On Load Balancing & Routing in Peer-to-Peer Systems

by

George Giakkoupis

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

On Load Balancing & Routing in Peer-to-Peer Systems

George Giakkoupis
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2009

A *peer-to-peer (P2P)* system is a networked system characterized by the lack of centralized control, in which all or most communication is symmetric. Also, a P2P system is supposed to handle frequent arrivals and departures of nodes, and is expected to scale to very large network sizes. These requirements make the design of P2P systems particularly challenging.

We investigate two central issues pertaining to the design of P2P systems: *load balancing* and *routing*. In the first part of this thesis, we study the problem of load balancing in the context of *Distributed Hash Tables (DHTs)*. Briefly, a DHT is a giant hash table that is maintained in a P2P fashion: Keys are mapped to a hash space I — typically the interval $[0, 1)$, which is partitioned into blocks among the nodes, and each node stores the keys that are mapped to its block. Based on the position of their blocks in I, the nodes also set up connections among themselves, forming a routing network, which facilitates efficient key location. Typically, in a DHT it is desirable that the nodes' blocks are roughly of equal size, since this usually implies a balanced distribution of the load of storing keys among nodes, and it also simplifies the design of the routing network. We propose and analyze a simple distributed scheme for partitioning I, inspired by the multiple random choices paradigm. This scheme guarantees that, with high probability, the ratio between the largest and smallest blocks remains bounded by a small constant. It is also message efficient, and the arrival or departure of a node perturbs the current partition of I minimally. A unique feature of this scheme is that it tolerates *adversarial* arrivals and departures of nodes.

In the second part of the thesis, we investigate the complexity of a natural decentralized routing protocol, in a broad family of randomized networks. The network family and routing protocol in question are inspired by a framework proposed by Kleinberg to model small-world phenomena in social networks, and they capture many designs that have been proposed for P2P systems. For this model we establish a general lower bound on the expected message complexity of routing, in terms of the average node degree. This lower bound almost matches the corresponding known upper bound.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Vassos Hadzilacos, for supporting me in every possible way. With his unmatched clarity of thought and intelligence, and his integrity as a human being, he has been a role model for me for life. I also want to thank the members of my examining committee: Peter Marbach, Allan Borodin, and Sam Toueg, for their insightful feedback.

I am greatly indebted to my wife, Vasso, for her support, love, and understanding. I am especially grateful to her for bringing little Nikitas-Paraskevas into our life. My gratitude also goes to my three sibling in Greece: Panayiota, Katerina, and Michalis, for their love and care. Finally, I would like to thank my friends in Toronto for making my stay there more enjoyable.

in memory of my parents,
Παρασκευά and Ιωάννας Γιακκούπη

Contents

List of Figures

Index of Notation

$\succeq$ a binary relation on $\mathbf{S}_n$ defined in Section 3.4, 28

$[i..j]$ $\{k \in \mathbb{Z} \,:\, i \leq k \leq j\}$; $i, j \in \mathbb{Z}$, 28

$\langle\!\langle i, k \rangle\!\rangle$ $[i..i+k-1]$; $i, k \in \mathbb{Z}$, 140

$\mathcal{A}_{S,W}$ a set of sequences of binary partitions defined in Section 4.5, 56

$\mathcal{A}_{k,S,W}$ a set of sequences of binary partitions defined in Section 5.7.2, 105

$\mathbf{B}_n$ the set of binary partitions of size n; $n \in \mathbb{N}^*$, 23

$\mathbf{B}$ $\bigcup_n \mathbf{B}_n$, 23

$\mathrm{Bi}(n, p)$ binomial random variable with parameters n and p; $n \in \mathbb{N}^*$, $p \in [0, 1]$, 89

$\mathrm{blk}(B, z)$ the block of B that contains z; $B \in \mathbf{B}$, $z \in I$, 24

ε constant equal to 1/16, 48

f_B a bijective function on I defined in Section 3.6.1; $B \in \mathbf{B}$, 39

$\mathbf{F}_n$ the set of safe sorted binary partitions of size n; $n \in \mathbb{N}^*$, 51

$f^{(k)}(x)$ the function $f(x)$ iteratively applied k times to an initial value of x; $k \in \mathbb{N}$, 154

$\mathbf{\Gamma}$ the set of binary segments, 23

$\mathbf{G}_n$ the set of directed graphs on the set of nodes $[0..n-1]$ that contain as a subgraph the directed ring $0 \to 1 \to 2 \to \cdots \to (n-1) \to 0$; $n \in \mathbb{N}^*$, 138

$\mathcal{G}(\varphi)$ the (uniformly-)augmented ring where the long-range contacts of nodes are selected using distribution φ; $\varphi \in \mathbf{\Phi}_{n,\ell}$, 127

I $[0, 1)$, 23

I_k $\{j2^{-k} \,:\, j = 0, \ldots, 2^k - 1\}$; $k \in \mathbb{N}$, 34

$\lambda_{+/-}$ sampling-size functions, 18, 33, 35

$\lambda(k)$ $\min\{\lambda_+(k),\, \lambda_-(k)\}$, 48, 118

$L(G, u, v)$ delivery time of greedy routing in G from u to v; $G \in \mathbf{G}_n$, $u, v \in [0..n-1]$, 134

$L(G)$ $(1/n) \sum_{0 \leq v < n} L(G, 0, v)$; $G \in \mathbf{G}_n$, 138

$\ell_k(B)$	the total length of blocks of depth k in B; $k \in \mathbb{N}$, $B \in \mathbf{B}$, 24
$\ell_{\bowtie k}(B)$	the total length of blocks in $B \in \mathbf{B}$ whose depths belong to the set $\{i : i \bowtie k\}$; $B \in \mathbf{B}$, $k \in \mathbb{N}$, $\bowtie \in \{\geq, >, <, \leq\}$, 24
$\log^* n$	$\min\{k \geq 0 \,:\, \log^{(k)} n \leq 1\}$; $n \in \mathbb{N}^*$, xi
$m \bmod n$	the $r \in [0..n-1]$ such that $m = kn + r$, for some $k \in \mathbb{Z}$; $m \in \mathbb{Z}$, $n \in \mathbb{N}^*$, xi
$\mu(B)$	the minimum depth of blocks in B; $B \in \mathbf{B}$, 24
$\mathbb{N}$	the set of natural numbers $\{0, 1, 2, \ldots\}$, xi
$\mathbb{N}^*$	$\mathbb{N} - \{0\}$, xi
$\mathbf{\Phi}_{n,\ell}$	the set of all probability distributions φ on the powerset of $[2..n-1]$ such that $\sum_{\Delta \subseteq [2..n-1]} (\varphi(\Delta) \cdot \lvert\Delta\rvert) = \ell$; $n \in \mathbb{N}^*$, $\ell \in [0, n-2]$;, 134
$\check{\Pi}_n$	the borderline-safe partition of size n; $n \in \mathbb{N}^*$, 53
ρ	imbalance, 10
$\varrho(B)$	the balance factor $\xi(B) - \mu(B)$ of B; $B \in \mathbf{B}$, 24
$\mathcal{R}_{S,W}$	a set of sequences of binary partitions defined in Section 4.5, 58
$\tilde{\mathcal{R}}_{S,W}$	a set of sequences of binary partitions defined in Section 5.4.1, 81
$\mathbf{S}_n$	the set of sorted binary partitions of size n; $n \in \mathbb{N}^*$, 23
$\mathbf{S}$	$\bigcup_n \mathbf{S}_n$, 23
$\mathrm{sbl}(b)$	the sibling of b; $b \in \mathbf{\Gamma} - \{I\}$, 23
$s_k(B)$	the number of blocks of depth k in B; $k \in \mathbb{N}$, $B \in \mathbf{B}$, 24
$s_{\bowtie k}(B)$	the number of blocks in B whose depths belong to the set $\{i : i \bowtie k\}$; $B \in \mathbf{B}$, $k \in \mathbb{N}$, $\bowtie \in \{\geq, >, <, \leq\}$, 24
$\mathrm{srt}(B)$	the sorted binary partition that has the same number of blocks of each depth as B; $B \in \mathbf{B}$, 23
$\theta(b)$	the depth of b; $b \in \mathbf{\Gamma}$, 23
$\theta(B, z)$	the depth of $\mathrm{blk}(B, z)$; $B \in \mathbf{B}$, $z \in I$, 24
$T(\varphi)$	the expected delivery time in $\mathcal{G}(\varphi)$, $\mathbb{E}[(1/n^2) \sum_{0 \leq u,v<n} L(G, u, v)]$; $\varphi \in \mathbf{\Phi}_{n,\ell}$, 134
$T(n, \ell)$	$\inf\{T(\varphi) \,:\, \varphi \in \mathbf{\Phi}_{n,\ell}\}$; $n \in \mathbb{N}^*$, $\ell \in [0, n-2]$, 134
$T(n)$	$T(n, 1)$; $n \in \mathbb{N}$, $n \geq 3$, 137
$\xi(B)$	the maximum depth of blocks in B; $B \in \mathbf{B}$, 24
$\mathbb{Z}$	the set of integers, xi

Chapter 1

Introduction

In this chapter we set the context for the problems we will study in this thesis. We begin, in Section 1.1, with an introduction to the peer-to-peer model. In Section 1.2, we describe Distributed Hash Tables, a common substrate for P2P systems. In Section 1.3, we talk about small-world models and how they have been used as prototypes for designing P2P networks. In Section 1.4, we give a brief outline of our work. We conclude, in Section 1.5, with a road-map of the rest of the thesis.

1.1 The peer-to-peer paradigm

One of the most intriguing trends in distributed computing in the past few years has been the surge in popularity of the *peer-to-peer (P2P) paradigm* for building Internet applications. A P2P system is a networked system characterized by the lack of centralized control or a-priori hierarchical organization, in which all or most communication is symmetric. This system is supposed to work in a dynamic setting where nodes (i.e., machines) join and leave the system frequently. It is also expected to scale gracefully as the size of the system grows.

The common model used for implementing P2P systems is that of an *overlay network*. The nodes participating in a P2P system are connected via some (often much larger) underlying network, e.g., the Internet. A node u in the P2P system can establish a *virtual* connection to any other node v (say a TCP connection, if the underlying network is the Internet) as long as u knows the address of v in the underlying network (v's IP address). Each node maintains such connections to a small set of other nodes. The union of these connections forms the P2P network.

The P2P paradigm is attractive for several reasons [9]:

- P2P systems provide the opportunity to aggregate and make use of vast untapped resources across the Internet that would otherwise go unused, such as processing power for large-scale computations, and enormous storage potential.

- The deployment and maintenance of a P2P system is relatively easy and inexpensive since no centralized administration or costly specialized hardware are required — as opposed to centralized systems.

- Designed to operate in dynamic environments, P2P systems have the potential to be robust to failures or malicious attacks.

Broadly speaking, there are two categories of P2P systems: *structured* and *unstructured*. In *unstructured* P2P systems the nodes choose their overlay neighbors arbitrarily. Typically, such systems are easy to build, they support complex queries, and remain functional despite frequent arrivals and departures of nodes. However, they offer no performance guarantees — they work on a best-effort basis. Unstructured systems have enjoyed great success, especially in the form of file-sharing applications, like Napster, Gnutella, Kazaa, eMule, and Bittorrent, which are used by millions of users.

For applications where imprecise/partial results to queries are acceptable, unstructured P2P systems are sufficient. For applications that require stronger guarantees on data storage and retrieval, *structured* P2P designs have been proposed. In these systems the set of overlay connections between nodes is dictated by some pre-defined topology. Because of the stronger guarantees on performance and correctness that these system are expected to deliver, they are typically harder to engineer. So far, structured systems have not witnessed deployments of the scale of unstructured P2P file-sharing systems. However, a wide range of applications have been proposed and deployed, mainly by the research community. These applications include cooperative data storage and archival (CFS [16], OceanStore [70], PAST [73], Ivy [61]), censorship-resistant storage (Freenet [14]), Web caching (Squirrel [33]), group communication and event notification (Bayeux [85], CAN-Multicast [69], Scribe [74], i3 [76], CorONA [66], POST [58]), naming and resource discovery (INS/Twine [10], SETS [12], CoDoNS [67]), content distribution (Coral [25], SplitStream [13]), DB query and indexing (PIER [32], OverCite [78]). Also the file-sharing applications eMule and Bittorrent have recently incorporated structured P2P components that they use for indexing.

1.2 Distributed Hash Tables

Most of the structured P2P systems that have been proposed use a *Distributed Hash Table (DHT)* as a substrate. Briefly, a DHT is a giant hash table that is maintained in a P2P manner by a large and dynamic set of nodes. This hash table is divided into pieces, one for each node, and every node stores the piece that is assigned to it. Connections that are set up among nodes allow each node to efficiently locate the node that is responsible for any given key.

A DHT is a large-scale, decentralized, self-organizing, distributed repository of data items. The nature of the data items depends on the P2P application that uses the DHT; e.g., in a file storage application a data item can be a file (or a piece of a file). Each item is identified by a *key*, which is assumed to be chosen uniformly at random from some (sufficiently large) *key-space*. In practice, the key-space is usually the set $\{0, 1\}^m$, for some large m, and the key associated with each item is generated by applying a cryptographic hash function (such as MD5 or SHA-1) to the item's name/description. Without loss of generality we will assume that the key-space is the *unit interval* $[0, 1)$. Also we will often assume that this interval wraps around such that point 1 coincides with 0 — thus, forming the *unit ring*.

Every node in a DHT is associated with a subset of the key-space, called the node's *block*, such that the blocks of the nodes that are in the system at any given time form a *partition* of the key-space. Each node stores the items whose keys lie in its block. For example, the following simple scheme is used in the Chord DHT [77] to partition its unit ring key-space into nodes' blocks. Every node, upon its arrival to the system, is associated with a randomly selected point in the key-space, called the node's *ID*. The block of each node is the arc consisting of the points that are closer to the node's ID than to the IDs of the other nodes, with respect to the clockwise distance along the ring — from the point to the ID. Note that every time a node arrives or departs from the system, the partition of the key-space changes; and each item whose key is in the subset of the key-space that gets reassigned to a different node must be transferred to the node that becomes responsible for that key.

Communication between nodes is facilitated by maintaining overlay connections between them. Specifically, each node maintains connections to a carefully selected, small set of nodes, such that the resulting *network* can support efficient decentralized *routing* from any node to the node responsible for any given key. A node chooses the nodes it connects to based on the positions of its block and of their blocks in the key-space. For example, Chord employs a hypercube-like routing network, where each node u has connections to the nodes that are responsible for the points at clockwise distances 2^{-k}, for $k = 1, 2, \ldots,$ from u's ID. It also uses the following distance-halving routing scheme: A node u greedily forwards requests for keys that u and its clockwise successor node in the ring are not responsible for to the furthest neighbor of u whose ID precedes the requested key. Note that the routing network must be updated when nodes arrives or depart from the system.

It is straightforward how a node can insert, retrieve or update a data item in a DHT. First it computes the key of the item (the hash-value of its name/description), and then it hands in the request for this key to the routing network which routes the request to the node responsible for this key.

