory requirement is independent of c.

Thus, when the servers pools are non-overlapping, one need not tradeoff memory for improvement in performance by increasing the size of resource pools c. Note, however, some systems might impose a parallelism constraint which limits the maximum number of servers one can use in parallel. Thus, one may not increase c arbitrarily.

The above tradeoff reflects the impact of splitting a file into multiple blocks when the server pools are non-overlapping. In the next section we explore the gains of splitting when server pools overlap.

4.3 Gains of splitting files under overlapping server pools

We have showed that for a system with non-overlapping server pools the gains of splitting files before replication are significant, in that achieving scalable delays does not require scaling of memory requirements. Further, as we showed in Section 2.5 using the RP-BF policy and allowing the server pools to overlap enables better load balancing across the server pools leading to significantly lower mean delays. Can we simultaneously achieve the gains of overlapping pools and of splitting files?

A challenge towards achieving this is following: the load balancing associated with RP-BF required the flexibility that a chunk for a file may be downloaded from an arbitrary server. However, splitting of files into blocks and storing them on different servers in a pool reduces the flexibility in fetching a file chunk, in that it may be downloaded from servers which store the associated blocks.

Below, we provide a policy which accounts for this reduced flexibility and yet achieves load-balancing gains.

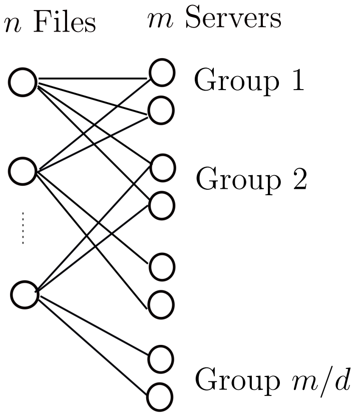


Figure 4.1: Creating server groups for block replication in RP-BF-Split

RP-BF-Split: Consider a system with n files and m servers, with peak service rate for each server being ξ. Split each file $i \in F^{(n)}$ into d blocks of equal size, namely, $i^{(1)}, i^{(2)}, \ldots, i^{(d)}$. Divide the server set into m/d groups, each of size d. For each $1 \leq l \leq m/d$, label servers in the l^{th} group by $s^{(l,1)}, s^{(l,1)}, \ldots, s^{(l,d)}$. Replicate blocks of files across different servers as follows (see Fig. 4.1). For each file i, choose c groups of servers at random. If the groups chosen are $l_1, l_2, \ldots, l_c$ then replicate block $i^{(k)}$ into servers $s^{(l_1,k)}, s^{(l_2,k)}, \ldots, s^{(l_c,k)}$.

Thus, each group of server stores the same set of files. We shall coordinate the service a job receives from a group as follows: a job u receives equal service rate $b_{u,s}$ from each server in a group. Thus, each server group essentially behaves like a single server with peak service rate $d\xi$. Thus, for a given realization of random group selections, the capacity region is a (possibly asymmetric) polymatroid.

Now let us study the delay vs. memory tradeoff achieved by RP-BF-

Split. Suppose that each file requires one unit of storage. Since c copies of each file are stored in the system, the memory requirement per server is nc/m. Below we provide an asymptotic analysis to obtain mean delay scaling for such a system.

To account for the random group selection, we shall use an approach similar to that developed in Section 2.4.1 and study asymptotic performance under the 'averaged capacity region' which is symmetric and hence simplifies analysis. For a system with n files and m servers, the associated averaged rank function is given by $\bar{\mu}^{(m,n)}(A) = h^{(m,n)}(|A|)$ for each $A \subset F^{(n)}$ where

$$h^{(m,n)}(k) = \xi m(1 - (1 - cd/m)^k) \text{ for } k = 0, 1, \ldots, n. \tag{4.1}$$

This follows since the probability that none of A's element files is stored at a group is $(1 - cd/m)^k$, so $\frac{m}{d}(1 - (1 - cd/m)^k)$ is the mean number of groups that can serve at least one file in A.

Note that, in the RP-BF-Split, resources of cd servers are pooled to serve a job, which thus receives the maximum service rate equal to ξcd if no other job exists in the system.

Along the lines of Theorem 4, one can show the following result:

Theorem 11. *Consider a sequence of* (m, n) *averaged RP-BF-Split systems with symmetric polymatroid capacity region with rank function as* $\bar{\mu}^{(m,n)}(A) = h^{(m,n)}(|A|)$ *for each* $A \subset F^{(n)}$ *where* $h^{(m,n)}(k) = \xi m(1 - (1 - cd/m)^k)$ *for* $k = 0, 1, \ldots, n$*. Let the load across files be symmetric with* $\rho_i^{(m,n)} = m\rho/n$ *for each* *file i where $\rho = \lambda\nu < \xi$. Then, the expected delay satisfies:*

$$\lim_{m\to\infty}\lim_{n\to\infty} E[D^{(m,n)}] = \frac{1}{\lambda cd}\log\left(\frac{1}{1-\rho/\xi}\right). \qquad (4.2)$$

The above result shows that the mean delays scale as $\frac{1}{cd}$. Recall, memory requirement scales linearly in c and is independent of d. If the system is subject to a constraint that server pools be no larger than β, what values of c and d should be chosen? To gain maximum advantages of resource pooling and load balancing while minimizing memory requirement, one may choose cd to be equal (or as close as possible) to β while choosing c to be as low as possible.

To be able to achieve load balancing gains under RP-BF-Split, one needs to replicate the blocks of each file over at least 2 server groups for server diversity. Thus, under parallelism constraints, memory vs. delay tradeoff is optimized by choosing $c = 2$ and $d = \beta/2$.

Thus, in RP-BF-Split, each file has at least two replicas in the system. Contrast this with NP-BF-Split where each file is replicated only once. A question arises as to whether one can further reduce the memory requirement under RP-BF-Split while still achieving load balancing gains. This is equivalent to asking the following question: does one need server diversity of 2 in order to achieve load balancing gains in systems employing resource pooling?

A recent paper [63] for a different setting in which load balancing is done via a distributed routing policy suggests that one may only need a diversity of value 'slightly greater than one' in order to achieve load balancing gains.

To achieve such a fractional diversity for content delivery systems employing pooling of servers, one needs a 'fractional' replication strategy. A fractional replication strategy can be achieved via network coding for storage systems [18] where d blocks of a file are converted into $d' > d$ number of encoded blocks which are then stored across d' different servers. The original file (or a block from a file) can be recovered by accessing any d of the d' encoded blocks.

Such codes have been used to improve reliability against server failures [18]. Further, network coding has also been used to improve performance in distributed content delivery systems such as P2P networks [23, 66]. However, their role in achieving load balancing gains in centralized content delivery systems via fractional diversity in server choices is an interesting open problem.

Chapter 5

Concentration in Servers' Activity and Impact of Bottlenecks

An important problem in engineering large scale content delivery systems is the optimization and efficient use of resource bottlenecks. The design of such systems is made complex by the dynamic characteristics of service demands, which include stochastic arrivals of user requests/jobs, diversity in demand types, and random service requirements.

System designers often adopt a pessimistic approach towards resource allocation, in that, they aim for acceptable user-performance under extreme or even worst-case scenarios. However, such extreme scenarios may be unlikely (or may be made unlikely) and a pessimistic design may result in overprovisioning. A basic question in this setting is: For what system configurations and demand characteristics can we be optimistic in provisioning resources?

This chapter has four key messages which we discuss below. The first message: *concentration in servers' activity facilitates resource provisioning.* As systems become large and service types become more diverse such that no single service dominates resource usage, the load across individual servers becomes increasingly uncorrelated. This may in turn result in concentration of servers' activity, i.e., the distribution of the number of active servers is concentrated around its mean. Such a result enables one to provision for the peak power capacity to be close to the average power requirement without a significant risk of overload. Similarly, for content delivery applications where activity of a server is connected to the rate at which bits are downloaded from the server, such concentration results would allow one to provision for a shared network link with capacity close to the average traffic demand without significantly affecting user-performance.

Existence of such a concentration result depends on the extent to which there is diversity and independence in the load spread across the servers. To better understand how diversity in service types impacts servers' activity, consider a system with m servers, each with service capacity μ. Let the job arrival rate be λm and mean service requirement of jobs be ν. For stability, assume $\lambda\nu < \mu$. As with previous chapters, we will consider systems that use resource pooling, in that the capacity of multiple servers may be pooled together as follows: if k servers are pooled together to serve a job, the job can be served at a maximum rate of $k\mu$. Note, however, these resources are shared among jobs and the pools may overlap. In this setting, consider the following two extreme cases:

Case 1: *Single service type and complete resource pooling:* Suppose that jobs belong to a single service type, and that all m servers can be pooled to serve each job. This system can be modeled as a $G/G/1$ queue with arrival rate λm and load $\rho = \lambda\nu/\mu$. For a work-conserving service policy, either all m servers are active or idle at the same time with probability ρ and $1-\rho$ respectively.

Case 2: *Multiple service types and no resource pooling:* Now, suppose that there are m job classes. Each job class has a dedicated server. The arrivals and service requirements of different classes are independent. Suppose the arrival rate for each class is λ, and mean service requirement for jobs in each class is ν. This system can be modeled as consisting of m independent $G/G/1$ queues, each with load $\rho = \lambda\nu/\mu$. For queues with work conserving service policy, at any time t each server is active with probability ρ and the activities of different servers are independent. By Weak Law of Large Numbers, for any $\epsilon > 0$ the stationary probability that the number of active servers exceeds $(1+\epsilon)\rho m$ tends to 0 as $m \to \infty$.

In Case 2 the servers' activity concentrate due to independence in load, thus facilitating resource provisioning for the large scale system. By contrast, in Case 1 the activities of different servers are correlated due to complete resource pooling and one may need to provision for the peak number of servers being active. Thus, a question arises: do servers' activity concentrate in systems where limited resource pooling is allowed? Such systems fall in between the above two extreme cases, in that, there may be diverse service types and a limited amount of resource pooling which correlates instantaneous server activities.

The second message: *servers' activity concentrate even if we allow limited resource pooling of servers.* To better understand the impact of limited resource pooling, in this chapter we consider multi-class multi-server systems where for each job class the capacity of a unique subset of servers can be pooled to jointly serve the class's jobs. Furthermore the pools of servers serving different classes may overlap, which opens up an opportunity to dynamically vary the allocation of service capacity across job classes. We consider the case where the service rate allocated to each class depends on the numbers of jobs across classes, and call the corresponding policy a resource allocation policy.

Such a system can model a centralized content delivery infrastructure where each file is replicated across multiple servers so as to address high demands and possible reliability issues. Systems which combine multipath transport with server diversity can support parallel downloads from multiple servers, where different chunks of a file can be downloaded in parallel from servers possibly via multiple paths. The resource allocation policy would thus model the dynamically varying sum-rate a download job receives from its server pool.

To understand the role of overlapping pools on possible concentration properties of such systems, we consider a sequence of systems where the number of servers m grows. We allow total system load and total server capacity to scale linearly with m. For a given m we consider server pools of fixed size $c^{(m)}$, which may scale with m but as $o(m)$. We assume that the load across different server pools (equivalently, job classes) is homogeneous. This may be achieved by, for example, grouping of several service types into a class so that the overall load per group is roughly the same. For such a system, we show that the joint stationary distribution of the activity of a *fixed finite* subset of servers takes a product form as $m \to \infty$, which in turn implies that a WLLN holds for the servers' activity. In summary, as long as resource pools are of size $o(m)$ one will see a concentration in server activity.

The above concentration result is 'insensitive', in that, the dependence on the service requirement distribution for each class is only through its mean. This follows from our adoption of insensitive balanced fair resource allocation [7]. This brings us to the third message: *one's optimism in resource allocation due to concentration in servers' activity is independent of service requirement distribution, i.e., only depends on its mean.* This is analogous to insensitivity in symmetric queueing systems where the distribution of the number of active jobs in a system is known to be insensitive to service-time distributions [28], although our interest is mainly in the distribution of servers' activity for large scale coupled systems.

The concentration result in turn allows us to show that the impact of shared network link becomes negligible as the system scales, provided its capacity is a fraction greater than the average traffic demand. We also consider in this chapter the impact of peak rate constraints on jobs' service. For a content delivery application, such constraints can model the impact of finite download capacity at users end under a simplifying assumption that each user requests a maximum of one job at a time. Peak rate constraints can also model end-to-end flow-control such as TCP transmission with finite window size, where each flow corresponds to a file download job.

We incorporate peak rate constraints into our performance analysis via balanced fairness. We show that even under arbitrary peak rate constraints for each job the underlying polymatroid structure of the system's capacity region is preserved. This in turn allows us to extend our analysis in Section 2.3 to provide a modified expression for mean delays. However, the expression is complex, and we thus resort to symmetry to gain insights. The insights gleaned form our fourth message: *If the overall load on the content delivery system is low, peak rate constraints drive the user performance. However, as the load increases their impact on performance reduces and eventually becomes negligible.*

5.1 Related work

Prior work which is closest in spirit to our concentration result in this chapter is that studying the existence of a mean field regime for the super-market queuing model [13, 43, 56]. In the super-market model the servers are coupled through a routing policy, unlike our model where they are coupled through a servicing policy. In the supermarket model, upon arrival of a job a random subset of servers of size d is selected and the job is routed to the server with the least number of jobs waiting for service. For a fixed value of d, asymptotic independence in the number of waiting jobs for a fixed finite subset of servers was shown in [13] for several classes of service distributions.

A mean field result for the number of waiting jobs was also shown for a symmetric loss network model [29, 57], where upon arrival of a job (or a call in their terminology) it is allocated to a fixed w number of servers at random. In this work, rather than routing to one of the w servers as in supermarket model, each job 'locks' resources at w servers for a random time. The maximum number of jobs that can lock resources at a given server at any point in time is fixed. Again, w is assumed to be constant. Further, the random locking time is assumed exponential.

In comparison, in this chapter we consider a setting where a job arrives with a random service requirement, is served jointly by a subset of servers, and leaves the system upon completion of its service. Sojourn times of jobs thus depend on how server resources are shared across different job-types. We allow the number of servers that can be pooled together for serving a job to scale with m. We also let the distribution of the service requirement be arbitrary. Under these assumptions, we are able to show the existence of a mean field only for servers' activity, and not for the number of waiting jobs.

Mean field results for queuing systems have been studied for several other models and asymptotic regimes as well, e.g., [2, 51, 55]. Most of the prior work show mean field existence by analyzing sample paths of the underlying stochastic processes. However, we could use the knowledge of stationary distribution of waiting jobs under balanced fair resource allocation [7]. Since its proposal in [7] as a bandwidth sharing policy for wireline network, it has been a useful device towards analyzing user-performance in several kinds of network models [6, 8, 10, 47, 49].

Outline of the chapter: In Section 5.2 we provide our main result where we show concentration in servers' activity. In Section 5.3 we consider implications of this result on shared network link and power capacity. In Section 5.4 we study the impact of finite download capacity at the end user.

5.2 Asymptotic independence and concentration in servers' activity

Consider a system with a set $S^{(m)} = \{s_1, s_2, \ldots, s_m\}$ of m servers. Each server has service capacity $\xi > 0$. Jobs arrive into the system as an independent Poisson process with rate λm. Job service requirements are i.i.d. with mean ν. Let $\rho = \lambda\nu$. We assume that $\rho < \xi$ to ensure stability. Upon arrival of a job, $c^{(m)} > 1$ servers are chosen at random, and their capacity pooled, to serve this job. Let $F^{(m)}$ represent the set of all possible pools of size $c^{(m)}$. Let $n^{(m)} \triangleq |F^{(m)}|$. Thus, $n^{(m)} = \binom{m}{c^{(m)}}$.

We view the system as consisting of $n^{(m)}$ job classes, where arrivals for each class occur as an independent Poisson process with rate $\lambda m/n^{(m)}$. Let $\boldsymbol{\rho}^{(m)} = (\rho_i^{(m)} : i \in F^{(m)})$, where $\rho_i^{(m)} = \rho m/n^{(m)}$ denotes the load associated with class i.[1] We view the association of classes with server pools via a bipartite

[1] This model may be generalized in the following ways without affecting our results in this section:
1) The service requirement distribution may be different for each class as long as the mean service requirement is same for each class.
2) The arrival rate and mean service requirement may be different for each class as long as their product (which equals to $\rho_i^{(m)}$) is same for each class.

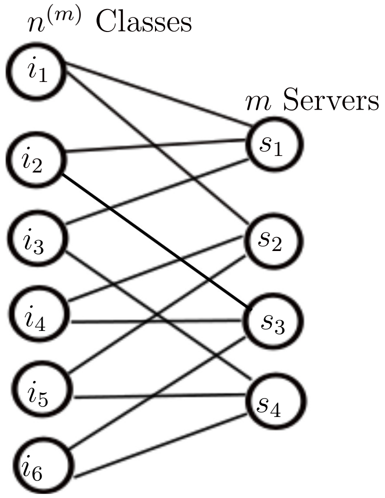


Figure 5.1: Graph $\mathcal{G}^{(m)} = (F^{(m)} \cup S^{(m)}; E^{(m)})$ for $m = 4$ and $c^{(m)} = 2$ modeling availability of servers $S^{(m)}$ to serve jobs in classes $F^{(m)}$.

graph $\mathcal{G}^{(m)} = (F^{(m)} \cup S^{(m)}; E^{(m)})$ where an edge $e \in E^{(m)}$ exists if it connects a class $i \in F^{(m)}$ to a server $s \in S^{(m)}$ associated with the server pool of i, see Fig. 5.1. For each class $i \in F^{(m)}$, let $S_i^{(m)}$ denote its neighbors, i.e., the set of servers available to serve jobs in class i. Let the capacity of the associated system be $\mathcal{C}^{(m)}$ with rank function $\mu^{(m)}(.)$. Further, the following holds for the rank function $\mu^{(m)}(.)$.

Proposition 3. *For each $k \leq n^{(m)}$, we have*

$$\sum_{A \subset F^{(m)}:|A|=k} \mu^{(m)}(A) = \xi m \left(\binom{n^{(m)}}{k} - \binom{n^{(m-1)}}{k} \right).$$

The proof of this proposition is straightforward. Notice that the term $\binom{n^{(m)}}{k} - \binom{n^{(m-1)}}{k}$ captures the number of subsets of $F^{(m)}$ of size k which are served by a given server.