DHTs have received considerable attention from the research community. Among the designs that have been proposed are CAN [68], Chord [77], Tapestry [31], Pastry [72], Viceroy [48], Kademlia [56], Symphony [52], Koorde [34], DH [62], d2b [22], and many more. Also a number of DHT implementations are available (Chord[77], Bamboo[71], P-Grid [1]).

The design of a DHT can be divided into thee components [51]: the *key-space partitioning scheme*, the *routing network*, and the *data management protocol*.

I. **Key-space partitioning scheme:** This is the protocol that describes how the key-space is partitioned among the nodes. Specifically, it describes how a new node is assigned its own block (pieces of which previously belonged to old nodes), and how the block of a departing node is redistributed among (some of) the remaining nodes. A good key-space partitioning scheme should be decentralized; it should have low message complexity; each arrival and departure should incur minimal perturbation to the current partition of the key-space; and it should ensure that all blocks are roughly of the same size. The last requirement is motivated by load balancing, and we quantify it in terms of the ratio ρ between the largest and smallest block sizes the scheme achieves. Specifically, we require that ρ should be small, ideally close to 1. As we will see in Chapter 2, under some reasonable assumptions, ρ is an accurate estimate of the ratio of maximum over minimum number of items stored per node. So, small ρ yields a balanced distribution of the storage load among the nodes. We will talk about key-space partitioning schemes in more detail in Chapter 2.

II. **Routing network:** This component specifies how each node chooses the set of nodes it connects to, and how these connections are updated in the face of arrivals and departures of other nodes. It also describes a decentralized routing protocol for the resulting network that facilitates routing from any node to the node responsible for any given key.

The dynamism and scale of P2P systems stipulate that each node in a DHT should maintain only a small number of connections to other nodes. On the other hand, these connections should facilitate short routing paths between arbitrary nodes. Typical designs achieve logarithmic (in the network size) path lengths, using a logarithmic number of connections per node. A common strategy in the design of routing networks for DHTs is to first identify a *static graph family* that is known to possess good properties, e.g., one of the classical interconnection networks (the hypercube, the butterfly, and the de Bruijn graph) [43], or a probabilistic variation of them; and then describe a dynamic network that "approximates" the topology of this static graph family in the face of arrivals/departures, and large variations in the number of nodes. The dynamic network maintains desired properties of the static graph family including low degree, and efficient routing protocol. Several designs have been proposed that approximate specific static families of graphs. We review some of these designs in Chapter 7. In addition, techniques have been suggested for approximating *arbitrary* static families of graphs [49, 2, 50, 63]. Note that the task of approximating a static graph family is significantly simplified if the key-space partitioning scheme achieves ρ that is bounded by a small constant [50]. This is an additional motivation (besides load balancing) for designing partitioning schemes that guarantee small ρ.

Short routing paths (in the overlay network) do not necessarily mean paths of low latency. Thus, it is also desired that the latency of every routing path be close to the latency of routing between the same pair of nodes in the underlying network (e.g., the ping-time, when the underlying network is the Internet). Routing paths of low latency can be achieved by taking into account the latency between nodes when a node chooses the nodes it sets up connections to, and also when it chooses the next hop to forward a message to in routing. (E.g., see [64, 72, 31, 50].)

Network maintenance is straightforward assuming that any two operations (arrivals or departures of nodes) do not overlap in time, or they affect non-overlapping sets of nodes. Updating the overlay network after each arrival or departure requires then a number of message that is proportional to the degree of the nodes. However, maintenance in the face of frequent (concurrent) arrivals and departures, and possibly failures is a more involved task. For work on this problem see [77, 45, 31, 44, 71].

III. **Data management protocol:** This component is responsible for the *availability* of data items in the face of failures, and for the alleviation of *hot spots* (i.e., the overloading of nodes responsible for keys that are requested extremely often). Various replication and caching protocols have been suggested for this purpose (e.g., see [68, 31, 17, 63]).

1.3 Small-world models and P2P networks

Randomized network constructions that model the *Small-World Phenomenon* — the premise that any two people in a society are connected with short chains of acquaintances, have attracted a lot of research interest recently. The quantitative study of the phenomenon started with Milgram's experiments in the 1960's [57], where people were asked to send letters to unfamiliar targets only through acquaintances. Watts and Strogatz [81] observed that the Small-World Phenomenon is common in many large-scale real-world networks. They also suggested modeling the phenomenon using a simple random graph model: Individuals are the nodes in a ring lattice; each node is connected to its k nearest ring neighbors, for some small constant k — the node's "local contacts," and to a small number of nodes chosen uniformly at random from the set of nodes — the node's "long-range contacts." This model captures two crucial properties of social networks: high clustering, and low diameter — the model has high clustering because of the local contact links, and it has low diameter (like uniform random graphs) because of the long-range contact links.

Kleinberg initiated the study of an algorithmic perspective of the Small-World Phenomenon [39]: small-world experiments showed not only that short paths exist in social networks, but that individuals can find such paths based on local information. He proposed a simple and elegant framework to model this ability of small-world networks to support efficient *decentralized search.* In his model, which extends the model of Watts and Strogatz, individuals are the nodes in a d-dimensional lattice. As in Watts and Strogatz's model, each node has links to its lattice neighbors (local contacts) and to a small number of random, independently selected, long-range contacts. Now, however, each long-range contact of a node u is chosen from a *non-uniform* distribution: a node at lattice distance s from u is selected with probability proportional to $1/s^{\alpha}$, for some constant parameter α of the model. Kleinberg showed that efficient decentralized search is possible only for $\alpha = d$. In particular, when $\alpha = d$, a simple greedy search algorithm works: each node forwards a request to its (local or long-range) contact that is closest to the destination node, with respect to the lattice distance.

Besides modeling a common property of social networks, Kleinberg's work implicitly suggested a new design principle for large-scale networked systems that support efficient decentralized routing. A natural application for this design principle has been the design of P2P systems. There are many examples of structured P2P systems [6, 52, 29, 84], where, as in Kleinberg's model: nodes are embedded in a d-dimensional lattice (with $d = 1$ in most cases); each node has links to its lattice neighbors, and to a small number of long-range contacts selected independently for each node, from a distribution that favors closer nodes over distant ones (in the lattice); and the routing protocol used is the greedy one, with respect to the lattice distance. This combination of network topology and routing protocol is particularly attractive for P2P systems, mainly for its simplicity and the nice properties of greedy routing. We discuss this model and its properties in greater detail in Chapter 7.

1.4 Our contribution

We study two issues that are central to the design of P2P systems: *load balancing*, and *routing*. Specifically, we focus on the following two problems.

Load balancing in DHTs

In the first part of this thesis, we propose and analyze a simple and efficient key-space partitioning scheme that guarantees, with high probability,[1] that ρ remains bounded by a small constant. A unique feature of this scheme is that it can tolerate *adversarial* arrivals and departures of nodes. A preliminary version of this work appeared in [27].

The complexity of greedy routing in uniformly-augmented rings

In the second part of this thesis, we study the message complexity of a natural *greedy* routing protocol, in a broad class of ring-based random networks, called *uniformly-augmented rings*. This model is inspired by Kleinberg's model for small worlds, and captures many P2P designs that have been proposed. For this model we establish a *lower bound* on its routing complexity, which almost matches the corresponding known upper bound. A preliminary version of this work was published in [28].

[1] By "with high probability" (*whp*) we mean "with probability $1 - \mathrm{O}(n^{-c})$ for some constant $c > 0$, for a system of size n."

1.5 Road-map of the thesis

The rest of this thesis is organized as follows. In Chapter 2, we study the problem of load balancing in DHTs, and present a novel key-space partitioning scheme. Chapters 3–6 contain an analysis of the load balancing properties of this scheme. In Chapter 7, we investigate the complexity of greedy routing in the class of uniformly-augmented rings, and we present a new lower bound. A proof of this bound is described in Chapter 8. We conclude, in Chapter 9, with a brief summary of our results, and an outline of some open research problems that are closely related to them.

Chapter 2

Balanced key-space partitioning in DHTs

In this chapter and the next four, we propose and analyze a simple key-space partitioning scheme for DHTs. This scheme is inspired by the multiple random choices paradigm [8, 59]. It achieves, with high probability, a ratio of at most 4 between the loads of the most and least burdened nodes, in the face of both arrivals and departures of nodes. Each arrival and departure incurs an $\mathrm{O}(\log^2 n)$ message cost, where n is the number of nodes, and causes the relocation of keys between at most two nodes (for arrivals) or three nodes (for departures). A unique feature of this scheme is that it provides the above performance guarantees even when the sequence of arrivals and departures is controlled by an adversary.

2.1 Introduction

One of the main components of a DHT is its *key-space partitioning scheme*, which determines how the key-space is divided among the nodes in the system. More precisely, it describes a protocol for repartitioning the key-space, which is executed every time a new node joins the system or an old one departs from it. So, when a new node arrives it is assigned its own block, pieces of which previously belonged to old nodes, and when an old node departs its block is redistributed among some of the remaining nodes. Besides transferring pieces of the key-space *to* the arriving node or *from* the departing node, the arrival or departure of a node may involve transferring pieces of the key-space between (a few) other nodes, as well.

Typically, a key-space partitioning scheme specifies that the arriving (or departing) node

u, which is assumed to be oblivious of the other nodes' blocks initially, learns the blocks of a small sample of other nodes, and then it decides how these blocks should be updated to accommodate the arrival (or departure) of u. In general, u can find out about the block of another node in one of two ways:

- *Local probe*: u retrieves the block of one of its neighbors in the overlay network, or the block of some neighbor of a node that u had previously discovered; for this operation, a constant overhead is required in terms of routing messages in the overlay network.
- *Random probe*: u retrieves the block containing a point selected (uniformly) at random from the key-space, by routing a request to the corresponding node through the overlay network; the incurred overhead is that of routing in the overlay network.

We will quantify the performance of a key-space partitioning scheme using the following three measures:

- *Message cost*: the number of routing messages in the overlay network required to determine how the partitioning of the key-space should be updated to accommodate the arrival or departure of a node. (We do not count the messages needed to transfer the actual items that are reassigned to different nodes, or the messages used to update the overlay connections.)
- *Perturbation*: the number of nodes whose blocks are modified per arrival (departure), in addition to the arriving (departing) node.
- *Imbalance* ρ: the ratio between the largest and smallest block sizes in the system.

The scalability requirement of DHT designs dictates that the message cost of a key-space partitioning scheme should be low; i.e., at most poly-logarithmic in the system size. The perturbation measure is indicative of the amount of changes that result from the arrival or departure of a node. Note that when the block of a node u is modified, say from b to b', the items with keys in $b - b'$, which are currently stored in u, need to be transferred to other nodes, and the items with keys in $b' - b$ stored in other nodes must be transferred to u. Also, overlay connections to and from u may need to be modified. It is, hence, desirable that the perturbation be small, ideally equal to 1. The imbalance ρ measures how balanced the storage load is. Notice that in the (typical) setting where the number of items stored in the system is significantly larger than the number of nodes, ρ is very close to the ratio of maximum over minimum number of items stored per node — recall that the items' keys are selected independently and uniformly at random over the key-space. It is therefore desirable that ρ remains small, ideally bounded by a constant close to 1. An additional advantage of having ρ bounded by a small constant is that it typically simplifies the overlay network construction — it makes it easier to approximate static network topologies [62, 50]. Also, it simplifies a number of useful operations such as obtaining an accurate estimate of the network size, and choosing a node uniformly at random [38].

In the analysis of key-space partitioning schemes it is typical to assume that arrivals and departures of nodes take place *sequentially.* Note, however, that, since in most of these schemes the outcome of an arrival or departure depends only on a very small, randomly selected fraction of the system, it is very likely that two concurrent operations involve disjoint sets of nodes, and, thus, they *appear* as if they occurred sequentially. Hence, usually similar results apply up to some degree of concurrency, as well.

In general, the performance of a key-space partitioning scheme depends on how the sequence of arrivals and departures that takes place is decided. We distinguish two different models.

- ⋆ *Random arrivals/departures*: the order in which arrivals and departures occur is determined *before* the first of these operations takes place. The actual node that departs when a departure takes place is decided later, when the departure is about to occur; the node to depart is selected uniformly at random among the nodes in the system at that time.
- ⋆ *Adversarial arrivals/departures*: the order in which arrivals and departures take place, and the actual node that departs when a departure occurs are determined by an adaptive adversary; the adversary decides the next operation to take place based on the complete history of the system up to that point.

Using standard terminology of online algorithms, in the first model the order in which arrivals and departures take place is determined by an *oblivious adversary*, while in the second model the order of arrivals and departures, and the nodes that depart are determined by an *adaptive online adversary.* Some of the key-space partitioning scheme that have been proposed were analyzed assuming that no nodes ever depart from the system. Note that under this restriction, the two models we described above are equivalent.

In the rest of this chapter, we first survey key-space partitioning schemes that have been proposed in the literature, in Section 2.2. Then, in Section 2.3, we describe a novel key-space partitioning scheme, discuss its properties, and compare it to existing schemes. We conclude, in Section 2.4, with a road-map of the analysis of this scheme, which is described in Chapters 3–6.

2.2 Existing key-space partitioning schemes

2.2.1 Early schemes

Most of the DHT structures that have been proposed so far employ one of two key-space partitioning schemes, suggested in early DHT designs. We will refer to them as the *consistent-hashing paradigm*, and the *random-tree paradigm*. We describe each of these schemes below.

Consistent-hashing paradigm

This scheme was first used in Chord [77], and it was inspired by the consistent-hashing technique [35]. The key-space used in this scheme is a ring, and every node is responsible for the keys in some arc of this ring. Specifically, every node upon its arrival is associated with a random *ID*, drawn independently and uniformly from the key-space. The block each node is responsible for is the arc consisting of the points that are closer to the ID of that node than to the ID of any other node in the system, with respect to the clockwise distance along the ring, from the point to the ID. (Other distance functions, such as the absolute distance along the ring, have also been considered.) Whenever a new node arrives to the system or an old one departs, the partition of the key-space is updated accordingly.

This scheme is very efficient in terms of message cost, since it requires a single random probe per arrival, while departures do not require any probes — the clockwise successor node of the departing node assumes responsibility of its blocks. Also, the incurred perturbation is minimum, i.e., the block of just one other node is modified per arrival or departure. However, this scheme does not achieve ρ bounded by a constant. Specifically, if arrivals and departures are random then the ratio of *largest* to *average* block sizes in an n-node system is $\Theta(\log n)$, with high probability (*whp*) [48]; and the ratio of *average* to *smallest* block sizes is $\Omega(n/\ln n)$, whp (see [38]).

If every *physical node* acts as $k = \Omega(\log n)$ independent *virtual nodes*, each associated with a distinct arc, then ρ is bounded by a constant, whp, in the face of random arrivals and departures [35, 77]. However, this approach inflates the message cost and the perturbation by a factor of k. The number of overlay connections a node should maintain is also increased by that factor.

A useful property of the consistent-hashing paradigm is that the current key-space partition depends only on the set of IDs currently in the system; specifically, it does *not* depend on the history of nodes' arrivals and departures that have taken place. On the other hand, the random-tree paradigm we describe next does not have this property. Consequently, the analysis of the consistent-hashing paradigm is simpler. We discuss this issue in more detail at the end of Section 2.3.

Random-tree paradigm

This is the key-space partitioning scheme proposed in CAN [68]. The key-space used is a d-dimensional hyper-rectangle, for some $d \geq 1$, and the block of each node is a d-dimensional hyper-rectangular subspace. Arrivals and departures of nodes are handled as follows. A newly-arrived node performs a random probe; the block retrieved is split in half along one of the dimensions; and the new node assumes responsibility for one half, while the other half remains with the node previously responsible for the whole block. Note that the split is done by assuming a certain ordering of the dimensions in deciding along which dimension the block is to be split, so that blocks can be re-merged when nodes depart (as we describe next). To describe how departures are handled we will need the following two definitions. The two blocks that result by splitting a block in half are called *sibling* blocks; and the blocks that result from a block by one or more splits are called the *descendants* of that block. When a node departs from the system, it hands over its block b to the node responsible for the sibling b' of b, if b' is not currently split in smaller blocks. Otherwise, two existing sibling blocks a, a' that are descendants of b' are identified. (There is always at least one such pair of blocks.) a and a' are then merged and the resulting block is assigned exclusively to one of the two nodes previously responsible for them, while the other node becomes responsible for b.

The above scheme can be conveniently described as a process executed on a binary tree. Think of the blocks in the current partition of the key-space as the leaves of a *full* binary tree. Each internal vertex of this tree represents a block that no longer exists — it was split at some previous time; its children correspond to the two sibling blocks into which it was split. We call this tree the *partition tree*. The procedure for handling the arrival of a new node in the system can then be described as: pick a leaf at random such that a leaf that is at depth ℓ is chosen with probability $2^{-\ell}$ (i.e., proportional to the size of the corresponding block), and attach two children to it. Likewise, the procedure for handling the departure of a node simply removes the leaf that corresponds to the departing node and its sibling vertex v, if v is also a leaf; if v is not a leaf, it removes, instead, two sibling leaves that are in the subtree rooted at v.

Note that the difference between the maximum and minimum depths of leaves in the partition tree is $\log \rho$. So, the notion of balance of a key-space partition readily translates into the balance of the corresponding partition trees. Another useful observation is that the depth of a leaf in a balanced partition tree is a very accurate estimate of $\log n$; in particular it approximates $\log n$ within a $\log \rho$ additive term.

It is not difficult to show that if no nodes ever depart from the system, the scheme achieves the same ratio of largest to average block sizes as the consistent-hashing paradigm, and better ratio of average to smallest block sizes. (The latter is $\Omega(\log n / \log \log n)$, whp.) The message cost and perturbation per arrival are the same as in the consistent-hashing paradigm. For departures, the message cost is $\mathrm{O}(\log \rho)$, and the perturbation is at most 2. (For the message cost of departures, we assume that each node u maintains connections to all the nodes whose blocks are adjacent to u's in the key-space. So, the departing node can identify the pair of sibling blocks that should be merged using at most $\log \rho + 1$ local probes.)

2.2.2 Schemes that achieve bounded ρ

Neither of the key-space partitioning schemes we described in the previous section achieves imbalance bounded by a constant (unless we employ the costly technique where each node acts as $\Theta(\log n)$ virtual nodes.) A number of schemes were subsequently proposed specifically to address this issue. Four of them [2, 4, 50, 37] are variations of the random-tree paradigm, and one [36] is a variant of the consistent-hashing paradigm. Two of these schemes [50, 36] provably maintain bounded ρ in the face of both (random) arrivals and departures. The other three either describe a procedure for handling departures but they do not analyze it, or they do not handle departures at all. We briefly describe these schemes below.

Variations of the random-tree paradigm

All the variations of the random-tree paradigm follow roughly the same approach. The arrival of a node is handled by sampling a set A of $\Theta(\log n)$ blocks, and then picking a *largest* block in A and splitting this block between the new node and its previous owner, as in the random-tree paradigm. (Recall that $\log n$ can be estimated within a $\log \rho$ additive term, from the size of the block of a node.) The departure of a node is accommodated by again sampling a set D of $\Theta(\log n)$ blocks, picking a *smallest* block in D, switching blocks between the owner of the block picked and the departing node, and then executing the procedure for departures of the original random-tree paradigm. The variations proposed are differentiated mainly by the way the sets A and D are chosen.

Inspired by the multiple random choices paradigm [8, 59], Abraham *et al.* [2] analyzed the scheme for handling only arrivals, where the sample set A consists of the blocks retrieved by $k \geq 2$ *independent* random probes. (Recall that in the original random-tree paradigm $k = 1$.) They showed that after a sequence of n arrivals (starting from an empty system), $\rho = \mathrm{O}(2^{2 \log\log n / \log k})$, whp; hence, for $k = \Omega(\log n)$, $\rho = \mathrm{O}(1)$. If $k = \Theta(\log n)$ the message cost is $\Theta(R \log n)$, where R is the message complexity of routing in the overlay network. The same protocol (for $k = \Theta(\log n)$) was independently studied by Naor and Wieder [62].

Adler *et al.* [4] studied the variation where A consists of a block b retrieved using a random probe, and of the blocks of the overlay neighbors of b's owner (retrieved using local probes); and D consists of the block of the departing node and the blocks of its overlay neighbors. Clearly, this scheme cannot achieve balanced partitions for arbitrary overlay networks. However, Adler *et al.* showed that it guarantees bounded ρ when combined with a hypercube-like overlay network of a logarithmic node degree that they described. Specifically, they proved that, whp, ρ is bounded after a sequence of n arrivals (starting from an empty system). Experimental evaluation suggests that the scheme works well in the face of random departures of nodes, as well, but no analytical results are known in this case. The message cost is $R + \Theta(\log n)$ for arrivals, and $\Theta(\log n)$ for departures, where $R = \Theta(\log n)$.

Manku [50] studied two variations of the random-tree paradigm: one that handles arrivals only, and a more involved version of it, which also supports departures. In the first scheme, A is chosen as follows. We perform a random probe; let b be the block retrieved. A then consists of the blocks that are leaves of a subtree T of the partition tree such that T contains b and has size $\Theta(\log n)$. The blocks in $A - \{b\}$ are retrieved using local probes. This scheme has similar performance and provides analogous guarantees as the scheme proposed by Adler *et al.* [4].

In the more involved variation of the random-tree paradigm that Manku described in [50], A is selected as before, and, similarly, D consists of the leaves in a subtree of size $\Theta(\log n)$ that contains the block of the departing node. The scheme also assigns to each node a random *ID*. A node's ID is initially the point selected by the random probe that is performed upon the node's arrival, but it may change subsequently — the node may switch its ID with another node's. When a node departs, its *current* ID disappears from the system. These IDs affect the choice of which among the blocks in A is split during an arrival, and which blocks in D are merged during a departure. Roughly speaking, the following invariant is maintained: each subtree T of size $\Theta(\log n)$ is balanced, and the IDs associated with its leaves belong to the block that corresponds to the root of T. This scheme guarantees that, starting from an empty system, a sequence of random arrivals and departures results in $\rho \leq 4$, whp (in the final number of nodes in the system). The message cost is $R + \Theta(\log n)$ for arrivals, and $\Theta(\log n)$ for departures.

Kenthapadi and Manku [37] proposed a scheme for handling only arrivals, which combines the first of the schemes presented in [50] and the scheme in [2, 62]. A is chosen as follows. We perform k independent random probes, where k is a system parameter, and it may be a function of the system size. Let $b_1, \ldots, b_k$ be the blocks retrieved by these probes. A consists then of the blocks that are leaves in the subtrees $T_1, \ldots, T_k$ of the partition tree, where T_i contains b_i, and has size $\Theta(\frac{1}{k} \log n)$. Note that when $k = 1$ the scheme is the same as that in [50], while for $k = \Theta(\log n)$ it resembles the scheme in [2, 62]. For $k = \mathrm{O}(\log n)$, it is shown that $\rho \leq 8$, whp, and the message cost is $\Theta(kR + \log n)$. A procedure symmetric to that of handling arrivals was suggested to handle departures but no analysis was provided.

A variation of the consistent-hashing paradigm

Karger and Ruhl [36] proposed the following variation of the consistent-hashing paradigm. The key-space is the unit ring $[0, 1)$. Each node is associated with $\Theta(\log n)$ randomly selected IDs — points in $[0, 1)$, only one of which is *active* at any time. The key-space is partitioned among nodes as in the consistent-hashing paradigm, based on the nodes' currently active IDs. A node chooses its active ID as follows. Consider the following ordering of all points in $[0, 1)$ that have a finite binary representation: 0, 1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, 1/16, . . . (Note that the i-prefix of this sequence divides the unit ring into i arcs of lengths $2^{-\lfloor \log i \rfloor}$ and $2^{-\lceil \log i \rceil}$.) After a newly-arrived node has chosen its IDs, it activates the one that results in assigning to the new node the block containing a point x that appears earliest in the above ordering. This may cause the node previously responsible for x to change its own active ID, and so on. Departures are also followed by chained reactions of nodes that change their active IDs. Karger and Ruhl showed that this scheme always converges to a (unique) stable key-space partition, and that this partition has $\rho = \mathrm{O}(1)$, whp (assuming random arrivals and departures). The incurred perturbation, for both arrivals and departures, is $\mathrm{O}(\log \log n)$ expected, and $\mathrm{O}(\log n)$ whp. Finally, the message cost is $\mathrm{O}(R \log n)$ per arrival and $\mathrm{O}(\log n)$

per departure.

Other approaches

Two key-space partitioning schemes that use additional data structures are sketched in [48] and [62], namely the Bucket Solution and the Cyclic Scheme, respectively. These schemes presumably achieve bounded ρ, but their details and formal analysis are not provided. The basic idea behind the Bucket Solution is similar to that in the scheme proposed by Manku in [50]: nodes are organized into *buckets* of size $\Theta(\log n)$, and the nodes inside each bucket balance the load among them. The Cyclic Scheme is deterministic and it works, roughly, by ensuring that each block (in the unit ring key-space) has length $2^{\lfloor \log n \rfloor}$ or $2^{\lceil \log n \rceil}$, and all large blocks are in one contiguous interval — and, thus, the same is true for the small blocks. A serious limitation of this scheme is that it does not allow for concurrent arrivals or departures of nodes.

2.3 A new key-space partitioning scheme

The key-space partitioning scheme we propose is a variation of the random-tree paradigm, and is inspired by the multiple random choices paradigm [8, 59] — similarly to the scheme suggested in [2, 62]. Below we give a detailed description of this scheme and discuss its properties.

2.3.1 Description of the scheme

As in the variations of the consistent-hashing paradigm we saw in Section 2.2.2, the scheme we propose differs from the original random-tree paradigm (described in Section 2.2.1) in the way it chooses which block is split when a new node arrives, and which pair of sibling blocks is merged when an old node departs. Roughly speaking, the arriving or departing node performs a logarithmic number of independent random probes; the block split in case of an arrival is the largest among those discovered; while the pair of blocks merged in case of a departure corresponds to the smallest block discovered.

More precisely, a newly-arrived node u performs the following steps. First it executes a single random probe. Let δ be the depth of the block retrieved — i.e., the depth of the corresponding leaf in the partition tree. u then curries out $\lambda_+(\delta) - 1$ additional independent random probes, where function λ_+ is a parameter of the scheme such that $\lambda_+(k) = \Theta(k)$.

Next, u picks a *largest* block b among those discovered (including the block retrieved in the initial probe). Finally, as in the original random-tree paradigm, b is split into two halves, and one half becomes the block of u while the other remains with the node previously responsible for b.

The departure of a node u is handled as follows. Before u leaves the system it performs $\lambda_-(\delta)$ independent random probes, where δ is the depth of u's block, and λ_- is again a parameter of the protocol such that $\lambda_-(k) = \Theta(k)$. If any of the blocks discovered by the random probes is smaller than u's, then u exchanges blocks with the owner of a *smallest* of these blocks. (Otherwise, no exchange of blocks takes place.) Finally, u executes the procedure that handles the departure of a node in the original random-tree paradigm.

2.3.2 Properties

As in the original random-tree paradigm, the depth of each block in an n-node system is within $\log n \pm \log \rho$. Hence, the number of *random* probes the scheme performs per arrival or departure of a node is $\mathrm{O}(\log n + \log \rho)$. In particular, when ρ is bounded by a constant, $\Theta(\log n)$ random probes are performed. The number of *local* probes required per departure is $\mathrm{O}(\log \rho)$ — the same as in the random-tree paradigm. Therefore, the message cost of the scheme is $\mathrm{O}(R \cdot (\log n + \log \rho))$ for both arrivals and departures, where R is the message complexity of routing in the overlay network. In particular, when ρ is bounded by a constant, the message cost is $\Theta(R \log n)$. Also, as in the random-tree paradigm, the perturbation is 1 for arrivals, and at most 2 for departures.

The scheme achieves ρ bounded by a constant in the face of *adversarial* arrivals and departures. We show that if functions λ_+ and λ_- are large enough (i.e., $\lambda_\pm(k) \geq ak + b$, for large enough constants a and b) then the following two results hold. We call a partition of the key-space *safe* if, roughly speaking, either $\rho \in \{1, 2\}$, or $\rho = 4$ and most blocks are of intermediate size.

(a) If we start from a safe partition of the key-space then, whp, in $\Theta(n)$ operations (arrivals or departures) another safe partition is reached, and all intermediate partitions have $\rho \leq 4$.

(b) Starting from any non-safe partition of the key-space, a safe partition is reached after $\mathrm{O}(d2^d)$ operations, whp, where d is the maximum depth of blocks initially.

Combining the above two results we also show that (for large enough λ_+, λ_-) the scheme typically results in "long" intervals where $\rho \leq 4$, interrupted by "much smaller" intervals where it may be $\rho > 4$.

Our analysis of the load balancing the scheme achieves depends critically on two properties of the scheme: (1) when a node arrives, a block of depth d is split only after $\Omega(d)$ random probes have been preformed; and (2) during each departure at least $\Omega(\mu)$ random probes are performed, where μ is the minimum depth of blocks in the system. So, the same analysis applies even if the first of the blocks sampled during an arrival (and, thus, δ) is chosen by the adversary — not by a random probe. Also, in departures, any value for δ such that $\delta \geq \mu$ would work; we let δ be the depth of the departing node just for simplicity.

Comparison with other schemes

In the scheme we presented, the procedure for handling arrivals is very similar to that suggested in [2, 62]. The procedure for handling departures, however, is novel, and so is the analysis of the scheme's load balancing properties.

A unique feature of our scheme is that it achieves bounded ρ in the face of *adversarial* arrivals/departures — resolving an open problem posed in [82]. As we saw in Section 2.2.2, only two other schemes provably achieve ρ bounded by a constant in the face of *both* arrivals and departures; namely those proposed in [36] and [50]. However, both of them do so under the (strict) model of *random* arrivals/departures.