System dynamics are as described in Section 2.2. We assume balance fair resource allocation. Thus, when number of waiting jobs in each class is given by vector $\mathbf{x}$, the service-rate allocated to each class $i \in F^{(m)}$ is given by:

$$r_i^{(m)}(\mathbf{x}) = \frac{\Phi^{(m)}(\mathbf{x} - \mathbf{e}_i)}{\Phi^{(m)}(\mathbf{x})}, \tag{5.1}$$

where the function $\Phi^{(m)}$ is called a balance function and is defined recursively as follows: $\Phi^{(m)}(\mathbf{0}) = 1$, and $\Phi^{(m)}(\mathbf{x}) = 0$ $\forall \mathbf{x}$ s.t. $x_i < 0$ for some i, otherwise,

$$\Phi^{(m)}(\mathbf{x}) = \max_{A \subset A_{\mathbf{x}}} \left\{ \frac{\sum_{i \in A} \Phi^{(m)}(\mathbf{x} - \mathbf{e}_i)}{\mu^{(m)}(A)} \right\}. \tag{5.2}$$

Further, if $\rho < \xi$, one can check that $\boldsymbol{\rho}^{(m)}$ lies in the interior of $\mathcal{C}^{(m)}$. Recall, this implies that the process $(\mathbf{X}^{(m)}(t) : t \in \mathbb{R})$ is stationary under balanced fair resource allocation. Further, the stationary distribution is given by

$$\pi^{(m)}(\mathbf{x}) = \frac{\Phi^{(m)}(\mathbf{x})}{G^{(m)}(\boldsymbol{\rho}^{(m)})} \prod_{i \in A_{\mathbf{x}}} \left(\rho_i^{(m)}\right)^{x_i}, \tag{5.3}$$

where,

$$G^{(m)}(\boldsymbol{\rho}^{(m)}) = \sum_{\mathbf{x}'} \Phi^{(m)}(\mathbf{x}') \prod_{i \in A_{x'}} \left(\rho_i^{(m)}\right)^{x'_i}.$$

Recall, $\rho_i^{(m)} = \rho m / n^{(m)}$ for each i.

Proceeding along the lines of Theorem 4 , one can show the following proposition.

Proposition 4 ([47]). *For each m, the following holds under balanced fairness: For each $A \subset F^{(m)}$, we have*

$$Pr_{\pi^{(m)}}\left(A_{\mathbf{X}^{(m)}} = A\right) = \frac{G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{G^{(m)}(\boldsymbol{\rho}^{(m)})},$$

where $G_A^{(m)}(\boldsymbol{\rho}^{(m)})$ can be computed recursively as follows. Let $G_\emptyset^{(m)}(\boldsymbol{\rho}^{(m)}) = 1$. Then,

$$G_A^{(m)}(\boldsymbol{\rho}^{(m)}) = \frac{\sum_{i\in A} \rho_i^{(m)} G_{A\setminus\{i\}}^{(m)}(\boldsymbol{\rho})}{\mu^{(m)}(A) - \sum_{j\in A} \rho_j^{(m)}}. \tag{5.4}$$

Recall, for each $s \in S^{(m)}$, $Y_s^{(m)}(t) = \mathbf{1}_{\left\{\exists i\in A_{\mathbf{X}(t)} \text{ s.t. } s\in S_i^{(m)}\right\}}$. We say that the server is active at time t if $Y_s^{(m)}(t)$ is 1. For a given m, for a stationary system we have

$$E_{\pi^{(m)}}\left[\mu^{(m)}(A_{\mathbf{X}^{(m)}})\right] = \rho m, \tag{5.5}$$

i.e., the average service rate must be equal to the system load. Further, by Pareto optimality of the balanced fair resource allocation for our system, for each $s \in S^{(m)}$ we have

$$E_{\pi^{(m)}}[Y_s^{(m)}] = \rho/\xi.$$

Indeed, showing concentration in $\sum_{l=1}^m Y_{s_l}^{(m)}$ as $m \to \infty$ is equivalent to showing concentration in $\mu^{(m)}(A_{\mathbf{X}^{(m)}})$ close to its mean. Further, for a given m, $\left(Y_{s_1}^{(m)}, Y_{s_2}^{(m)}, \ldots, Y_{s_m}^{(m)}\right)$ is an exchangeable vector of random variables. A weak convergence result for a sequence of exchangeable vectors was shown in [1, 26], which when applied to $\left(\left(Y_{s_1}^{(m)}, Y_{s_2}^{(m)}, \ldots, Y_{s_m}^{(m)}\right) : m \in \mathbb{N}\right)$ implies that $\sum_{l=1}^m Y_{s_l}^{(m)}$ converges to a constant in probability if and only if the joint-distribution of $\left(Y_{s_1}^{(m)}, Y_{s_2}^{(m)}, \ldots, Y_{s_k}^{(m)}\right)$ for a finite k takes a product form as $m \to \infty$. The following theorem is the first main result in this paper.

Theorem 12. *Consider a sequence of systems with an increasing number of servers m. Suppose that the total arrival rate of jobs is λm for the m^{th} system,* *and that the service capacity of each server is a constant ξ. Let load per server $\rho = \lambda\nu$ be a constant such that $\rho < \xi$.*

Upon arrival of a job, $c^{(m)} > 1$ servers are selected at random for its service (equivalently, its class is selected at random). Assume $c^{(m)}$ is $o(m)$. Jobs share the server resources according to the balanced fair resource allocation. For each m, the system is stationary. Under stationary distribution, let $Y_s^{(m)}$ represent instantaneous activity of server s.

Then, the following equivalent statements hold:

(a) For any finite integer k, the random variables $Y_{s_1}^{(m)}, Y_{s_2}^{(m)}, \ldots, Y_{s_k}^{(m)}$ are asymptotically i.i.d. as $m \to \infty$.

(b) $\lim_{m\to\infty} \frac{\sum_{l=1}^{m} Y_{s_l}^{(m)}}{m} = E\left[Y_{s_1}^{(m)}\right]$ *in probability.*

Proof: Equivalence of *(a)* and *(b)* thus follows from Proposition 7.20 in [1]. We prove *(a)* below for $\xi = 1$ without loss of generality. Again by Proposition 7.20 in [1], it is sufficient to show that the result holds for $k = 2$.

For the proof below, to make the dependence on ρ of stationary distribution $\pi^{(m)}$ and random variable $Y_s^{(m)}$ explicit, we denote them as $\pi^{(m,\rho)}$ and $Y_s^{(m,\rho)}$.

Let

$$\mathcal{T}_{s,1}^{(m)} \triangleq \{A \subset F^{(m)} : s \in \cup_{i\in A} S_i^{(m)}\}$$

and similarly,

$$\mathcal{T}_{s,0}^{(m)} \triangleq \{A \subset F^{(m)} : s \notin \cup_{i\in A} S_i^{(m)}\}$$

Recall the definitions of $G_A^{(m)}(\boldsymbol{\rho}^{(m)})$ and $G^{(m)}(\boldsymbol{\rho}^{(m)})$. Then, for $b \in \{0, 1\}$, we have

$$\begin{aligned}
Pr\left(Y_s^{(m,\rho)} = b\right) &= \sum_{\mathbf{x} \text{ s.t. } A_{\mathbf{x}} \in \mathcal{T}_{s,b}^{(m)}} \pi^{(m,\rho)}(\mathbf{x}) \\
&= \sum_{A \in \mathcal{T}_{s,b}^{(m)}} \frac{G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{G^{(m)}(\boldsymbol{\rho}^{(m)})} \\
&= (1-\rho)\mathbf{1}_{\{b=0\}} + \rho \mathbf{1}_{\{b=1\}}.
\end{aligned}$$

Further,

$$\begin{aligned}
Pr\left(Y_{s_1}^{(m,\rho)} = b_1, Y_{s_2}^{(m,\rho)} = 0\right) &= \sum_{A \in \mathcal{T}_{s_1,b_1}^{(m)} \cap \mathcal{T}_{s_2,0}^{(m)}} \frac{G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{G^{(m)}(\boldsymbol{\rho}^{(m)})} \\
&= \frac{\sum_{A \in \mathcal{T}_{s_1,b_1}^{(m)} \cap \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})} \frac{\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{G^{(m)}(\boldsymbol{\rho}^{(m)})} \\
&= \frac{\sum_{A \in \mathcal{T}_{s_1,b_1}^{(m)} \cap \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})} Pr\left(Y_{s_2}^{(m,\rho)} = 0\right) \\
&= \frac{\sum_{A \in \mathcal{T}_{s_1,b_1}^{(m)} \cap \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})}{\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})} (1-\rho) \qquad (5.6)
\end{aligned}$$

Consider the denominator $\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})$. By symmetry we have $\sum_{A \in \mathcal{T}_{s_2,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)}) = \sum_{A \in \mathcal{T}_{s_m,0}^{(m)}} G_A^{(m)}(\boldsymbol{\rho}^{(m)})$. Also for each $A \in \mathcal{T}_{s_m,0}^{(m)}$,

$$G_A^{(m)}(\boldsymbol{\rho}^{(m)}) = \sum_{\mathbf{x}: A_{\mathbf{x}} = A} \Phi^{(m)}(\mathbf{x}) \left(\frac{m}{n^{(m)}}\rho\right)^{|\mathbf{x}|}$$

$$= \sum_{\mathbf{x}:A_{\mathbf{x}}=A} \Phi^{(m)}(\mathbf{x}) \left(\frac{m-1}{n^{(m-1)}} \frac{(m-c^{(m)})\rho}{m-1} \right)^{|\mathbf{x}|},$$

since $\frac{m-1}{n^{(m-1)}} \frac{m-c^{(m)}}{m-1} = \frac{m}{n^{(m)}}$.