About the other performance measures, the message cost of our scheme is larger by a logarithmic factor compared to the scheme in [50], while both schemes incur the same perturbation. Compared to the scheme in [36], our scheme has similar message cost (the same cost for arrivals and worse cost by a logarithmic factor for departures), and smaller perturbation.

The schemes in [36] and [50] possess the nice property that the partition of the key-space at each time depends only on the IDs of the nodes that are in the system at that time; so, it does *not* depend on the actual sequence of nodes' arrivals and departures that resulted in these IDs.[1] This Markovian property means that the system can be analyzed as if it were static, with a fixed set of nodes and resources. Such an analysis is generally much simpler than a dynamic, history-dependent analysis. Our scheme, on the other hand, does not have this property, thus, its analysis is more intricate. It is important to stress, however, that this complexity is an aspect of the analysis, not of the scheme itself. Our scheme is much simpler than the schemes in [36] and [50].

[1] More precisely, this is true for the scheme in [36]; for the scheme in [50] a slightly weaker invariant holds.

We conclude with an informal explanation of why the schemes in [36] and [50] cannot guarantee that ρ stays bounded by a constant in the face of adversarial arrivals/departures. As we mentioned above, in both schemes at each time ρ depends only on the IDs of the nodes that are in the system at that time. The schemes' ability to guarantee that ρ stays bounded under random arrivals/departured depends critically on the fact that in this model the IDs that there are in the system at each time are distributed independently and uniformly over the key-space. (Recall that new IDs are generated independently and uniformly at random; the order of arrivals and departures is predetermined; and in each departure the ID that is deleted is selected independently and uniformly at random among the existing IDs.) Under the adversarial model of arrivals/departures, however, the IDs in the system may not be distributed independently and uniformly. For example, the adversary may choose to remove all the IDs that fall in certain parts of the key-space resulting in an arbitrarily skewed distribution.

2.4 Road-map of the analysis

In the next four chapters we provide an analysis of the key-space partitioning scheme we presented above.

In Chapter 3, we formulate as a random process the evolution of the key-space partition in a DHT that employs the above scheme; we call this a $\mathcal{B}$-process. We then describe a slight variation of this process, called $\mathcal{S}$-process, which is easier to analyze and is shown to have worse load balancing performance that the $\mathcal{B}$-process. This result will allow us to analyze the $\mathcal{S}$-process, instead, and obtain performance bounds that also apply to the $\mathcal{B}$-process.

In Chapters 4 and 5, we study the load balancing properties of the $\mathcal{S}$-process. Specifically, in Chapter 4, we consider the case where the initial partition of the key-space is safe. We show that if we start from a safe partition of n blocks (and λ_+, λ_- are large enough) then, whp, in $\Theta(n)$ steps (i.e., arrivals or departures) another safe partition is reached, and all intermediate partitions have $\rho \leq 4$. In Chapter 5, we consider the complementary case, where the starting partition is not safe, and we provide a probabilistic upper bound on the number of steps required to reach a safe partition. In particular, we show that if λ_+, λ_- are large enough then a safe partition is reached after $\mathrm{O}(d2^d)$ steps, whp, where d is the maximum depth of blocks initially.

Finally, in Chapter 6, we use the result of Chapter 3 to show that the results of Chapters 4 and 5, which were shown for the $\mathcal{S}$-process, readily apply to the $\mathcal{B}$-process, as well. Also, we establish a lower bound on the expected fraction of time in the $\mathcal{B}$-process during which we have $\rho \leq 4$.

Chapter 3

Analysis of our scheme – Part I: Switching to a simpler process

In this chapter we formulate as a random process the evolution of the key-space partition of a DHT that uses the key-space partitioning scheme described in Section 2.3. We then describe a slight variation of this process, which is easier to analyze, and we show it has "stochastically worse" performance than the original process, with respect to load balancing. In the next chapters we analyze this new process, and then use the results of this chapter to show that certain performance bounds we proved for the new process apply to the original process, as well.

We begin, in Section 3.1, with a description of the model of arrivals and departures we will assume in the analysis. In Section 3.2, we describe binary partitions, the class of feasible partitions under the scheme we consider, and we introduce some related terminology. We define some basic operations on binary partitions in Section 3.3, and describe a means to compare the balance of different binary partitions in Section 3.4. In Section 3.5, we introduce the two random processes we mentioned above. We compare these two processes, with respect to their load balance, in Section 3.6.

3.1 The model

As is typical in the analysis of key-space partitioning schemes, we only consider the case where arrivals and departures of nodes occur *sequentially.* (Recall the relevant discussion in Section 2.1.) The sequence of arrivals and departures that takes place is assumed to be controlled by an *adaptive online adversary*: the adversary decides the next operation — i.e., whether a new node arrives or an old one departs, and, in the latter case, which particular node departs — based on the past history of the system so far. The adversary cannot affect the nodes' random choices, nor does she know the outcome of those random choices in advance.

As in the random-tree paradigm, the key-space partitioning scheme we propose works for any underlying key-space that is a d-dimensional hyper-rectangle (or a d-torus), for $d \geq 1$. For the sake of concreteness of the analysis, however, we will assume (without loss of generality) that the key-space is one-dimensional ($d = 1$) and has unit size; i.e., it is the unit interval $I = [0, 1)$.

3.2 Binary partitions

In this section we look at the partitions of the key-space $I = [0, 1)$ that can result by using the key-space partitioning scheme we proposed in Section 2.3; we call these partitions *binary partitions*.

The set of *binary segments*, denoted $\mathbf{\Gamma}$, is the set of intervals that is recursively defined by

– $I \in \mathbf{\Gamma}$, and

– if $[x, y) \in \mathbf{\Gamma}$ then $[x, \frac{x+y}{2})$ and $[\frac{x+y}{2}, y)$ are also in $\mathbf{\Gamma}$.

Note that each binary segment b is a subinterval of I, and it has length 2^{-k}, for some $k \in \mathbb{N}$; this k is called the *depth* of b, and is denoted $\theta(b)$. I has depth 0, and all the other binary segments have depth ≥ 1. For every $b \in \mathbf{\Gamma} - \{I\}$, the *sibling* of b, denoted $\mathrm{sbl}(b)$, is the (unique) binary segment such that

$$b \cap \mathrm{sbl}(b) = \emptyset \quad \text{and} \quad b \cup \mathrm{sbl}(b) \in \mathbf{\Gamma}$$

For example, if $b = [3/8, 1/2)$ then $\mathrm{sbl}(b) = [1/4, 3, 8)$. Note that

$$\theta(\mathrm{sbl}(b)) = \theta(b) \quad \text{and} \quad \theta(b \cup \mathrm{sbl}(b)) = \theta(b) - 1$$

A partition of I into blocks is called a *binary partition* if all the blocks are binary segments. For $n \geq 1$, we denote by $\mathbf{B}_n$ the set of all binary partitions of size n, and define $\mathbf{B} = \bigcup_n \mathbf{B}_n$. It is easy to see that $\mathbf{B}$ represents the set of all partitions that can result by using our key-space partitioning scheme (or, any key-space partitioning scheme that follows the random-tree paradigm). In most of our analysis we deal with a special class of binary partitions, called *sorted* binary partitions. A binary partition is *sorted* if its blocks are sorted (from left to right) in *non-decreasing depth* (or, equivalently, in *non-increasing length*); i.e., for any two blocks $b = [x, y)$ and $b' = [x', y')$, if $x < x'$ then $\theta(b) \leq \theta(b')$. For any $B \in \mathbf{B}$, there is a unique sorted binary partition that has the same number of blocks of each depth as B; we denote it by $\mathrm{srt}(B)$. Similarly to $\mathbf{B}_n$ and $\mathbf{B}$, we let $\mathbf{S}_n$ be the set of all sorted binary partitions of size n, and $\mathbf{S} = \bigcup_n \mathbf{S}_n$.

Next, we introduce some notation for quantities associated with $B \in \mathbf{B}$. The total number of blocks of depth k, for $k \geq 0$, is denoted $s_k(B)$, and the sum of their lengths is denoted $\ell_k(B)$; so, $\ell_k(B) = 2^{-k} s_k(B)$. For $\bowtie \in \{\geq, >, <, \leq\}$, we denote by $s_{\bowtie k}(B)$ the number of blocks whose depths belong to the set $\{i : i \bowtie k\}$; e.g., $s_{\geq k}(B) = \sum_{i \geq k} s_i(B)$. We define $\ell_{\bowtie k}(B)$ analogously. The minimum depth of blocks in B is denoted $\mu(B)$, and the corresponding maximum $\xi(B)$. The difference $\xi(B) - \mu(B)$ is called the *balance factor* of B, and is denoted $\varrho(B)$. Note that if ρ is the imbalance of B (i.e., the ratio between the largest and smallest block sizes) then

$$\varrho(B) = \log \rho$$

For any $z \in I$, $\mathrm{blk}(B, z)$ denotes the block of B that contains z, and $\theta(B, z)$ denotes the depth of this block; i.e., $\theta(B, z) = \theta(\mathrm{blk}(B, z))$.

In all the above notation, we will often omit parameter B when it is clear which binary partition we are referring to. For example, we will say “for every $B \in \mathbf{B}$, $\mu \leq \xi$” instead of “for every $B \in \mathbf{B}$, $\mu(B) \leq \xi(B)$;” also, we will write $\ell_\xi(B)$ instead of $\ell_{\xi(B)}(B)$ — or even ℓ_ξ when B is clear from the context.

Partition trees

In our analysis, it will often be convenient to think of a binary partition B in terms of a binary tree. Each node in this tree is a binary segment, and it has either two children or none: The root of the tree is the whole interval I. For every internal node $[x, y)$, its left and right children are the binary segments $[x, \frac{x+y}{2})$ and $[\frac{x+y}{2}, y)$, respectively. The leaves of the tree are the blocks of B. We call this tree the *partition tree* of B. An example is illustrated in Figure 3.1(a). Note that for any node u of the tree, $\theta(u)$ is equal to the depth of u in the tree — i.e., u's distance from the root. Also $\mathrm{sbl}(u)$ is precisely the sibling node of u in the tree — i.e., the node with the same parent as u. Finally, note that if B is sorted then in each level of the tree the nodes of the tree occupy the rightmost spots consecutively, with no spot left unoccupied in between any two.

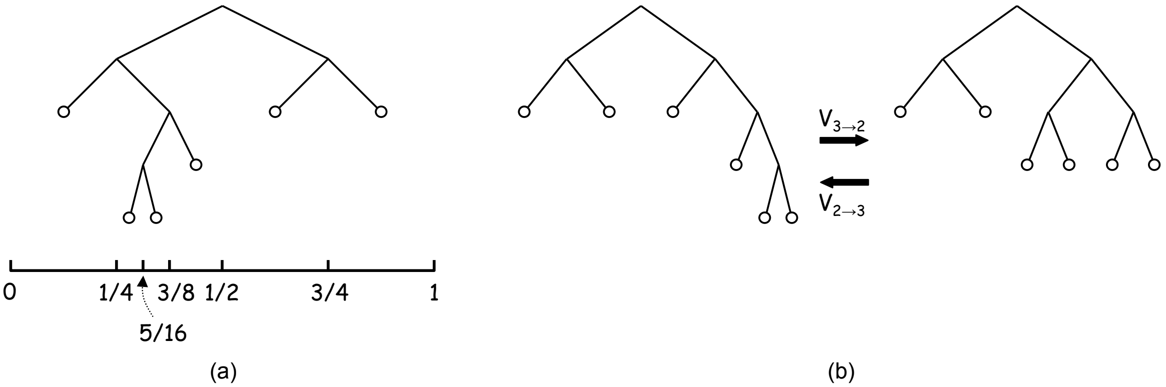

Figure 3.1: (a) A binary partition and the corresponding partition tree; (b) example of a $\mathbb{V}_{k\to k'}$ operation.

3.3 Basic operations on binary partitions

The two most basic operations on a binary partition are splitting a given block into two blocks, and merging a given pair of sibling blocks into a single block. Let $B \in \mathbf{B}$ and $b \in B$. We denote by $\textsc{spltBlk}(B, b)$:

the binary partition that results from B by splitting b into two sibling blocks

If also $|B| > 1$ and $\mathrm{sbl}(b) \in B$, we denote by $\textsc{mrgBlk}(B, b)$:

the binary partition that results from B by merging b and $\mathrm{sbl}(b)$ into a single block

Thinking of binary partitions in terms of their partition trees, $\textsc{spltBlk}(B, b)$ adds a pair of children to leaf b in the partition tree of B, while $\textsc{mrgBlk}(B, b)$ deletes leaf b and its sibling node (which must also be a leaf).

We describe now a variation of the above two operations that are specific for *sorted* binary partitions. For any $S \in \mathbf{S}$ and $k \in \mathbb{N}$ such that $s_k(S) > 0$, we define

$$\mathbb{S}_k(S) = \textsc{spltBlk}(S, b), \quad \text{where } b \text{ is the last (rightmost) block in } S \text{ of depth } k.$$

Also, if $s_k(S) \geq 2$ we let

$$\mathbb{M}_k(S) = \textsc{mrgBlk}(S, b'), \quad \text{where } b' \text{ is the first (leftmost) block in } S \text{ of depth } k.$$

Clearly, $\mathbb{S}_k(S)$ and $\mathbb{M}_k(S)$ are also sorted binary partitions. In the context of partition trees, $\mathbb{S}_k(S)$ attaches a pair of children to the rightmost leaf in the partition tree of S at depth k, while $\mathbb{M}_k(S)$ removes the two leftmost leaves at depth k.

The next lemma states a simple relation between $\textsc{spltBlk}$ and $\mathbb{S}$, and the analogous relation between $\textsc{mrgBlk}$ and $\mathbb{M}$. The proof is straightforward and is omitted.

Lemma 3.1. *For all $B \in \mathbf{B}$ and $b \in B$,*

(a) $\operatorname{srt}(\textsc{spltBlk}(B, b)) = \mathbb{S}_{\theta(b)}(\operatorname{srt}(B))$

(b) if $|B| > 1$ *and* $\operatorname{sbl}(b) \in B$ *then* $\operatorname{srt}(\textsc{mrgBlk}(B, b)) = \mathbb{M}_{\theta(b)}(\operatorname{srt}(B))$

The lemma below, describes some simple properties of $\mathbb{S}$ and $\mathbb{M}$. Parts (a) and (b) say that $\mathbb{S}_k$ and $\mathbb{M}_{k+1}$ are the inverse of one another. Parts (c)–(e) give sufficient conditions under which two such operations commute. To simplify notation when we have nested applications of $\mathbb{S}$ and/or $\mathbb{M}$ operations, we will often omit all pairs of brackets except for the innermost pair; for example, we will write $\mathbb{S}_{k_3}\,\mathbb{S}_{k_2}\,\mathbb{M}_{k_1}(S)$ to denote $\mathbb{S}_{k_3}(\mathbb{S}_{k_2}(\mathbb{M}_{k_1}(S)))$.

Lemma 3.2. *For all $S \in \mathbf{S}$ and $k, k' \in \mathbb{N}$*

(a) if $s_k \geq 1$ then $\mathbb{M}_{k+1}\,\mathbb{S}_k(S) = S$

(b) if $s_k \geq 2$ then $\mathbb{S}_{k-1}\,\mathbb{M}_k(S) = S$

(c) if $k \neq k'$ and $s_k, s_{k'} \geq 1$ then $\mathbb{S}_{k'}\,\mathbb{S}_k(S) = \mathbb{S}_k\,\mathbb{S}_{k'}(S)$

(d) if $k \neq k'$ and $s_k, s_{k'} \geq 2$ then $\mathbb{M}_{k'}\,\mathbb{M}_k(S) = \mathbb{M}_k\,\mathbb{M}_{k'}(S)$

(e) if $k = k'$ and $s_k \geq 3$, or $k \neq k'$ and $s_k \geq 1$ and $s_{k'} \geq 2$ then $\mathbb{M}_{k'}\,\mathbb{S}_k(S) = \mathbb{S}_k\,\mathbb{M}_{k'}(S)$

The proof of Lemma 3.2 is straightforward and is omitted.

We now define another operation that we will use extensively. We begin by describing the operation in the context of partitions trees, where its definition is more natural, and then we express it in terms of the basic operation we defined earlier. Let $S \in \mathbf{S}$ and $k, k' \in \mathbb{N}$ be such that in the partition tree of S

(i) there is a pair of (sibling) leaves that are at depth $k+1$ — or, equivalently, their parent is at depth k; and

(ii) there is a leaf at level $k' \neq k$, different from the pair of leaves in (i).

We denote by $\mathbb{V}_{k \to k'}(S)$ the sorted binary partition that results from S, if in the partition tree of S the leftmost pair of sibling leaves whose parent is at depth k are detached from their parent and then attached to the rightmost leaf at depth k'. An example is illustrated in Figure 3.1(b). Note that if $k > k'$ the two leaves move closer to the root, while if $k < k'$ they move farther away. In the former case we say that a *move-up* operation occurs, while in the latter that a *move-down* operation occurs. Notice that a move-up operation results in a more balanced partition tree than the original, while a move-down operation results in a less balanced tree. We will make use of this observation in Section 3.4 to compare binary partitions in terms of their balance.

It is straightforward to verify that conditions (i) and (ii) above are equivalent to

$$k' = k+1 \text{ and } s_{k+1} \geq 3 \quad \text{or} \quad k' \notin \{k, k+1\} \text{ and } s_{k+1} \geq 2 \text{ and } s_{k'} \geq 1$$

and that $\mathbb{V}_{k\to k'}(S)$ can be expressed as

$$\mathbb{V}_{k\to k'}(S) = \mathbb{S}_{k'}\,\mathbb{M}_{k+1}(S),$$

By Lemma 3.2(e), we also have the following equivalent definition for $\mathbb{V}_{k\to k'}(S)$:

$$\mathbb{V}_{k\to k'}(S) = \mathbb{M}_{k+1}\,\mathbb{S}_{k'}(S)$$

Finally note that, by Lemmata 3.2(a) and (b),

$$\mathbb{S}_{k'}\,\mathbb{M}_{k+1}(S) = S' \;\Rightarrow\; \mathbb{M}_{k+1}(S) = \mathbb{M}_{k'+1}(S') \;\Rightarrow\; S = \mathbb{S}_{k}\,\mathbb{M}_{k'+1}(S')$$

therefore,

$$\mathbb{V}_{k\to k'}(S) = S' \;\Rightarrow\; S = \mathbb{V}_{k'\to k}(S') \tag{3.1}$$

The above result is also immediate from the definition of $\mathbb{V}_{k\to k'}(S)$ in terms of the partition trees.

Recall that to apply operation $\mathbb{S}_k$ or $\mathbb{M}_{k'}$ to $S \in \mathbf{S}$ we must ensure that $s_k(S) \geq 1$ or $s_{k'}(S) \geq 2$, respectively. We now define two operations that extend $\mathbb{S}$ and $\mathbb{M}$, by relaxing the above preconditions. Recall that $\mu(S)$ denotes the minimum depth of blocks in S, and $\xi(S)$ denotes the corresponding maximum. For any $S \in \mathbf{S}$ and $k \geq \mu(S)$, we let

$$\hat{\mathbb{S}}_k(S) = \mathbb{S}_a(S), \quad \text{where } a = \max\{i \leq k \,:\, s_i(S) \geq 1\}.$$

Also, for any $S \in \mathbf{S}$ such that $|S| > 1$, and any $k' \leq \xi(S)$, we let

$$\hat{\mathbb{M}}_{k'}(S) = \mathbb{M}_{a'}(S), \quad \text{where } a' = \min\{i \geq k' \,:\, s_i(S) \geq 2\}.$$

(a' is well defined since $s_\xi(S) \geq 2$.) Note that if $s_k(S) \geq 1$ then $\hat{\mathbb{S}}_k(S) = \mathbb{S}_k(S)$, and if $s_{k'}(S) \geq 2$ then $\hat{\mathbb{M}}_{k'}(S) = \mathbb{M}_{k'}(S)$.

When we have nested application of $\mathbb{V}$, $\hat{\mathbb{S}}$, and $\hat{\mathbb{M}}$, we will use the same convention regarding brackets that we use for $\mathbb{S}$ and $\mathbb{M}$.

3.4 Comparing the balance of binary partitions: the $\succeq$ relation

A simple measure of the balance of a binary partition is its balance factor ϱ. (Recall that ϱ is the difference between the maximum depth ξ of blocks and the corresponding minimum μ.) For our analysis, however, we will need a more elaborate measure. We describe a *partial order* on $\mathbf{S}$, which provides for a very natural way to compare the balance of sorted binary partitions. Non sorted binary partitions can also be compared, by considering their respective sorted binary partitions.

We define the binary relation $\succeq$ on $\mathbf{S}_n$ (the set of all sorted binary partitions of size n) as follows. For any $S, S' \in \mathbf{S}_n$, $S \succeq S'$ if it is possible to obtain S' by applying zero or more move-down operations to S. Formally, $S \succeq S'$ iff one of the next two conditions holds:

- $S' = S$
- $S' = \mathbb{V}_{k_j \to k'_j} \cdots \mathbb{V}_{k_1 \to k'_1}(S)$, where $j \geq 1$ and $k_i < k'_i$ for all $i \in [1..j]$.[1]

Equivalently, $S \succeq S'$ if we can obtain S by applying zero or more move-up operations to S'. (The equivalence follows from (3.1).) We define relations $\preceq$, $\succ$, and $\prec$ on $\mathbf{S}_n$ in the obvious way.

Lemma 3.3. *$\succeq$ is a partial order.*

Proof. *Reflexivity* and *transitivity* are immediate from the definition of $\succeq$. *Antisymmetry* follows from the observation that a move-down operation *strictly increases* the average depth of nodes in the partition tree of the binary partition it is applied to. ■

The $\succeq$ relation provides for an intuitively reasonable means to compare the balance of sorted binary partitions. $S \succeq S'$ implies that S' can be obtained from S via a sequence of operations that make the partition tree of S progressively more unbalanced (by moving pairs of sibling leaves farther from the root). Therefore, it makes sense to say that if $S \succeq S'$ then "S is at least as balanced as S'." Note that it is not possible to compare the balance of binary partitions of *different* sizes based on the $\succeq$ relation. There are even pairs of sorted binary partitions of the *same* size that we cannot compare, as illustrated in Figure 3.2; i.e., $\succeq$ is not a *total* order. However, this approach suffices for the purposes of our analysis.

The next lemma says that if S is at least as balanced as S', with respect to $\succeq$, then the same is true when balance is measured in terms of ϱ.

[1] By $[i..i']$, for integers i and i', we denote the set $\{k \in \mathbb{Z} : i \leq k \leq i'\}$.

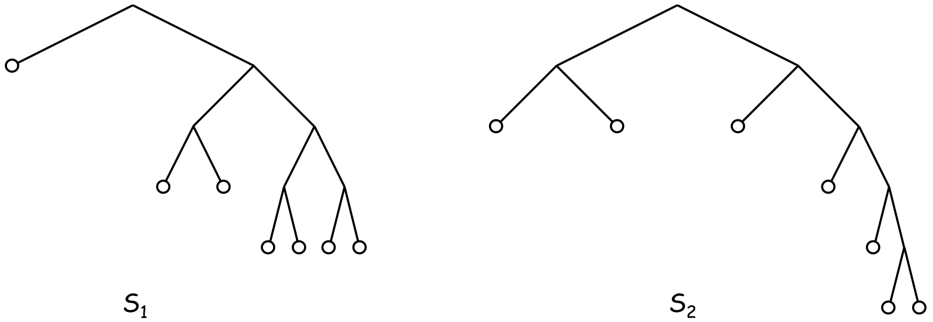

Figure 3.2: An example of two sorted binary partitions of the same size that are not comparable in terms of the $\succeq$ relation. $S_1 \not\succeq S_2$ follows from Lemma 3.4 and the fact that $\mu(S_1) < \mu(S_2)$; $S_2 \not\succeq S_1$ follows from Lemma 3.4 and fact that $\xi(S_2) > \xi(S_1)$.

Lemma 3.4. *If $S \succeq S'$ then $\mu(S) \geq \mu(S')$, $\xi(S) \leq \xi(S')$, and $\varrho(S) \leq \varrho(S')$.*

Proof. We show that the three inequalities above hold for the case where $S' = \mathbb{V}_{d \to d'}(S)$, for some $d < d'$. The general case follows then by induction. Recall that S' results from S by merging two blocks of depth $d + 1 \leq d'$, and splitting a block of depth d'. Clearly, these operations do not increase μ nor reduce ξ. Specifically,

$$\mu(S') = \min\{\mu(S), d\} \quad \text{and} \quad \xi(S') = \max\{\xi(S), d' + 1\}$$

So, the first two inequalities hold. The third follows directly from them. ■

An attractive property of $\succeq$ on which our analysis will be heavily based is that, roughly speaking, if $S \succeq S'$ and

- P results from S by applying to it an $\hat{\mathbb{S}}$ (or $\hat{\mathbb{M}}$) operation
- P' results from S' by applying to it a “similar or worse” $\hat{\mathbb{S}}$ (or $\hat{\mathbb{M}}$) operation

then $P \succeq P'$, as well. We describe this property formally in the two lemmata below.

Lemma 3.5.

(a) If $S \succeq S'$ and $k' \geq k \geq \mu(S)$ then $\hat{\mathbb{S}}_k(S) \succeq \hat{\mathbb{S}}_{k'}(S')$.

(b) If $S \succeq S'$ and $k' \leq k \leq \xi(S)$ then $\hat{\mathbb{M}}_k(S) \succeq \hat{\mathbb{M}}_{k'}(S')$.

Proof. Part (a) is immediate from Claims 3.6 and 3.7 that we prove below. The proof for part (b) is similar and is omitted.

Claim 3.6. *$\hat{\mathbb{S}}_k(S) \succeq \hat{\mathbb{S}}_{k'}(S)$.*

Proof. Let

$$a = \max\{i \le k \,:\, s_i(S) \ge 1\} \qquad a' = \max\{i \le k' \,:\, s_i(S) \ge 1\}$$

Since $k \le k'$, $a \le a'$.
If $a = a'$ then

$$\hat{\mathbb{S}}_k(S) = \mathbb{S}_a(S) = \mathbb{S}_{a'}(S) = \hat{\mathbb{S}}_{k'}(S)$$

If $a < a'$ then

$$\hat{\mathbb{S}}_k(S) = \mathbb{S}_a(S) = \mathbb{S}_a(\mathbb{M}_{a'+1}\,\mathbb{S}_{a'}(S)) = \mathbb{V}_{a' \to a}(\hat{\mathbb{S}}_{k'}(S)) \succeq \hat{\mathbb{S}}_{k'}(S)$$

where the second relation holds because $\mathbb{M}_{a'+1}\,\mathbb{S}_{a'}(S) = S$, by Lemma 3.2(a), and the last relation holds because $a' > a$. ■ {of Claim 3.6}

Claim 3.7. $\hat{\mathbb{S}}_{k'}(S) \succeq \hat{\mathbb{S}}_{k'}(S')$.

Proof. We consider only the case where $S' = \mathbb{V}_{d \to d'}(S)$, for some $d < d'$. The general case follows then by induction. Let

$$a = \max\{i \le k' \,:\, s_i(S) \ge 1\} \qquad a' = \max\{i \le k' \,:\, s_i(S') \ge 1\}$$

We distinguish three cases, depending on the values of k', d, and d'.

Case 1: $k' < d$.
For all $i < d$, $s_i(S') = s_i(S)$, so,

$$a = a' < d$$

We have

$$\begin{aligned}\hat{\mathbb{S}}_{k'}(S') &= \mathbb{S}_{a'}(S') = \mathbb{S}_{a'}(\mathbb{M}_{d+1}\,\mathbb{S}_{d'}(S)) = \mathbb{M}_{d+1}\,\mathbb{S}_{a'}\,\mathbb{S}_{d'}(S)\\ &= \mathbb{M}_{d+1}\,\mathbb{S}_{d'}\,\mathbb{S}_{a'}(S) = \mathbb{V}_{d \to d'}\,\hat{\mathbb{S}}_{k'}(S) \preceq \hat{\mathbb{S}}_{k'}(S)\end{aligned}$$

The last relation in the first line holds because of Lemma 3.2(e), since $a' < d$; in the second line, the first relation holds because of Lemma 3.2(c), since $a' < d < d' + 1$, the second because $a = a'$, and the last because $d < d'$.

Case 2: $d \le k' \le d'$.
Since $k' \le d'$, by Claim 3.6 we have

$$\hat{\mathbb{S}}_{k'}(S) \succeq \hat{\mathbb{S}}_{d'}(S) = \mathbb{S}_{d'}(S)$$

where the second relation holds because, by definition, $s_{d'}(S) > 0$. Similarly, since $k' \geq d$,

$$\hat{\mathbb{S}}_{k'}(S') \preceq \hat{\mathbb{S}}_d(S') = \mathbb{S}_d(S')$$

Also,

$$\mathbb{S}_d(S') = \mathbb{S}_d(\mathbb{M}_{d+1}\,\mathbb{S}_{d'}(S)) = \mathbb{S}_{d'}(S)$$

by Lemma 3.2(b). Combining the above three results, yields $\hat{\mathbb{S}}_{k'}(S') \preceq \hat{\mathbb{S}}_{k'}(S)$.

Case 3: *$k' > d'$.*
For all $i > d' + 1$, $s_i(S') = s_i(S)$; also $s_{d'+1}(S') > s_{d'+1}(S)$. From these two facts it follows that

$$a = a' \geq d' + 1 \quad \text{or} \quad a < d' + 1 = a'$$

If the first of the two conditions holds then the proof is very similar to that of Case 1 and is omitted. So, suppose that the second condition holds. Since $a \leq d'$, by Claim 3.6,

$$\hat{\mathbb{S}}_{k'}(S) = \hat{\mathbb{S}}_a(S) \succeq \hat{\mathbb{S}}_{d'}(S)$$

Similarly, since $k' > d'$,

$$\hat{\mathbb{S}}_{k'}(S') \preceq \hat{\mathbb{S}}_{d'}(S')$$

Also, by Case 2,

$$\hat{\mathbb{S}}_{d'}(S) \succeq \hat{\mathbb{S}}_{d'}(S')$$

Combining the above three results, yields $\hat{\mathbb{S}}_{k'}(S) \succeq \hat{\mathbb{S}}_{k'}(S')$.