However, it is easy to check that $\mathcal{T}^{(m)}_{s_m,0}$ is the power set of $F^{(m-1)}$. Thus,

$$\begin{aligned}
\sum_{A \in \mathcal{T}^{(m)}_{s_m,0}} G^{(m)}_A(\boldsymbol{\rho}^{(m)}) &= \sum_{A \subset F^{(m-1)}} \sum_{\mathbf{x}:A_{\mathbf{x}}=A} \Phi^{(m-1)}(\mathbf{x}) \left(\frac{m-1}{n^{(m-1)}} \frac{(m-c^{(m)})\rho}{m-1} \right)^{|\mathbf{x}|} \\
&= \sum_{A \subset F^{(m-1)}} G^{(m-1)}_A \left(\boldsymbol{\rho}'^{(m-1)} \right),
\end{aligned}$$

where,

$$\boldsymbol{\rho}'^{(m-1)} \triangleq \left(\rho'^{(m-1)}_i \triangleq \frac{m-1}{n^{(m-1)}} \frac{(m-c^{(m)})\rho}{m-1} : i \in F^{(m-1)} \right).$$

Thus, in turn,

$$\begin{aligned}
\sum_{A \in \mathcal{T}^{(m)}_{s_2,0}} G^{(m)}_A(\boldsymbol{\rho}^{(m)}) &= \sum_{A \subset F^{(m-1)}} G^{(m-1)}_A \left(\boldsymbol{\rho}'^{(m-1)} \right) \\
&= G^{(m-1)} \left(\boldsymbol{\rho}'^{(m-1)} \right)
\end{aligned}$$

Using similar arguments, one can show that

$$\sum_{A \in \mathcal{T}^{(m)}_{s_1,b_1} \cap \mathcal{T}^{(m)}_{s_2,0}} G^{(m)}_A(\boldsymbol{\rho}^{(m)}) = \sum_{A \in \mathcal{T}^{(m-1)}_{s_1,b_1}} G^{(m-1)}_A \left(\boldsymbol{\rho}'^{(m-1)} \right)$$

Combining above equalities, we get,

$$\frac{\sum_{A \in \mathcal{T}^{(m)}_{s_1,b_1} \cap \mathcal{T}^{(m)}_{s_2,0}} G^{(m)}_A(\boldsymbol{\rho}^{(m)})}{\sum_{A \in \mathcal{T}^{(m)}_{s_2,0}} G^{(m)}_A(\boldsymbol{\rho}^{(m)})} = \frac{\sum_{A \in \mathcal{T}^{(m-1)}_{s_1,b_1}} G^{(m-1)}_A \left(\boldsymbol{\rho}'^{(m-1)} \right)}{G^{(m-1)} \left(\boldsymbol{\rho}'^{(m-1)} \right)}$$

$$= Pr\left(Y_{s_1}^{(m-1,\frac{(m-c^{(m)})\rho}{m-1})} = b_1\right)$$
$$= \left(1 - \frac{(m-c^{(m)})\rho}{m-1}\right)\mathbf{1}_{\{b_1=0\}} + \frac{(m-c^{(m)})\rho}{m-1}\mathbf{1}_{\{b_1=1\}}$$

By substituting this in (5.6), using law of total probability, and taking the limit as $m \to \infty$, we get,

$$Pr\left(Y_{s_1}^{(m,\rho)} = b_1, Y_{s_2}^{(m,\rho)} = b_2\right) \overset{m\to\infty}{\longrightarrow} \prod_{i=1}^{2}\left((1-\rho)\mathbf{1}_{\{b_i=0\}} + \rho\mathbf{1}_{\{b_i=1\}}\right)$$

Thus, the theorem. □

In next section, we consider engineering implications of this result.

5.3 Impact of a shared network link and power capacity

In previous section, we showed that the total number of active servers concentrates close to its mean. In this section we study its implication for provisioning of the peak power capacity and/or of a shared network link for such large scale systems.

Several modern systems are designed so that the power usage of a server is low when it is inactive [3, 35]. Thus, the total instantaneous power draw in such systems is an increasing function of the total number of active servers. Thus, a concentration in servers' activity implies that the peak power draw is unlikely to be significantly away from the mean power consumption. This allows one to reduce infrastructure costs by provisioning for a peak power capacity which is close to the average power requirement without a significant risk of overload.

Similarly, for centralized content delivery systems where the servers are collocated and are connected to the users via a shared network link, see Fig. 1.2, a concentration in servers' activity facilitates provisioning of the network link. In such systems, the total number of active servers is proportional to the overall network traffic volume. Intuitively, Theorem 12 implies that if the bandwidth of the shared network link is greater than $(1+\epsilon)\rho m$, the link may cease to become a bottleneck as m becomes large. Below, we make this intuition more precise.

Consider a content delivery system where the bandwidth the shared network link is $\beta^{(m)}$. Thus, the maximum sum-rate at which bits may be downloaded from the servers is $\beta^{(m)}$. In this section we assume that if upon arrival of a job the service capacity $\mu^{(m)}\left(A_{\hat{\mathbf{X}}^{(m)}}(t)\right)$ exceeds $\beta^{(m)}$ then the job is blocked, where $\hat{\mathbf{X}}^{(m)}(t)$ represents the number of jobs in the modified system. Thus, $\hat{\mathbf{X}}^{(m)}(t)$ is restricted to remain within the following set:

$$\mathcal{A}^{(m)} \triangleq \left\{\mathbf{x} : \mu^{(m)}(A_{\mathbf{x}}) \leq \beta^{(m)}\right\}.$$

For such a system, the stationary distribution $\hat{\mathbf{X}}^{(m)}$, namely $\hat{\pi}^{(m)}(.)$, is a truncated version of $\pi^{(m)}(.)$ (see (5.3) for definition of $\pi^{(m)}(.)$). Indeed this follows since $\pi^{(m)}(.)$ satisfies detailed balance conditions, see [28]. Thus, $\hat{\pi}^{(m)}(.)$ can be given as follows: for each $\mathbf{x}$,

$$\hat{\pi}^{(m)}(\mathbf{x}) = \frac{\pi^{(m)}(\mathbf{x})\mathbf{1}_{\left\{\mathbf{x}\in\mathcal{A}^{(m)}\right\}}}{\sum_{\mathbf{x}\in\mathcal{A}^{(m)}} \pi^{(m)}(\mathbf{x})}. \tag{5.7}$$

A class i arrival gets blocked if it sees a state $\mathbf{x}$ in the following set:

$$\mathcal{B}_i \triangleq \{\mathbf{x} \in \mathcal{A}^{(m)} : \mathbf{x} + \mathbf{e}_i \notin \mathcal{A}^{(m)}\}.$$

By PASTA, the probability that a class i arrival gets blocked is given by:

$$p_i^{(m)} = \sum_{\mathbf{x} \in \mathcal{B}_i} \hat{\pi}^{(m)}(\mathbf{x}).$$

Let

$$\mathcal{B} \triangleq \{\mathbf{x} \in \mathcal{A}^{(m)} : \mu^{(m)}(A_{\mathbf{x}}) > \beta^{(m)} - c^{(m)}\}$$

and note that for each i we have $\mathcal{B}_i \subset \mathcal{B}$. It follows that

$$p_i^{(m)} \leq \sum_{\mathbf{x} \in \mathcal{B}} \hat{\pi}^{(m)}(\mathbf{x}).$$

Now, suppose that $\beta^{(m)}$ scales as $(1+\epsilon)\rho m$ for some $\epsilon > 0$. Then, from Theorem 12, the denominator in (5.7), namely $\sum_{\mathbf{x} \in \mathcal{A}^{(m)}} \pi^{(m)}(\mathbf{x})$, tends to 1 as $m \to \infty$. Thus, the impact of truncation by network capacity becomes negligible as the system becomes large. More formally, the theorem below follows by noting that $c^{(m)}$ is $o(m)$.

Theorem 13. *Consider a sequence of content delivery systems as in Theorem 12. Further suppose that the servers are connected to the users via a shared network link of bandwidth $\beta^{(m)} = (1+\epsilon)\rho m$ for some $\epsilon > 0$. Suppose if upon arrival of a job the aggregate service capacity $\mu^{(m)}(A_{\mathbf{X}^{(m)}})$ exceeds $\beta^{(m)}$ then the job is blocked.*

For each m, the system is stationary. Under stationary distribution, let $\hat{Y}_s^{(m)}$ represent the instantaneous activity of server s. Let $p_i^{(m)}$ be the probability that a class i job is blocked. Then, the following statements hold:

(a) $\lim_{m\to\infty} \sup_i p_i^{(m)} \to 0.$

(b) $\lim_{m\to\infty} \frac{\sum_{l=1}^{m} \hat{Y}_{s_l}^{(m)}}{m} = E\left[\hat{Y}_{s_1}^{(m)}\right]$ *in probability.*

5.4 Impact of peak rate constraints

In this section we incorporate peak rate constraints in our system model and analysis to investigate how they impact user performance. Such constraints can model either finite download capacity at the end user or be a result of end-to-end flow-control mechanism such as TCP transmission which has a finite window size which limits a user's peak rate.