■ {of Claim 3.7 and Lemma 3.5}

The next lemma is similar to Lemma 3.5, but it uses different conditions on the depths k and k'. Recall that $\ell_{<k}(B)$ and $\ell_{\leq k}(B)$ are the total lengths of the blocks in B that have depths $< k$ and $\leq k$, respectively. So, for a sorted binary partition, $\ell_{<k}$ is the left endpoint of the first block of depth $\geq k$, if $\mu \leq k \leq \xi$; and $\ell_{<k} = 1$, if $k > \xi$. Similarly, for $k \leq \xi$, $\ell_{\leq k}$ is the right endpoint of the last block of depth $\leq k$. In particular, if $s_k > 0$ then $\ell_{<k}$ is the left endpoint of the first block of depth k, and $\ell_{\leq k}$ is the right endpoint of the last block of depth k.

Lemma 3.8.

(a) If $S \succeq S'$ and $\ell_{<k}(S) < \ell_{\leq k'}(S')$ and $k \geq \mu(S)$ then $\hat{\mathbb{S}}_k(S) \succeq \hat{\mathbb{S}}_{k'}(S')$.

(b) If $S \succeq S'$ and $\ell_{\leq k}(S) > \ell_{<k'}(S')$ and $k \leq \xi(S)$ then $\hat{\mathbb{M}}_k(S) \succeq \hat{\mathbb{M}}_{k'}(S')$.

Proof. We only show part (a); the proof for part (b) is similar and is omitted. We begin by proving the following claim. Recall that, for $B \in \mathbf{B}$ and $x \in I$, $\text{blk}(B, x)$ is the block of B that contains x, and $\theta(B, x)$ is the depth of that block.

Claim 3.9. *For all $z \in I$, $\mathbb{S}_{\theta(S,z)}(S) \succeq \mathbb{S}_{\theta(S',z)}(S')$.*

Proof. We consider only the case where $S' = \mathbb{V}_{d \to d'}(S)$, for some $d < d'$. (The general case follows then by induction.) We distinguish two cases:
If $\text{blk}(S', z)$ is the last block in S' of depth d then $\theta(S, z) = d + 1$ and $\theta(S', z) = d$. So,

$$\mathbb{S}_{\theta(S',z)}(S') = \mathbb{S}_d(\mathbb{M}_{d+1}\,\mathbb{S}_{d'}(S)) = \mathbb{S}_{d'}(S) \preceq \mathbb{S}_{\theta(S,z)}(S)$$

where the second relation holds because of Lemma 3.2(b), and the last relation holds because of Lemma 3.5(a), since $\theta(S, z) = d + 1 \leq d'$.
If $\text{blk}(S', z)$ is *not* the last block in S' of depth d then it is easy to verify that $\theta(S, z) \leq \theta(S', z)$, and, so, the claim follows from Lemma 3.5(a). ■ {of Claim 3.9}

Let

$$z = \ell_{<k}(S)$$

If $s_k(S) > 0$ then $\theta(S, z) = k$, otherwise, $\theta(S, z) > k$; hence, in either case, $\theta(S, z) \geq k$. So, by Lemma 3.5(a),

$$\hat{\mathbb{S}}_k(S) \succeq \mathbb{S}_{\theta(S,z)}(S)$$

Since $z < \ell_{\leq k'}(S')$, we have that if $\ell_{<k'}(S') \leq z$ then $\theta(S', z) = k'$, otherwise, $\theta(S', z) < k'$; hence, $\theta(S', z) \leq k'$. So, by Lemma 3.5(a),

$$\hat{\mathbb{S}}_{k'}(S') \preceq \mathbb{S}_{\theta(S',z)}(S')$$

Combining the above two relations and Claim 3.9, yields the desired result. ■

3.5 Two random processes on binary partitions

We describe two families of random processes on binary partitions. The random processes of the first family, called $\mathcal{B}$-processes, model the evolution of the key-space partition of a DHT that uses the key-space partitioning scheme described in Section 2.3.1, under the model of adversarial arrivals/departures described in Section 3.1. The second family of random processes, called $\mathcal{S}$-processes, is a slight variation of the first one that considers *sorted* binary partitions (instead of arbitrary ones). We introduce $\mathcal{S}$-processes because they are easier to analyze, and their analysis yields performance bounds that apply to $\mathcal{B}$-processes, as well.

3.5.1 $\mathcal{B}$-processes

We begin with an informal description. A $\mathcal{B}$-process generates an infinite sequence $B_0, B_1, \ldots$ of binary partitions, where B_0 is provided as a parameter of the process, and for each $t \geq 1$, B_t is generated (in step t of the process) from B_{t-1} by applying to it either a single spltBlk or a single mrgBlk operation. The decision of which of the two types of operations will take place is the result of a random experiment that depends on the history of the process up to step $t-1$. The outcome of this experiment, denoted E_t, is either the $+$ symbol or a block in B_{t-1}. If $E_t = +$ then a spltBlk operation takes place; otherwise a mrgBlk operation occurs. The actual experiments used to determine the E_t, or, equivalently the distributions of the E_t conditioned on the process' history up to step $t-1$, is a parameter of the process. We can perceive this parameter as a randomized (adversarial) strategy for deciding the sequence of operations. The block that is split or the pair of blocks that is merged in step t is determined as follows. We sample a sequence $X_t = \langle X_{t,1}, X_{t,2}, \ldots \rangle$ of points in I, each point selected independently and uniformly at random. (More correctly, the points are chosen form a "quantization" of I, as we describe later.) If $E_t = +$, X_t consists of $\lambda_+(\delta)$ points, where δ is the depth of the block containing the first point $X_{t,1}$, and function λ_+ is a parameter of the process. The block that is split in this case is the largest among the blocks containing the points in X_t. If $E_t \neq +$ (thus, E_t is a block of B_{t-1}), X_t consists of $\lambda_-(\delta')$ points, where δ' is the depth of E_t, and function λ_- is also a parameter of the process. Let a be the smallest among E_t and the blocks containing the points in X_t. If the sibling a' of a is a leaf then a and a' are merged; otherwise, a pair of sibling leafs that are descendants of a' are merged, instead.

More formally, a $\mathcal{B}$-processes is parameterized by:

1. an *initial partition* $B_0 \in \mathbf{B}$;
2. a pair of *non-decreasing* functions $\lambda_+, \lambda_- : \mathbb{N} \to \mathbb{N}^*$, called the *sampling-size functions*; and
3. a family of distributions, called the *strategy of the adversary*; we elaborate on this later.

In each step $t = 1, 2, \ldots$ of the process, a binary partition B_t is randomly generated as follows. Assume the outcome of all random choices made before step t are known. First we choose E_t from the set $\{+\} \cup B_{t-1}$ according to a distribution specified by the strategy of the adversary. Next, we choose a sequence $X_t = \langle X_{t,1}, X_{t,2}, \ldots, X_{t,\Lambda_t} \rangle$ of points in I, where

$$\Lambda_t = \begin{cases} \lambda_+(\theta(B_{t-1}, X_{t,1})), & \text{if } E_t = + \\ \lambda_-(\theta(E_t)), & \text{if } E_t \in B_{t-1} \end{cases} \tag{3.2}$$

Each $X_{t,i}$ is chosen independently and uniformly at random from the set $I_{\xi(B_{t-1})}$. I_k, for $k \in \mathbb{N}$, denotes the set of all points in I whose binary representation has length (at most) k; i.e.,

$$I_k = \{j2^{-k} \,:\, j = 0, \ldots, 2^k - 1\}$$

Note that I_k consists of the left endpoints of all the binary segments of depth k. Finally, we set

$$B_t = \begin{cases} \textsc{addBlk}(B_{t-1}, X_t), & \text{if } E_t = + \\ \textsc{rmBlk}(B_{t-1}, E_t, X_t), & \text{if } E_t \in B_{t-1} \end{cases} \tag{3.3}$$

The functions ADDBLK and RMBLK are described in Figure 3.3.

The strategy of the adversary specifies, for each step t, the conditional distribution of E_t given the history of the process up to (and including) step $t-1$; i.e.,

$$\mathbb{Pr}[E_t = e \mid E_1, X_1, \ldots, E_{t-1}, X_{t-1}], \quad \text{for all } e \in \{+\} \cup \mathbf{\Gamma}$$

(Recall $\mathbf{\Gamma}$ denotes the set of binary segments.) We require that, for all t, either $E_t = +$ or E_t be a block of B_{t-1}; in particular, if B_{t-1} consists of a single binary segment then $E_t = +$. Formally,

$$\mathbb{Pr}[E_t \in \{+\} \cup B_{t-1}] = 1 \quad \text{and} \quad \mathbb{Pr}[\{E_t \neq +\} \cap \{|B_{t-1}| = 1\}] = 0$$

Note that choosing the $X_{t,i}$ independently and uniformly at random from the discrete interval $I_{\xi(B_{t-1})}$ is equivalent, with respect to the resulting B_t, to choosing the $X_{t,i}$ independently and uniformly at random from the continuous interval I. We use the first approach to avoid dealing with uncountably infinite probability spaces.

It is easy to see that a $\mathcal{B}$-process describes the evolution of the key-space partition of a DHT that employs the key-space partitioning scheme of Section 2.3.1, when arrivals and departures of nodes take place as described in Section 3.1. B_0 is the initial key-space partition, and B_t, for $t \geq 1$, is the corresponding partition right after the t-th operation (arrival or departure) has been completed. If $E_t = +$, the t-th operation is the arrival of a node; otherwise, it is a departure, and the departing node is that responsible for block E_t. The sampling-size functions correspond to the functions λ_+, λ_- the scheme uses to determine the number of random probes that should be performed in each operation. The sample points used in the random probes during the t-th operation are the elements of X_t. (As we discussed above choosing the $X_{t,i}$ from $I_{\xi(B_{t-1})}$ is effectively equivalent to choosing them from I.) Finally, note that the model of arrivals and departures is determined by the strategy of the adversary, which can be used to describe any *adaptive online adversary*, who makes decisions in each step t based on the past history of the process up to step $t-1$. In our analysis, we will *not* impose any restrictions on the strategy of the adversary.

We will refer to $\langle B_0, B_1, \ldots \rangle$ as the *partition-sequence* of the $\mathcal{B}$-process.

3.5.2 $\mathcal{S}$-processes

Again we start with an informal description. As in a $\mathcal{B}$-process, a sequence $S_0, S_1, \ldots$ of binary partitions is generated. Unlike a $\mathcal{B}$-process, however, the binary partitions are sorted. Also, the sequence has a finite length, which is a parameter of the process. S_0 is also provided; each S_t, for $t \neq 0$, is generated from S_{t-1} by applying to it either a single $\mathbb{S}$ or a single $\mathbb{M}$ operation. Again, the type of operation that will take place is the result of a random experiment that depends on the past history of the process and is a parameter of the process. The outcome of this experiment, denoted V_t, takes one of the two values $+$ or $-$. If $V_t = +$ an $\mathbb{S}$ operation takes place; if $V_t = -$ an $\mathbb{M}$ operation occurs. (More correctly, the outcome of the experiment is denoted U_t and takes values in some finite set $\mathbf{U}$; V_t is a function of U_t that takes values in $\{+, -\}$.) The depth of the block that is split or the depth of the blocks that are merged in step t is determined as follows. We sample a sequence $Y_t = \langle Y_{t,1}, Y_{t,2}, \ldots \rangle$ of points in I, each point selected independently and uniformly at random. (Again, for technical reasons, the points are actually chosen from a quantization of I. Also, the number of points that Y_t consists of is the same for all t.) If $V_t = +$, the depth of the block that is split is the largest integer d such that: (i) d is smaller or equal to the depths of the blocks containing the points $Y_{t,1}, \ldots, Y_{t,\lambda_+(d)}$, where λ_+ is a parameter of the process; and (ii) B_{t-1} contains at least one block of depth d. If $V_t = -$, instead, the depth of the blocks that are merged is the smallest integer d' such that: (i$'$) d' is greater or equal to the depths of the blocks containing the points $Y_{t,1}, \ldots, Y_{t,\lambda_-(\delta)}$, where δ is the minimum depth of blocks in B_{t-1}, and λ_- is a parameter of the process; and (ii$'$) B_{t-1} contains at least two blocks of depth d.

More precisely, like a $\mathcal{B}$-process, an $\mathcal{S}$-process is parameterized by:

1. an *initial partition* $S_0 \in \mathbf{S}$;
2. a pair of non-decreasing *sampling-size functions* $\lambda_+, \lambda_- : \mathbb{N} \rightarrow \mathbb{N}^*$; and

In the definitions below:

- $B \in \mathbf{B}$; $b \in B$; and $Z = \langle z_1, z_2, \ldots \rangle$ is a *sufficiently long* sequence of $z_i \in I$, i.e., $|Z| \geq \lambda_+(\theta(B, z_1))$ for the first two definitions, and $|Z| \geq \lambda_-(\theta(b))$ for the other two
- functions $\textsc{spltBlk}$ and $\textsc{mrgBlk}$ are defined in Section 3.3
- in lines 6, 12, and 16 ties are broken in an arbitrary but deterministic way

```
 1: function $\textsc{addBlk}(B, Z)$
 2:     $b \leftarrow \textsc{getLrgBlk}(B, Z)$
 3:     return $\textsc{spltBlk}(B, b)$

 4: function $\textsc{getLrgBlk}(B, Z)$
 5:     $\delta \leftarrow \theta(B, z_1)$
 6:     return a largest block in $\{\text{blk}(B, z_i) : i = 1, \ldots, \lambda_+(\delta)\}$

 7: function $\textsc{rmBlk}(B, b, Z)$
 8:     $a \leftarrow \textsc{getSmlBlk}(B, b, Z)$
 9:     if $|b| \leq |a|$ then
10:         $a \leftarrow b$
11:     if $\text{sbl}(a) \notin B$ then
12:         $a \leftarrow$ a block $c \in B$ such that $\text{sbl}(c) \in B$ and $c \subseteq \text{sbl}(b)$
13:     return $\textsc{mrgBlk}(B, a)$

14: function $\textsc{getSmlBlk}(B, Z)$
15:     $\delta \leftarrow \theta(b)$
16:     return a smallest block in $\{\text{blk}(B, z_i) : i = 1, \ldots, \lambda_-(\delta)\}$
```

Figure 3.3: Functions used in the definition of a $\mathcal{B}$-processes.