Recall that we let q_F denote the set of all jobs currently in the system. Let $\mathbf{b} = (b_u : u \in q_F)$ where b_u is the rate at which job u is being served. For each job u in the system, let $c(u)$ represent its class, and let $c(U) = \cup_{u \in U} c(u)$ for any subset $U \subset q_F$. Let $\gamma(u)$ denote the peak rate constraint on job u. Clearly, the peak rate constraint on each job can affect the rates at which other jobs will be served. We say that a bandwidth allocation $\mathbf{b}$ is feasible if both the server constraints (i.e., the sum-rate constraints) $\sum_{u:\text{ s.t. } c(u)\in A} b_u \leq \mu(A)$ for each $A \in F$ and the peak rate constraints $b_u \leq \gamma(u)$ for each $u \in q_F$ are satisfied.

We now characterize the capacity region, i.e., the set of all feasible resource allocations $\mathbf{b}$ and show that it has a polymatroid structure. Here we take an approach which is slightly different than that of Sec. 2.2 where the capacity region $\mathcal{C}$ was defined as the set of feasible rates allocated to each class $i \in F$, i.e., the set of feasible $\mathbf{r} = (r_i : i \in F)$. Instead, since each job u entering the system can potentially have its own peak rate constraint $b_u \leq \gamma(u)$, we first characterize the 'per-job capacity region' $\mathcal{P}$ as the set of all feasible $\mathbf{b}$, and then characterize the corresponding per-class capacity region. The following result characterizes the the 'per-job capacity region' $\mathcal{P}$.

Theorem 14. *Consider a system where sum-rate constraints across the classes are given by a rank function $\mu(.)$ and the peak constraints for jobs given by the $\gamma(.)$. At time t suppose q_F is the set of jobs in the system. Let*

$$\mathcal{P} = \left\{ b \geq 0 : \sum_{u \in U} b_u \leq \nu(U), \forall U \in q_F \right\},$$

where,

$$\nu(U) = \min_{A \subset c(U)} \left\{ \mu(A) + \sum_{\substack{u \in U \\ s.t.\ c(u) \notin A}} \gamma(u) \right\}.$$

Then, 1.) $\nu(.)$ is a rank function, and 2.) $\mathcal{P}$ is a polymatroid and capacity region associated with the jobs.

Proof. We first prove that ν is submodular. From the definition it is clear that $\nu(\emptyset) = 0$ and that it is monotonic. Thus, we only need to check that ν is submodular. Consider sets $U, V \subset q_F$. From the definition of ν, there exist sets $A \subset c(U)$ and $B \subset c(V)$ such that

$$\nu(U) = \mu(A) + \sum_{u \in U \text{ s.t. } c(u) \notin A} \gamma(u),$$

$$\nu(V) = \mu(B) + \sum_{u \in V \text{ s.t. } c(u) \notin B} \gamma(u)$$

and

$$\nu(U \cap V) \leq \mu(A \cap B) + \sum_{u \in U \cap V \text{ s.t. } c(u) \notin A \cap B} \gamma(u).$$

Thus,

$$\begin{aligned}
\nu(U) + \nu(V) - \nu(U \cap V) &\geq \mu(A) + \mu(B) - \mu(A \cap B) \\
&\quad + \sum_{\substack{u \in U \\ \text{s.t. } c(u) \notin A}} \gamma(u) + \sum_{\substack{u \in V \\ \text{s.t. } c(u) \notin B}} \gamma(u) - \sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A \cap B}} \gamma(u) \\
&\geq \mu(A \cup B) + \sum_{\substack{u \in U \setminus V \\ \text{s.t. } c(u) \notin A}} \gamma(u) + \sum_{\substack{u \in V \setminus U \\ \text{s.t. } c(u) \notin B}} \gamma(u) \\
&\quad + \left(\sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A}} \gamma(u) + \sum_{\substack{u \in V \cap U \\ \text{s.t. } c(u) \notin B}} \gamma(u) - \sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A \cap B}} \gamma(u) \right)
\end{aligned}$$

where the last inequality follows from the submodularity of μ and partitioning $\sum_{\substack{u \in U \\ \text{s.t. } c(u) \notin A}} \gamma(u)$ and $\sum_{\substack{u \in V \\ \text{s.t. } c(u) \notin B}} \gamma(u)$ into $\sum_{\substack{u \in U \setminus V \\ \text{s.t. } c(u) \notin A}} \gamma(u) + \sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A}} \gamma(u)$ and $\sum_{\substack{u \in V \setminus U \\ \text{s.t. } c(u) \notin B}} \gamma(u) + \sum_{\substack{u \in V \cap U \\ \text{s.t. } c(u) \notin B}} \gamma(u)$, respectively. It can be checked that the term inside parenthesis in the last inequality is equal to $\sum_{u \in U \cap V \text{ s.t. } c(u) \notin A \cup B} \gamma(u)$.

Thus,

$$\begin{aligned}
\nu(U) + \nu(V) - \nu(U \cap V) &\geq \mu(A \cup B) + \sum_{\substack{u \in U \setminus V \\ \text{s.t. } c(u) \notin A \cup B}} \gamma(u) + \sum_{\substack{u \in V \setminus U \\ \text{s.t. } c(u) \notin B \cup A}} \gamma(u) \\
&\quad + \sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A \cup B}} \gamma(u) \\
&\geq \mu(A \cup B) + \sum_{\substack{u \in U \setminus V \\ \text{s.t. } c(u) \notin A \cup B}} \gamma(u) + \sum_{\substack{u \in V \setminus U \\ \text{s.t. } c(u) \notin B \cup A}} \gamma(u) + \sum_{\substack{u \in U \cap V \\ \text{s.t. } c(u) \notin A \cup B}} \gamma(u),
\end{aligned}$$

since $\{u \in U \setminus V : c(u) \notin A\} = \{u \in U \setminus V : c(u) \notin A \cup B\}$ as $B \subset c(V)$, and similarly $\{u \in V \setminus U : c(u) \notin B\} = \{u \in V \setminus U : c(u) \notin B \cup A\}$. Thus,

$$\nu(U) + \nu(V) - \nu(U \cap V) = \mu(A \cup B) + \sum_{\substack{u \in U \cup V \\ \text{s.t. } c(u) \notin A \cup B}} \gamma(u) \geq \nu(U \cup V)$$

Thus, ν is submodular.

We now show that every point in $\mathcal{P}$ is achievable. By definition of ν, $\nu(\{u\}) \leq \gamma(u)$ and $\nu(\{u \in F : c(u) \in A\}) \leq \mu(A)$. Suppose $\mathbf{b} \in \mathcal{P}$. Then, by definition of $\mathcal{P}$, $b_u \leq \nu(\{u\})$ and $\sum_{u \in F \text{ s.t. } c(u) \in A} b_u \leq \nu(\{u \in F : c(u) \in A\})$ which implies that the peak constraints as well as the sum-rate constraints are satisfied. Hence, $\mathbf{b}$ is feasible.

Further, if $\mathbf{b} \geq 0$ and $\mathbf{b} \notin \mathcal{P}$, then there exists U such that $\sum_{u \in U} b_u \geq \nu(U)$. By definition of $\nu(U)$, there exist $A \subset c(U)$ such that $\sum_{u \in U} b_u \geq \mu(A) + \sum_{u \in U \text{ s.t. } c(u) \notin A} \gamma(u)$. Thus, we either have $\sum_{u \in U \text{ s.t. } c(u) \in A} b_u \geq \mu(A)$ or we have $\sum_{u \in U \text{ s.t. } c(u) \notin A} b_u \geq \sum_{u \in U \text{ s.t. } c(u) \notin A} \gamma(u)$. So, either a capacity constraint or a peak constraint is violated. Hence, the result. □

Next, let us characterize the per-class capacity region. With some loss of generality, suppose that for each job $u \in q_F$, $\gamma(u) = \beta_i$ if $c(u) = i$. Now, for a given $\mathbf{x}$, the resource allocation $\mathbf{r}(\mathbf{x})$ is feasible if it satisfies the sum-rate constrains $\sum_{i \in A} r_i(\mathbf{x}) = \mu(A)$ for all $A \subset F$ as well as the peak rate constraints $r_i(\mathbf{x}) \leq \beta_i x_i$. Now, since each job brings an additional constraint into the system, the per-class capacity region itself is a function of $\mathbf{x}$. To make this fact explicit, we shall denote the capacity region by $\tilde{\mathcal{C}}(\mathbf{x})$ for the rest of this section, where $\tilde{\mathcal{C}}(\mathbf{x})$ can be given as:

$$\tilde{\mathcal{C}}(\mathbf{x}) = \left\{ \mathbf{r} : \sum_{i \in A} r_i = \mu(A), \text{ and } r_i \leq \beta_i x_i \right\}.$$

One can check that

$$\tilde{\mathcal{C}}(\mathbf{x}) = \left\{ \mathbf{r} : \exists \mathbf{b} \in \mathcal{P} \text{ s.t. } x_i = |\{u : c(u) = i \;\&\; b_u > 0\}| \;\&\; \sum_{u : c(u) = i} b_u = r_i, \; \forall i \right\}.$$

We now study balanced fair resource allocation for such systems. Following analysis similar to that in Sec. 2.3, the balance function $\tilde{\Phi}(\mathbf{x})$ for any $\mathbf{x}$ with peak rate constraints is recursively defined as $\tilde{\Phi}(\mathbf{0}) = \mathbf{1}$ and $\tilde{\Phi}(\mathbf{x}) = 0$ $\forall \mathbf{x}$ s.t. $x_i < 0$ for some i, otherwise,

$$\tilde{\Phi}(\mathbf{x}) = \sup \left\{ \delta^{-1} : \mathbf{r} = (\delta \tilde{\Phi}(\mathbf{x} - \mathbf{e}_i) : i \in F) \in \tilde{\mathcal{C}}(\mathbf{x}) \right\}.$$

Or equivalently,

$$\tilde{\Phi}(\mathbf{x}) = \max \left\{ \max_{A \subset F} \left\{ \frac{\sum_{i \in A} \tilde{\Phi}(\mathbf{x} - \mathbf{e}_i)}{\mu(A)} \right\}, \max_{\substack{i \in F \\ \text{s.t. } x_i > 0}} \left\{ \frac{\tilde{\Phi}(\mathbf{x} - \mathbf{e}_i)}{x_i \beta_i} \right\} \right\}.$$

In other words, we recursively choose the largest $\tilde{\Phi}(\mathbf{x})$ such that $\mathbf{r}(\mathbf{x})$ as given by (5.1) is as large as possible while being feasible. We now exhibit how balanced fairness allocates rates to the individual jobs $u \in q_F$. Recall that for each u, $b_u = \frac{r_i(\mathbf{x})}{x_i}$ where $i = c(u)$. Under balanced fairness one can think of b_u in the light of following lemma.

Lemma 17. *Consider a set function Ψ be such that $\Psi(\emptyset) = 1$ and for each set of jobs U*

$$\Psi(U) = \max_{V \subset U} \frac{\sum_{u \in V} \Psi(U \backslash u)}{\nu(V)}.$$

Then, for any U, balanced fair resource allocation for each job $u \in U$ is given by

$$b_u = \frac{\Psi(U \backslash u)}{\Psi(U)},\ \forall u \in U.$$

Proof. We will first show that

$$\Psi(U) = (\Pi_i x_i!)\, \tilde{\Phi}(\mathbf{x}) \text{ where } x_i = |\{u \in U : c(u) = i\}|, \tag{5.8}$$

by using induction on $|U|$. Clearly, (5.8) holds for $|U| = 1$. Suppose it holds for $|U| = k - 1$. Then, consider U such that $|U| = k$. Suppose $x_i = |\{u \in U : c(u) = i\}|$. Then, from the first part of Theorem 14,

$$\begin{aligned}
&(\Pi_i x_i!)\, \tilde{\Phi}(\mathbf{x}) \\
&= (\Pi_i x_i!) \sup \left\{ \delta^{-1} : \mathbf{r} = (\delta \tilde{\Phi}(\mathbf{x} - \mathbf{e}_i) : i \in F) \text{ is feasible } \right\} \\
&= (\Pi_i x_i!) \sup \left\{ \delta^{-1} : (b_u = \delta \tilde{\Phi}(\mathbf{x} - \mathbf{e}_{c(u)})/x_{c(u)} : u \in U) \in \mathcal{P} \right\} \\
&= \sup \left\{ \delta^{-1} : (b_u = \delta\, (\Pi_i x_i!)\, \tilde{\Phi}(\mathbf{x} - \mathbf{e}_{c(u)})/x_{c(u)} : u \in U) \in \mathcal{P} \right\} \\
&= \sup \left\{ \delta^{-1} : (b_u = \delta \Psi(U \backslash u) : u \in U) \in P \right\} \\
&= \max_{V \subset U} \frac{\sum_{u \in V} \Psi(U \backslash u)}{\nu(V)} \\
&= \Psi(U)
\end{aligned}$$

Thus, (5.8) holds for all U. Now, for a given set of ongoing jobs V, let $x_j = |\{u \in V : c(u) = j\}$ for each $j \in F$. Then, for each $u \in V$ such that $c(u) = i$, we have that

$$b_u = \frac{r_i(\mathbf{x})}{x_i} = \frac{\tilde{\Phi}(\mathbf{x} - \mathbf{e}_i)}{x_i \tilde{\Phi}(\mathbf{x})} = \frac{\Psi(U \backslash u)/((\mathbf{x}_i - 1)! \Pi_{j \neq i} x_j!)}{x_i \Psi(U)/(\Pi_j x_j!)},$$

where the last equality follows from (5.8). The result thus follows directly from the above expression. □

Thus, one can equivalently think of the corresponding resource allocation for each job as balanced fair resource allocation on $\mathcal{P}$. Further, since $\mathcal{P}$ is a polymatroid, Theorem 2 implies Pareto optimality of balanced fair resource allocation under peak rate constraints, so it follows that

$$\Psi(U) = \frac{\sum_{u \in U} \Psi(U \backslash u)}{\nu(U)}. \tag{5.9}$$

Next we provide a recursive expression for the normalization constant $\tilde{G}(\boldsymbol{\rho}) = \sum_{\mathbf{x}} \tilde{\Phi}(\mathbf{x}) \prod_{i \in F} \rho_i^{x_i}$ along the lines of Theorem 4. An expression for mean delay for each class i, follows since $E[D_i] = \frac{\nu_i \frac{\partial}{\partial \rho_i} \tilde{G}(\boldsymbol{\rho})}{\tilde{G}(\boldsymbol{\rho})}$. To develop recursions, the dependence of the normalization constant on the underlying capacity region which results from ignoring peak rate constraints, namely, $\mathcal{C} = \left\{ \mathbf{r} : \sum_{i \in A} r_i \leq \mu(A),\ \forall A \subset F \right\}$ becomes important. Thus, from now on, we represent normalization constant as $\tilde{G}(\mathcal{C})$. Before we provide an explicit expression for $\tilde{G}(\mathcal{C})$ (see Theorem 15 below), we need some additional notation:

For capacity region $\mathcal{C}$, let

$$\Omega(\mathcal{C}) = \left\{ \mathbf{x} : \sum_{i \in A} x_i \beta_i \leq \mu(A), \forall A \subset F \right\}.$$

This can be viewed as state space of a loss system where a loss (blocking of an arrival) happens if upon arrival the service rate $\sum_{i \in A} x_i \beta_i$ allocated to a subset of classes exceeds the capacity $\mu(A)$ associated with it. Let

$$\tilde{G}_{\emptyset}(\mathcal{C}) = \sum_{\mathbf{x} \in \Omega(C)} \prod_i \frac{1}{x_i!} \left(\frac{\rho_i}{\beta_i} \right)^{x_i},$$

which can be interpreted as the normalization constant for the above mentioned loss system.

Further, let

$$\mathcal{C}_A = \{\mathbf{r} \in \mathcal{C} : r_i = 0,\ \forall i \notin A\}.$$

$$\mathcal{C}_{-A} = \left\{\mathbf{r} : \sum_{i\in B} r_i \leq \mu(A \cup B) - \mu(B), \quad \forall B \subset F\backslash A\right\}.$$

Intuitively, $\mathcal{C}_{-A}$ can be viewed as follows: remove all the servers from the system that serve classes in A. For each $B \subset F\backslash A$, serve the corresponding jobs with the remaining servers. The associated capacity region is denoted as $\mathcal{C}_{-A}$. Also, let $\tilde{G}_\emptyset(\mathcal{C}_A)$ be the normalization constant associated with corresponding loss system. Further, for $B \subset A$, we define

$$\mathcal{C}_{A-B} = \left\{\mathbf{r} \in \mathcal{C}_A : \sum_{i\in B'} r_i \leq \mu(B \cup B') - \mu(B'), \quad \forall B' \subset A\backslash B\right\}.$$

Also, we shall let

$$\mathcal{C}^{(i)} = \{\mathbf{r} \in \mathcal{C} : \mathbf{r} + \beta_i \mathbf{e}_i \notin \mathcal{C}\},$$

or equivalently,

$$\mathcal{C}^{(i)} = \left\{\mathbf{r} : \sum_{j\in A} r_j \leq \mu(A) - \beta_i \mathbf{1}_{\{i\in A\}}, \forall A \subset F\right\}.$$

Finally, let $L_i(\mathcal{C})$ be the probability of blocking a class i job in a loss system associated with capacity region $\mathcal{C}$ due to violation of the constraint $\sum_{i\in F} x_i\beta_i \leq \mu(F)$. This can be given as:

$$L_i(\mathcal{C}) = 1 - \frac{\tilde{G}_\emptyset(\mathcal{C}^{(i)})}{\tilde{G}_\emptyset(\mathcal{C})} - \sum_{A \subsetneq F : i\in A} \frac{\tilde{G}_\emptyset(\mathcal{C}_{-A})\tilde{G}_\emptyset(\mathcal{C}_A)}{\tilde{G}_\emptyset(\mathcal{C})} L_i(\mathcal{C}_A).$$

The following expression for the normalization constant can be developed via a similar approach as that used for that for wireline networks with tree topology in [11]. Recall, tree topology in networks exhibits a capacity region which is a special case of polymatroids.