In the definitions below:

- $S \in \mathbf{S}$; and $Z = \langle z_1, z_2, \ldots \rangle$ is a *sufficiently long* sequence of $z_i \in I$, i.e., $|Z| \geq \lambda_+(\xi(S))$ for the first two definitions, and $|Z| \geq \lambda_-(\mu(S))$ for the other two
- functions $\hat{\mathbb{M}}_k$ and $\hat{\mathbb{S}}_k$ are defined in Section 3.3

```
17: function $\textsc{addBlk}_{\mathbf{S}}(S, Z)$
18:     $k \leftarrow \textsc{getSmlDpth}(S, Z)$
19:     return $\hat{\mathbb{S}}_k(S)$

20: function $\textsc{getSmlDpth}(S, Z)$
21:     return $\max\big\{j : j \leq \min\{\theta(S, z_i) : i = 1, \ldots, \lambda_+(j)\}\big\}$

22: function $\textsc{rmBlk}_{\mathbf{S}}(S, Z)$
23:     $k \leftarrow \textsc{getLrgDpth}(S, Z)$
24:     return $\hat{\mathbb{M}}_k(S)$

25: function $\textsc{getLrgDpth}(S, Z)$
26:     return $\max\{\theta(S, z_i) : i = 1, \ldots, \lambda_-(\mu(S))\}$
```

Figure 3.4: Functions used in the definition of an $\mathcal{S}$-processes.

3. a *strategy of the adversary*, which we explain later.

Unlike $\mathcal{B}$-processes, $\mathcal{S}$-processes are *finite*. Also the number of points sampled in each step and the space from which they are chosen is *the same* for all steps. To address these issues, two additional parameters are used:

4. a *length* $N \in \mathbb{N}$, that is the number of steps the process involves; and
5. a *precision* $g \in \mathbb{N}$, such that
$$g \geq |S_0| + N \tag{3.4}$$
that is the (maximum) length in bits of the points sampled.

In each step $t = 1, 2, \ldots, N$ of the process, a sorted binary partition S_t is randomly generated as follows. Assume the outcome of all random choices made before step t are known. First we choose U_t according to a finite distribution specified by the strategy of the adversary. From U_t we then compute $V_t = V_t(U_t)$, also as described in the strategy of the adversary; V_t takes values in the set $\{+, -\}$. Next, we choose a sequence $Y_t = \langle Y_{t,1}, Y_{t,2}, \ldots, Y_{t,\Lambda}\rangle$ of points in I, where
$$\Lambda = \lambda_+(g) + \lambda_-(g) \tag{3.5}$$
and each $Y_{t,i}$ is chosen independently and uniformly at random from I_g. Finally, we set
$$S_t = \begin{cases} \textsc{addBlk}_{\mathbf{S}}(S_{t-1}, Y_t), & \text{if } V_t = + \\ \textsc{rmBlk}_{\mathbf{S}}(S_{t-1}, Y_t), & \text{if } V_t = - \end{cases} \tag{3.6}$$
where functions $\textsc{addBlk}_{\mathbf{S}}$ and $\textsc{rmBlk}_{\mathbf{S}}$ are described in Figure 3.4.

The definitions of functions $\textsc{addBlk}_{\mathbf{S}}$ and $\textsc{rmBlk}_{\mathbf{S}}$ are similar to those of $\textsc{addBlk}$ and $\textsc{rmBlk}$, in Figure 3.3, adapted for *sorted* binary partitions. We now briefly explain the definition of $\textsc{getSmlDpth}(S, Z)$ — the rest of the definitions are straightforward. Let $Q(j)$, for $j \in \mathbb{N}$, denote the assertion: j is smaller or equal to the depths of all the blocks in S that contain the points $z_1, \ldots, z_{\lambda_+(j)}$. Note that $Q(j)$ may be true only for $j \leq \xi(S)$, since $\lambda_+(j) \geq 1$ and $\theta(S, z_1) \leq \xi(S)$. Also, since λ_+ is non-decreasing, if $Q(j)$ holds then $Q(j')$ holds for all $j' < j$, as well. $\textsc{getSmlDpth}(S, Z)$ returns the largest j such that $Q(j)$ holds.

The strategy of the adversary specifies, for each step $t \in [1..N]$, the conditional distribution of U_t given the history of the process up to step $t-1$; i.e.,
$$\mathbb{Pr}[U_t = u \mid U_1, Y_1, \ldots, U_{t-1}, Y_{t-1}], \quad \text{for all } u \in \mathbf{U}$$
where $\mathbf{U}$ is some finite set. It also describes a mapping $\varphi_t : \mathbf{U} \to \{+, -\}$; we let $V_t = \varphi(U_t)$. The strategy of the adversary should ensure that if $|B_{t-1}| = 1$ then $V_t = +$. Note that we

could have defined the strategy of the adversary in a more direct way, by having U_t be $+$ or $-$ and using U_t in place of V_t (i.e., letting $\mathbf{U} = \{+, -\}$ and φ_t be the identity function). The approach we use, however, simplifies the coupling of $\mathcal{B}$-processes and $\mathcal{S}$-processes we describe in Section 3.6.

Unlike the X_t in the definition of a $\mathcal{B}$-process, whose size and the space from which their elements are drawn vary with t, all the Y_t have the same size Λ, and their elements are drawn from the same set I_g. Note that, by (3.4), $g > \xi(S_t)$, for all $t \leq N$. Thus, each Y_t contains at least as many elements as required by the $\textsc{addBlk}_{\mathbf{S}}$ or $\textsc{rmBlk}_{\mathbf{S}}$ operation executed at step t, and the resulting S_t has the same distribution as if the $Y_{t,i}$ were drawn uniformly from I. We follow this approach, instead of defining the Y_t similarly to the X_t, because it is more convenient for our analysis. Notice that this choice dictates that $\mathcal{S}$-processes be finite — unlike $\mathcal{B}$-processes.

As in a $\mathcal{B}$-process, we will refer to $\langle S_0, \ldots, S_N \rangle$ as the *partition-sequence* of the $\mathcal{S}$-process.

We will use the term *random point-vector* to denote a vector $\langle Z_1, \ldots, Z_m \rangle$, where each Z_i is chosen independently and uniformly at random from I_k, for some $k \in \mathbb{N}$; we call m and k the *height* and *precision* of the random point-vector, respectively. Also, we will use the term *random point-array* to denote a vector of d independent random point-vectors of the same height and precision; we call d the *length* of the random point-array. The *height* and *precision* of a random point-array are the height and precision of its elements. For example, each Y_t is a random point-vector of height Λ and precision g, and $\langle Y_1, \ldots, Y_N \rangle$ is a random point-array of length N.

Remark: In our analysis of $\mathcal{S}$-processes, in Chapters 4 and 5, we will often use the phrase "long enough $\mathcal{S}$-process" in the context: "for all long enough $\mathcal{S}$-processes, a certain condition is met (with some probability) within a given number steps, say k." This should be interpreted as: "for all $\mathcal{S}$-process of length $N \geq k$, a certain condition is met..."

We will use a similar convention regarding random point-vectors/arrays. Typically, a random point-vector will be used as an argument to an $\textsc{addBlk}_{\mathbf{S}}$ and/or $\textsc{rmBlk}_{\mathbf{S}}$ operation for some $S \in \mathbf{S}$, and some underlying sampling-size functions. In this context, we will say a "large enough random point-vector" to denote that its height is at least equal to that required to perform the operation, and its precision is at least equal to $\xi(S)$. Likewise, we will typically use the elements of a random point-array as arguments in a sequence of $\textsc{addBlk}_{\mathbf{S}}$ and/or $\textsc{rmBlk}_{\mathbf{S}}$ operations, with respect to some starting $S \in \mathbf{S}$, and some sampling-size functions. In such a setting, a "large enough random point-array" is one whose length is at least equal to the number of operations to be performed, its height is at least equal to that required to perform each operation, and its precision is at least equal to the largest ξ of the binary partitions the operations are applied to.

3.6 Bounding the balance in a $\mathcal{B}$-process by that in an $\mathcal{S}$-process

In this section, we relate the balance of the binary partitions in a $\mathcal{B}$-process to the balance of the sorted binary partitions in some $\mathcal{S}$-process. Roughly speaking, we show that given any $\mathcal{B}$-process, we can construct a parallel $\mathcal{S}$-process that has "similar" parameters and its random choices depend on those of the $\mathcal{B}$-process, such that the partition-sequence of the $\mathcal{B}$-process is more balanced than that of the $\mathcal{S}$-process. The rigorous statement of this result, as Theorem 3.10 below, uses the concept of *coupling* [79]. Informally, a coupling of two random elements X, Y (defined on different probability spaces) is another pair of random elements $\langle \hat{X}, \hat{Y} \rangle$, defined on the same probability space, such that $\hat{X}$ has the same distribution as X and $\hat{Y}$ the same as Y.

Theorem 3.10. *For any $\mathcal{B}$-process $\mathcal{P}_{\mathcal{B}}$ and any $n \in \mathbb{N}$, there is an $\mathcal{S}$-process $\mathcal{P}_{\mathcal{S}}$ such that*

(i) $S_0 = \mathrm{srt}(B_0)$, $N = n$, and both $\mathcal{P}_{\mathcal{B}}, \mathcal{P}_{\mathcal{S}}$ have the same λ_+, λ_-; and

(ii) there is a coupling $\langle \hat{\mathcal{P}}_{\mathcal{B}}, \hat{\mathcal{P}}_{\mathcal{S}} \rangle$ of $\mathcal{P}_{\mathcal{B}}, \mathcal{P}_{\mathcal{S}}$ such that, for all $t \in [0..n]$, $\mathrm{srt}(\hat{B}_t) \succeq \hat{S}_t$.

($\hat{B}_t$ and $\hat{S}_t$ denote the binary partitions in $\hat{\mathcal{P}}_{\mathcal{B}}$ and $\hat{\mathcal{P}}_{\mathcal{S}}$, respectively.)

We will use Theorem 3.10 later in our analysis (in Chapter 6) to argue that certain probabilistic bounds with respect to balance we show for $\mathcal{S}$-processes (in Chapters 4 and 5), apply to $\mathcal{B}$-processes as well. We describe the proof of Theorem 3.10 in Section 3.6.2. Before that, in Section 3.6.1, we define the concept of an f_B mapping, which we will use in the coupling construction.

3.6.1 The f_B mapping

Consider the one-to-one mapping between the blocks of $B \in \mathbf{B}$ and those of $\mathrm{srt}(B)$ that pairs blocks of the same depth, while preserving the relative order of blocks of the same depth. More formally, for any $B \in \mathbf{B}$, let $F_B : B \to \mathrm{srt}(B)$ such that

(i) for all $b \in B$, $\theta(F_B(b)) = \theta(b)$, and

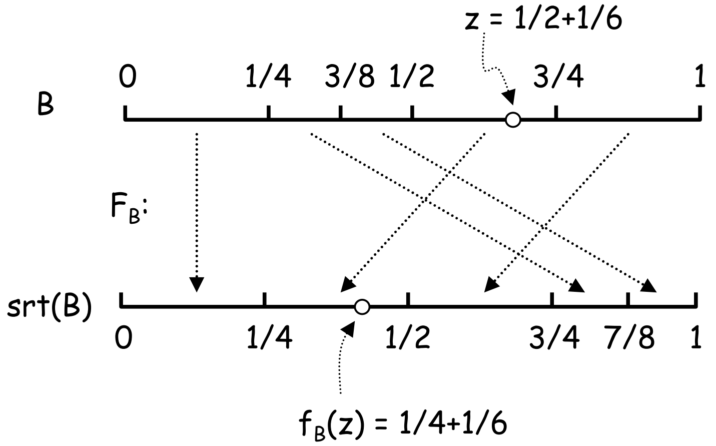

Figure 3.5: Examples of F_B and f_B.

(ii) for all $b, b' \in B$ such that $\theta(b) = \theta(b')$, if $b.x < b'.x$ then $F_B(b).x < F_B(b').x$.

By $a.x$ we denote the left endpoint of $a \in \mathbf{\Gamma}$.

We let f_B be the one-to-one mapping from I to itself induced by the "reordering" of the blocks in B described by F_B. More formally, $f_B : I \to I$ such that, for every $z \in I$,

$$f_B(z) = F_B(\mathrm{blk}(B, z)).x + (z - \mathrm{blk}(B, z).x)$$

An example of the above definitions is illustrated in Figure 3.5. The next lemma states a simple property of f_B. The proof is straightforward and is omitted.

Lemma 3.11. *For all $B \in \mathbf{B}$ and $z \in I$, $\theta(\mathrm{srt}(B), f_B(z)) = \theta(B, z)$.*

For any $Z = \langle z_1, \dots, z_m \rangle \in I^m$, we will write $f_B(Z)$ to denote the vector $\langle f_B(z_1), \dots, f_B(z_m) \rangle$.

3.6.2 Proof of Theorem 3.10

Condition (i) describes three of the five parameters of $\mathcal{P}_{\mathcal{S}}$; so, we need to specify the other two. $\mathcal{P}_{\mathcal{S}}$'s precision g can be arbitrary — as long as it satisfies (3.4). The strategy of the adversary will become apparent from the coupling, so, we postpone its description.

Before we describe the coupling, we state a simple result we exploit in the construction. Its proof is straightforward and is omitted.

Lemma 3.12. *Let $Z = \langle Z_1, \dots, Z_m \rangle$ be a random point-vector of precision κ. Then,*

(a) For any $k \in \mathbb{N}$ with $k \le \kappa$, $\left\langle \lfloor Z_1 2^k \rfloor / 2^k, \dots, \lfloor Z_m 2^k \rfloor / 2^k \right\rangle$ is a random point-vector of precision k.

(b) For any $B \in \mathbf{B}$ with $\xi(B) \leq \kappa + 1$, $f_B(Z)$ is a random point-vector of precision κ.

We now describe the coupling construction. We use "hatted" notation to denote quantities associated with $\hat{\mathcal{P}}_{\mathcal{B}}$ and $\hat{\mathcal{P}}_{\mathcal{S}}$, to distinguish them from the corresponding quantities of $\mathcal{P}_{\mathcal{B}}$ and $\mathcal{P}_{\mathcal{S}}$. The construction proceeds in steps, where each step t decides the values of $\hat{E}_t, \hat{X}_t, \hat{B}_t$ and $\hat{U}_t, \hat{V}_t, \hat{Y}_t, \hat{S}_t$. We let

$$\hat{B}_0 = B_0 \qquad \hat{S}_0 = S_0 = \mathrm{srt}(B_0)$$

In each step $t = 1, \ldots, N$, first we choose $\hat{E}_t$ based on the $\hat{E}_1, \hat{X}_1, \ldots, \hat{E}_{t-1}, \hat{X}_{t-1}$, according to $\mathcal{P}_{\mathcal{B}}$'s strategy of the adversary. We also let

$$\hat{U}_t = \begin{cases} \hat{E}_t, & \text{if } t = 1 \\ \langle \hat{X}_{t-1}, \hat{B}_{t-1}, \hat{E}_t \rangle, & \text{if } t \neq 1 \end{cases} \qquad \hat{V}_t = \begin{cases} +, & \text{if } \hat{E}_t = + \\ -, & \text{otherwise} \end{cases}$$

Then, we choose $\mathcal{X}_t = \langle \mathcal{X}_{t,1}, \ldots, \mathcal{X}_{t,\Lambda} \rangle$ independently and uniformly at random from I_g^{Λ}. (Λ was defined in (3.5).) We set

$$\hat{X}_t = \left\langle \lfloor \mathcal{X}_{t,i} 2^{\xi(\hat{B}_{t-1})} \rfloor / 2^{\xi(\hat{B}_{t-1})} \right\rangle_{i=1}^{\hat{\Lambda}_t} \qquad \hat{Y}_t = f_{\hat{B}_{t-1}}(\mathcal{X}_t)$$

where $\hat{\Lambda}_t$ is defined as in (3.2). Note that, for all $t \in [1..N]$,

$$\hat{\Lambda}_t \leq \Lambda \qquad \xi(\hat{B}_{t-1}) \leq g$$

Finally, $\hat{B}_t$ and $\hat{S}_t$ are determined as in (3.3) and (3.6), respectively.
For $t = N+1, N+2, \ldots$, we choose $\hat{E}_t, \hat{X}_t, \hat{B}_t$ according to the definition of a $\mathcal{B}$-process.