Theorem 15. *The recursion for $\tilde{G}(\mathcal{C})$ is given as follows:*

$$\tilde{G}(\mathcal{C}) = \sum_{A \subset F} \tilde{G}_A(\mathcal{C}),$$

where for each $A \subset F$, we have

$$\tilde{G}_A(\mathcal{C}) = \tilde{G}_\emptyset(\mathcal{C}_{-A}) H_A(\mathcal{C}_A),$$

where

$$H_A(\mathcal{C}_A) = \frac{\sum_{B \subsetneq A} \tilde{G}_B(\mathcal{C}_A) \sum_{i \in F \setminus A} \rho_i L_i(\mathcal{C}_{A-B})}{\mu(A) - \sum_{i \in A} \rho_i}.$$

The above expression for normalization constant is complex; we thus resort to symmetry and heuristics to gain insights on the impact of peak rate constraints on mean delays.

5.4.1 Asymptotic analysis of symmetric peak rate constrained systems

In this section we study the asymptotic performance of systems with peak rate constraints in a regime where overall system load and number of servers increase proportionally. We assume symmetry in arrival rates across classes, in mean service requirements, and in peak rate constants. Let the overall arrival rate to the system be λm, thus arrival rate for each class is

$\lambda_i = \lambda m/n$. The mean service requirement for each class ν is a constant. Let the peak rate constraints for each class i be $\beta_i = \beta$, i.e., they are homogeneous. Further, we assume that the capacity region associated with the system when the peak rate constraints are relaxed, namely $\mathcal{C}$, is symmetric. Recall that when peak rate constraints are included, the associated capacity region depends on $\mathbf{x}$, and is denoted by $\tilde{\mathcal{C}}(\mathbf{x})$ which may not be symmetric.

As in Section 2.4.1, we consider a limiting regime where first the number of classes n increases to ∞ for a fixed m, and then the number of servers m increases to ∞. For a finite system, the dependence on m and n is exhibited via the superscript (m, n). The polymatroid capacity region $\mathcal{C}^{(m,n)}$ is also assumed to be same as that in Section 2.4.1, where it was obtained via randomized replication of files across c servers and averaging of the resulting random capacity region. Recall that the associated symmetric rank function is given as $\bar{\mu}^{(m,n)}(A) = h^{(m,n)}(|A|)$ where

$$h^{(m,n)}(k) = \xi m(1 - (1 - c/m)^k) \text{ for } k = 0, 1, \ldots, n.$$

The main difference between the asymptotic analysis in Section 2.4.1 and that developed below is that we now include peak rate constraints for each job. Note that if $\beta \geq c\xi$ then peak rate constraints are redundant since $c\xi$ is the maximum service rate a job can get from the servers. Thus, we assume that $\beta < c\xi$.

To aid our asymptotic analysis, let us first summarize the key insights of our proof for Theorem 4 for a system without peak rate constraints:

1. Due to thinning of the load per class while keeping the load per server constant, the probability of having more than one active job in any given class tends to zero asymptotically.

2. The dynamics for the total number of active jobs $|\mathbf{x}|$ can be approximated by a birth-death process with birth rate λm and death rate $\frac{1}{\nu} h^{(m,n)}(|\mathbf{x}|)$.

3. As the system size scales there is concentration in probability measure of the birth-death process across states $\mathbf{x}$ such that $h^{(m,n)}(|A_{\mathbf{x}}|) \approx \rho m$.

We now use these insights to include the peak rate constraints into the asymptotic analysis. Due to symmetry and Pareto optimality, with high probability, all jobs are either bottlenecked by peak rate constraints or by the capacity region $\mathcal{C}^{(m,n)}$ depending on the value of $|\mathbf{x}|$. In other words, the dynamics for $|\mathbf{x}|$ can now be approximated by a birth-death process with birth rate λm and death rate $\frac{1}{\nu} \min\left(\beta|\mathbf{x}|, h^{(m,n)}(|\mathbf{x}|)\right)$. Using the fact that $h^{(m,n)}(.)$ is concave and that $\beta < c\xi$, one can show that there exists a threshold such that for values of $|\mathbf{x}|$ below the threshold the system is bottlenecked by peak rate constraints, and otherwise by server capacities.

For such a birth-death process, the following proposition can be proven along the lines of the proof for Theorem 4.

Proposition 5. *Consider a sequence of birth-death processes $\left(\mathcal{B}^{(m)} : m \in \mathbb{N}\right)$ where $\mathcal{B}^{(m)}$ has, in each state k, the birth rate equal to λm and death rate equal to $\frac{1}{\nu} \min\left(\beta k, h^{(m,\infty)}(k)\right)$ where $h^{(m,\infty)}(k) = \xi m(1 - (1 - c/m)^k)$, and β, c, and*

ξ are positive constants such that $c > 1$ and $\beta < c\xi$. Let $\rho = \lambda\nu$. Suppose that $\rho < \xi$. Then $\mathcal{B}^{(m)}$ is stationary for each m. Let $\pi_k^{(m)}$ denote the stationary probability for each state k.

Further, let

$$\tau^* = \frac{\xi}{\beta} + \frac{1}{c} W\left(-\frac{c\xi}{\beta} e^{-c\xi/\beta}\right)$$

where $W(.)$ is the principle branch of standard Lambert W function [16]. Also let

$$\alpha^* = \mathbf{1}_{\{\rho<\tau^*\}} \frac{\rho}{\beta} + \frac{1}{c} \log\left(\frac{1}{1-\rho/\xi}\right) \mathbf{1}_{\{\rho\geq\tau^*\}}.$$

Then, for each $\epsilon > 0$, we have:

$$\lim_{m\to\infty} \sum_{k=\lfloor\alpha^* m(1-\epsilon)\rfloor}^{\lfloor\alpha^* m(1+\epsilon)\rfloor} \pi_k^{(m)} = 1. \tag{5.10}$$

The intuition behind this result is given in following three steps:

1. Notice that $\lim_{m\to\infty} \frac{1}{m} h^{(m,\infty)}(\tau m) = \xi(1 - e^{-c\tau})$. Also, using definition of the standard Lambert W function, one can show that τ^* is a solution to the following transcendental equation:

$$\beta\tau = \xi(1 - e^{-c\tau}). \tag{5.11}$$

   Thus, intuitively, $\tau^* m$ captures the threshold where the bottleneck transitions from peak rate constraints to server capacities.

2. For states k which are close to $\alpha^* m$, birth rate is approximately equal to the death rate.

3. There is concentration in $\pi_k^{(m)}$ at states which are close to $\alpha^* m$.

With an application of Little's law, the above proposition suggests following heuristic expression for mean delays for large symmetric systems with peak rate constraints:

$$E[D] = \mathbf{1}_{\{\rho<\tau^*\}} \frac{\rho}{\lambda\beta} + \frac{1}{c\lambda} \log\left(\frac{1}{1-\rho/\xi}\right) \mathbf{1}_{\{\rho\geq\tau^*\}}.$$

Thus, for large systems, when the overall system load is low the peak rate constraints drive the user-performance. However, as the load increases, they cease to be a dominant bottleneck.

Notice the equilibrium condition (5.11) between the impact of peak rate constraints and of server capacities. Similar equilibrium conditions appear in certain problems of epidemics and giant components in random graphs [16,19]. It would be interesting to further explore connections of our model to these problems.

Chapter 6

Conclusions

Our main conclusions address both theoretical and practical aspects associated with the design of content delivery systems aimed at serving large files. Our results show that infrastructure which allows a user to download in parallel from a pool of servers can achieve negligible download delays under limited heterogeneity in file demands. Some elements of content delivery infrastructure may see less pronounced heterogeneity in demands, e.g., a centralized back end used to deliver files that are not available at distributed sites/caches. Our result suggests a scalable approach towards delivering content for such centralized systems without requiring complex caching strategies internally.

On the theoretical side we have established: (i) basic new results linking fairness in resource allocation to delays, (ii) the asymptotic symmetry of randomly configured large-scale systems with heterogenous components, (iii) a fundamental result linking concentration in servers' activity to scaling in the size of interacting server pools. Together these results suggest large systems might eventually be robust to heterogeneity and even the fairness criterion.

Bibliography

[1] D. J. Aldous. *Exchangeability and related topics.* Springer, 1985.

[2] F. Baccelli, F. Karpelevich, M. Kelbert, A. Puhalskii, A. Rybko, and Y. Suhov. A mean-field limit for a class of queueing networks. *Journal of Statistical Physics*, 66(3-4):803–825, 1992.

[3] L. Barroso and U. Hölzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, 4(1):1–108, 2009.

[4] T. Bonald. Throughput performance in networks with linear capacity contraints. In *Proceedings of CISS*, pages 644 –649, 2006.

[5] T. Bonald and L. Massoulié. Impact of fairness on internet performance. In *Proceedings of ACM Sigmetrics*, pages 82–91, 2001.

[6] T. Bonald, L. Massoulié, A. Proutière, and J. Virtamo. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems*, 53:65–84, 2006.

[7] T. Bonald and A. Proutière. Insensitive bandwidth sharing in data networks. *Queueing Systems*, 44:69–100, 2003.

[8] T. Bonald and A. Proutière. On performance bounds for the integration of elastic and adaptive streaming flows. In *Proceedings of ACM Sigmetrics*, pages 235–245, 2004.

[9] T. Bonald and A. Proutière. On stochastic bounds for monotonic processor sharing networks. *Queueing Systems*, 47:81–106, 2004.

[10] T. Bonald, A. Proutière, J. Roberts, and J. Virtamo. Computational aspects of balanced fairness. In *Proceedings of ITC*, 2003.

[11] T. Bonald and J. Virtamo. Calculating the flow level performance of balanced fairness in tree networks. *Perform. Eval.*, 58(1):1–14, Oct. 2004.

[12] M. Bramson, Y. Lu, and B. Prabhakar. Randomized load balancing with general service time distributions. In *Proceedings of the ACM Sigmetrics*, pages 275–286, 2010.

[13] M. Bramson, Y. Lu, and B. Prabhakar. Asymptotic independence of queues under randomized load balancing. *Queueing Systems*, 71(3):247–292, 2012.

[14] A. Cidon, S. Rumble, R. Stutsman, S. Katti, J. Ousterhout, and M. Rosenblum. Copysets: Reducing the frequency of data loss in cloud storage. In *Usenix Advanced Technical Conference*, 2013.

[15] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[16] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.

[17] G. de Veciana, T.-J. Lee, and T. Konstantopoulos. Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Transactions on Networking*, 9(1):2–14, Feb. 2001.

[18] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *Information Theory, IEEE Transactions on*, 56(9):4539–4551, Sept 2010.

[19] M. Draief and L. Massoulié. *Epidemics and Rumours in Complex Networks*. Cambridge University Press, 1st edition, 2010.

[20] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(2):99–124, 1998.

[21] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Proceedings of Calgary International Conference on Combinatorial Structures and Applications*, pages 69–87, 1969.

[22] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Perform. Eval.*, 67(11):1155–1171, Nov. 2010.

[23] C. Gkantsidis, J. Miller, and P. Rodriguez. Comprehensive view of a live network coding p2p system. In *Proceedings of ACM SIGCOMM*, pages 177–188, New York, NY, USA, 2006.

[24] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Trans. Netw.*, 14(6):1260–1271, Dec. 2006.

[25] V. Joseph and G. de Veciana. Stochastic networks with multipath flow control: Impact of resource pools on flow-level performance and network congestion. In *Proceedings of the ACM Sigmetrics*, pages 61–72, 2011.

[26] O. Kallenberg. Canonical representations and convergence criteria for processes with interchangeable increments. *Zeitschrift fr Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 27(1):23–36, 1973.

[27] F. Kelly, L. Massoulié, and N. Walton. Resource pooling in congested networks: proportional fairness and product form. *Queueing Systems*, 63(1-4):165–194, 2009.

[28] F. P. Kelly. *Reversibility and Stochastic Networks*. Wiley, 1979.

[29] F. P. Kelly. Loss networks. *Ann. Appl. Probab.*, 1(3):319–378, 08 1991.

[30] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *The Journal of the Operational Research Society*, 49(3):237–252, 1998.

[31] P. Key, L. Massoulié, and D. Towsley. Path selection and multipath congestion control. *Commun. ACM*, 54(1):109–116, Jan. 2011.

[32] T. Lan, D. Kao, M. Chiang, and A. Sabharwal. An axiomatic theory of fairness in network resource allocation. In *Proceedings of IEEE Infocom*, pages 1–9, March 2010.

[33] M. Leconte, M. Lelarge, and L. Massoulié. Bipartite graph structures for efficient balancing of heterogeneous loads. In *Proceedings of ACM Sigmetrics/Performance*, pages 41–52, 2012.

[34] M. Leconte, M. Lelarge, and L. Massoulié. Adaptive replication in distributed content delivery networks. *arXiv preprint arXiv:1401.1770*, 2014.

[35] M. Lin, A. Wierman, L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proceedings of IEEE Infocom*, pages 1098–1106, 2011.

[36] X. Lin and N. Shroff. Utility maximization for communication networks with multipath routing. *IEEE Transactions on Automatic Control*, 51(5):766 – 781, May 2006.

[37] A. Mandelbaum and A. L. Stolyar. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research*, 52(6):836–855, 2004.

[38] A. W. Marshall, I. Olkin, and B. C. Arnold. *Inequalities: Theory of Majorization and Its Applications.* Springer, 2nd edition, 2011.

[39] L. Massoulié. Structural properties of proportional fairness: Stability and insensitivity. *Annals of Applied Probability*, 17(3):809–839, 2007.

[40] L. Massoulié and J. Roberts. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15(1-2):185–201, 2000.

[41] M. Mitzenmacher, A. W. Richa, and R. Sitaraman. The power of two random choices: A survey of techniques and results. In P. Pardalos, S. Rajasekaran, and J. Rolim, editors, *in Handbook of Randomized Computing*, pages 255–312. Springer US, 2001.

[42] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[43] M. D. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing.* PhD thesis, University of California, Berkeley, 1996.

[44] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556 –567, Oct. 2000.

[45] S. Moharir, J. Ghaderi, S. Sanghavi, and S. Shakkottai. Serving content with unknown demand: The high-dimensional regime. In *Proceedings of ACM Sigmetrics*, pages 435–447, 2014.

[46] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley, 1988.

[47] V. Shah and G. de Veciana. Performance evaluation and asymptotics for content delivery networks. In *IEEE Infocom*, pages 2607–2615, 2014.

[48] V. Shah and G. de Veciana. High performance centralized content delivery infrastructure: Models and asymptotics. *IEEE/ACM Transactions on Networking*, 2015. To appear.

[49] V. Shah and G. de Veciana. Impact of fairness and heterogeneity on delays in large-scale content delivery networks. In *ACM Sigmetrics*, 2015.

[50] A. L. Stolyar. An infinite server system with customer-to-server packing constraints. In *Proceedings of Allerton Conference*, 2012.

[51] A. L. Stolyar. Diffusion-scale tightness of invariant distributions of a large-scale flexible service system. *Adv. in Appl. Probab.*, 47(1):251–269, Mar. 2015.

[52] A. S. Sznitman. Topics in propagation of chaos. In *Ecole d'Eté de Probabilités de Saint-Flour XIX1989*, pages 165–251. Springer, 1991.

[53] J. N. Tsitsiklis and K. Xu. Flexible queueing architectures. *arXiv preprint arXiv:1505.07648*, 2015.

[54] J. Vondrák. A note on concentration of submodular functions. *arXiv preprint arXiv:1005.2791*, 2010.

[55] N. D. Vvedenskaya. Large queueing system where messages are transmitted via several routes. *Problemy Peredachi Informatsii*, 34(2):98–108, 1998.

[56] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii*, 32(1):20–34, 1996.

[57] W. Whitt. Blocking when service is required from several facilities simultaneously. *AT&T Technical Journal*, 64(8):1807–1856, 1985.

[58] A. Wierman, L. L. H. Andrew, and A. Tang. Power-aware speed scaling in processor sharing systems: Optimality and robustness. *Perform. Eval.*, 69(12):601–622, Dec. 2012.

[59] D. Wischik, M. Handley, and M. B. Braun. The resource pooling principle. *SIGCOMM Comput. Commun. Rev.*, 38(5):47–52, Sept. 2008.

[60] K. Xu. *On the power of (even a little) flexibility in dynamic resource allocation.* PhD thesis, Massachusetts Institute of Technology, 2014.

[61] X. Yang and G. de Veciana. Performance of peer-to-peer networks: Service capacity and role of resource sharing policies. *Perform. Eval.*, 63(3):175–194, Mar. 2006.

[62] E. Yeh. *Multiaccess and fading in communication networks.* PhD thesis, Massachusetts Institute of Technology, 2001.

[63] L. Ying, R. Srikant, and X. Kang. The power of slightly more than one sample in randomized load balancing. In *Proc. of IEEE INFOCOM*, 2015.

[64] J. Zhou, Y. Li, V. K. Adhikari, and Z.-L. Zhang. Counting youtube videos via random prefix sampling. In *Proceedings of ACM Sigcomm*, pages 371–380, 2011.

[65] Y. Zhou, T. Fu, and D. M. Chiu. A unifying model and analysis of P2P VoD replication and scheduling. In *Proceedings of IEEE Infocom*, pages 1530–1538, March 2012.

[66] J. Zhu and B. Hajek. Stability of a peer-to-peer communication system. In *Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '11, pages 321–330, New York, NY, USA, 2011. ACM.

Vita

Virag Shah is a PhD candidate in Electrical and Computer Engineering department at The University of Texas at Austin. He received his B.E. degree from University of Mumbai in 2007. He received his M.E. degree from Indian Institute of Science, Bangalore in 2009. He was a Research Fellow at Indian Institute of Technology, Bombay from 2009 to 2010. His research interests include designing algorithms for content delivery systems, cloud computing systems, and internet of things; performance modeling; applied probability and queuing theory. He is a recipient of two best paper awards: IEEE INFOCOM 2014 conference at Toronto, Canada; National Conference on Communications 2010 at Chennai, India.

Permanent email: virag@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.