We show now that $\hat{\mathcal{P}}_{\mathcal{B}}$ and $\hat{\mathcal{P}}_{\mathcal{S}}$ have the desired (marginal) distributions. That $\hat{\mathcal{P}}_{\mathcal{B}}$ has the same distribution as $\mathcal{P}_{\mathcal{B}}$ is immediate from the following fact. For each $t \in [1..N]$, conditioned on the event $\{\hat{\Lambda}_t = k\} \cap \{\xi(\hat{B}_{t-1}) = d\}$, for k and d such that this event occurs with positive probability, $\hat{X}_t$ is a random point-vector of height k and precision d. This fact follows from Lemma 3.12(a) and the observation we made earlier that $\xi(\hat{B}_{t-1}) \leq g$. To show that $\hat{\mathcal{P}}_{\mathcal{S}}$ has the same distribution as $\mathcal{P}_{\mathcal{S}}$ (for some strategy of the adversary for $\mathcal{P}_{\mathcal{S}}$), we need to show that:

1. $\hat{Y}_t$ is a random point-vector of height Λ and precision g, and it is independent of $\hat{U}_1, \hat{Y}_1, \ldots, \hat{U}_{t-1}, \hat{Y}_{t-1}, \hat{U}_t$.
2. $\hat{U}_1, \ldots, \hat{U}_N$ corresponds to a valid strategy of the adversary.

By Lemma 3.12(b), we have that, conditioned on any value for $\hat{B}_{t-1}$, $\hat{Y}_t$ is a random point-vector of height Λ and precision g. From this it is immediate that 1 holds. To show 2 we describe the strategy of the adversary for $\mathcal{P}_{\mathcal{S}}$ that is induced by the coupling construction. Essentially, this strategy "simulates" a copy of $\mathcal{P}_{\mathcal{B}}$ that is coupled with $\mathcal{P}_{\mathcal{S}}$, in the same way that $\hat{\mathcal{P}}_{\mathcal{B}}$ is with $\hat{\mathcal{P}}_{\mathcal{S}}$. We will use primed notation to denote quantities associated with this copy of $\mathcal{P}_{\mathcal{B}}$. In the first step of $\mathcal{P}_{\mathcal{S}}$, we choose $U_1 = E'_1$ according to the strategy of the adversary of $\mathcal{P}_{\mathcal{B}}$. In each step $t \in [2..N]$, we choose $U_t = \langle X'_{t-1}, B'_{t-1}, E'_t \rangle$ as follows. First we let

$$X'_{t-1} = \big\langle \lfloor f^{-1}_{B'_{t-2}}(Y_{t-1,i}) \cdot 2^{\xi(\hat{B}_{t-2})} \rfloor / 2^{\xi(\hat{B}_{t-2})} \big\rangle_{i=1}^{\Lambda'_{t-1}}$$

Note that B'_{t-2} and E'_{t-1} were decided in step $t-1$ when U_{t-1} was chosen. B'_{t-1} is then computed from X'_{t-1} B'_{t-2}, and E'_{t-1} — the last two are elements of U_{t-1}. Finally, E'_t is chosen according to the strategy of the adversary of $\mathcal{P}_{\mathcal{B}}$, based on $E'_1, X'_1, \ldots, E'_{t-1}, X'_{t-1}$ — which, except for X'_{t-1}, are elements of $U_1, \ldots, U_{t-1}$.

To complete the proof of the theorem it remains to show that

$$\mathrm{srt}(\hat{B}_t) \succeq \hat{S}_t, \quad \text{for all } t \in [0..N] \tag{3.7}$$

For this, we will use the next two lemmata, whose proofs are given at the end of this section.

Lemma 3.13. *For all $B \in \mathbf{B}$, $b \in B$, and $Z \in I^m$, for a large enough m,*

(a) $\mathrm{srt}(\textsc{addBlk}(B, Z)) \succeq \textsc{addBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z))$

(b) $\mathrm{srt}(\textsc{rmBlk}(B, b, Z)) \succeq \textsc{rmBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z))$

Lemma 3.14. *If $S \succeq S'$ and $Z \in I^m$, for a large enough m,*

(a) $\textsc{addBlk}_{\mathbf{S}}(S, Z) \succeq \textsc{addBlk}_{\mathbf{S}}(S', Z)$

(b) $\textsc{rmBlk}_{\mathbf{S}}(S, Z) \succeq \textsc{rmBlk}_{\mathbf{S}}(S', Z)$

We show (3.7) by induction on t. For $t = 0$, it is $\hat{S}_0 = \mathrm{srt}(B_0) = \mathrm{srt}(\hat{B}_0)$. For $1 \le t \le N$, we distinguish two cases, depending on the value of $\hat{E}_t$.
If $\hat{E}_t = +$ then,

$$\mathrm{srt}(\hat{B}_t) = \mathrm{srt}(\textsc{addBlk}(\hat{B}_{t-1}, \hat{X}_t)) = \mathrm{srt}(\textsc{addBlk}(\hat{B}_{t-1}, \mathcal{X}_t))$$

So, by Lemma 3.13(a),

$$\mathrm{srt}(\hat{B}_t) \succeq \textsc{addBlk}_{\mathbf{S}}(\mathrm{srt}(\hat{B}_{t-1}), f_{\hat{B}_{t-1}}(\mathcal{X}_t))$$

By the induction hypothesis, $\mathrm{srt}(\hat{B}_{t-1}) \succeq \hat{S}_{t-1}$, so, by applying Lemma 3.14(a) to the right-hand side of the above relation, we obtain

$$\mathrm{srt}(\hat{B}_t) \succeq \textsc{addBlk}_{\mathbf{S}}(\hat{S}_{t-1}, f_{\hat{B}_{t-1}}(\mathcal{X}_t)) = \hat{S}_t$$

where the second relation holds because $\hat{V}_t = +$ (since $\hat{E}_t = +$).

If $\hat{E}_t \neq +$ the proof is similar, using Lemmata 3.13(b) and 3.14(b) in place of Lemmata 3.13(a) and 3.14(a), respectively.

Proof of Lemma 3.13

We begin by proving the following claim.

Claim 3.15.

(a) $\theta(\textsc{getLrgBlk}(B, Z)) \leq \textsc{getSmlDpth}(\mathrm{srt}(B), f_B(Z))$

(b) $\theta(\textsc{getSmlBlk}(B, Z)) \geq \textsc{getLrgDpth}(\mathrm{srt}(B), f_B(Z))$

Proof. Let $k = \theta(\textsc{getLrgBlk}(B, Z))$, and $Z = \langle z_1, \ldots, z_m \rangle$. By the definition of function $\textsc{getLrgBlk}$,

$$k = \min\{\theta(B, z_i) \,:\, i \leq \lambda_+(\theta(B, z_1))\}$$

So, $k \leq \theta(B, z_1)$, and, thus, (since λ_+ is non-decreasing)

$$k \leq \min\{\theta(B, z_i) \,:\, i \leq \lambda_+(k)\} \tag{3.8}$$

From the definition of $\textsc{getSmlDpth}$,

$$\begin{aligned}\textsc{getSmlDpth}(\mathrm{srt}(B), f_B(Z)) &= \max\{j \,:\, j \leq \min\{\theta(\mathrm{srt}(B), f_B(z_i)) \,:\, i \leq \lambda_+(j)\}\} \\ &= \max\{j \,:\, j \leq \min\{\theta(B, z_i) \,:\, i \leq \lambda_+(j)\}\}\end{aligned}$$

by Lemma 3.11. Combining this and (3.8), yields

$$\textsc{getSmlDpth}(\mathrm{srt}(B), f_B(Z)) \geq k$$

Hence, part (a) holds.

For part (b), from the definitions of $\textsc{getSmlBlk}$ and $\textsc{getLrgDpth}$ we have

$$\begin{aligned}\theta(\textsc{getSmlBlk}(B, Z)) &= \max\{\theta(B, z_i) \,:\, i \leq \lambda_-(\theta(b))\} \\ &\geq \max\{\theta(B, z_i) \,:\, i \leq \lambda_-(\mu)\} \\ &= \max\{\theta(\mathrm{srt}(B), f_B(z_i)) \,:\, i \leq \lambda_-(\mu)\} \\ &= \textsc{getLrgDpth}(\mathrm{srt}(B), f_B(Z))\end{aligned}$$

where the second line holds because $\theta(b) \geq \mu$ and λ_- is non-decreasing; and the third line holds because of Lemma 3.11. ■

We now show part (a) of the lemma. By the definition of $\textsc{addBlk}$,

$$\textsc{addBlk}(B, Z) = \textsc{spltBlk}(B, b), \quad \text{where } b = \textsc{getLrgBlk}(B, Z)$$

So,

$$\mathrm{srt}(\textsc{addBlk}(B, Z)) = \mathrm{srt}(\textsc{spltBlk}(B, b)) = \mathbb{S}_{\theta(b)}(\mathrm{srt}(B)) = \hat{\mathbb{S}}_{\theta(b)}(\mathrm{srt}(B)) \tag{3.9}$$

where the middle relation holds because of Lemma 3.1(a). By the definition of $\textsc{addBlk}_{\mathbf{S}}$,

$$\textsc{addBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z)) = \hat{\mathbb{S}}_k(\mathrm{srt}(B)), \tag{3.10}$$

where

$$k = \textsc{getSmlDpth}(\mathrm{srt}(B), f_B(Z))$$

By Claim 3.15(a), we have that $k \geq \theta(b)$; so, by applying Lemma 3.5(a) to the right-hand side of (3.10) we obtain

$$\textsc{addBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z)) \preceq \hat{\mathbb{S}}_{\theta(b)}(\mathrm{srt}(B))$$

Combining this and (3.9), yields part (a).

Next, we show part (b). From the definition of $\textsc{rmBlk}$

$$\textsc{rmBlk}(B, b, Z) = \textsc{mrgBlk}(B, a)$$

for some $a \in B$ such that $\mathrm{sbl}(a) \in B$ and

$$\theta(a) \geq \theta(\textsc{getSmlBlk}(B, Z)) \tag{3.11}$$

So,

$$\mathrm{srt}(\textsc{rmBlk}(B, b, Z)) = \mathrm{srt}(\textsc{mrgBlk}(B, a)) = \mathbb{M}_{\theta(a)}(\mathrm{srt}(B)) = \hat{\mathbb{M}}_{\theta(a)}(\mathrm{srt}(B)) \tag{3.12}$$

where the middle relation holds because of Lemma 3.1(b). From the definition of $\textsc{rmBlk}_{\mathbf{S}}$,

$$\textsc{rmBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z)) = \hat{\mathbb{M}}_k(\mathrm{srt}(B)) \tag{3.13}$$

where

$$k = \textsc{getLrgDpth}(\mathrm{srt}(B), f_B(Z))$$

By Claim 3.15(b), we have that $k \leq \theta(\textsc{getSmlBlk}(B, Z))$, so, by (3.11), $k \leq \theta(a)$. Hence, by applying Lemma 3.5(b) to the right-hand side of (3.13) we get

$$\textsc{rmBlk}_{\mathbf{S}}(\mathrm{srt}(B), f_B(Z)) \preceq \hat{\mathbb{M}}_{\theta(a)}\, \mathrm{srt}(B)$$

which, combined with (3.12), yields part (b).

Proof of Lemma 3.14

We begin by proving the following claim.

Claim 3.16.

(a) if $\textsc{getSmlDpth}(S, Z) > \textsc{getSmlDpth}(S', Z)$ *then* $\ell_{<\textsc{getSmlDpth}(S,Z)}(S) < \ell_{\leq\textsc{getSmlDpth}(S',Z)}(S')$

(b) $\ell_{\leq\textsc{getLrgDpth}(S,Z)}(S) > \ell_{<\textsc{getLrgDpth}(S',Z)}(S')$

Proof. Let $Z = \langle z_1, \ldots, z_m\rangle$. For part (a), let also

$$k = \textsc{getSmlDpth}(S, Z) \qquad k' = \textsc{getSmlDpth}(S', Z)$$

From the definition of $\textsc{getSmlDpth}$ we have

$$k \leq \min\{\theta(S, z_i) \,:\, i \leq \lambda_+(k)\} = \theta(S, z^*)$$

where

$$z^* = \min\{z_i \,:\, i \leq \lambda_+(k)\}$$

Therefore,

$$\ell_{<k}(S) \leq \ell_{<\theta(S,z^*)}(S) \leq z^* \tag{3.14}$$

From the definition of $\textsc{getSmlDpth}$ we also have that

$$k' \geq \min\{\theta(S', z_i) \,:\, i \leq \lambda_+(k'+1)\} \geq \min\{\theta(S', z_i) \,:\, i \leq \lambda_+(k)\} = \theta(S', z^*)$$

where the second relation holds because $k > k'$ (or, equivalently, $k \geq k' + 1$) and λ_+ is non-decreasing. Therefore,

$$\ell_{\leq k'}(S') \geq \ell_{\leq\theta(S',z^*)}(S') > z^*$$

Combining this with (3.14) yields $\ell_{<k}(S) < \ell_{\leq k'}(S')$, as desired.
For part (b), let

$$d = \textsc{getLrgDpth}(S, Z) \qquad d' = \textsc{getLrgDpth}(S', Z)$$

From the definition of $\textsc{getLrgDpth}$,

$$d = \max\{\theta(S, z_i) \,:\, i \leq \lambda_-(\mu(S))\} = \theta(S, \hat{z})$$

where

$$\hat{z} = \max\{z_i \,:\, i \leq \lambda_-(\mu(S))\}$$

Therefore,

$$\ell_{\leq d}(S) > \hat{z} \tag{3.15}$$

Similarly,

$$d' = \max\{\theta(S', z_i) \,:\, i \leq \lambda_-(\mu(S'))\} \leq \max\{\theta(S', z_i) \,:\, i \leq \lambda_-(\mu(S))\} = \theta(S', \hat{z})$$

where the second relation holds because $\mu(S) \geq \mu(S')$ (by Lemma 3.4) and λ_- is non-decreasing. So,

$$\ell_{<d'}(S') \leq \hat{z}$$

Combining this and (3.15) yields part (b). ■

We can now show part (a) of the lemma as follows. By the definition of $\textsc{addBlk}_{\mathbf{S}}$,

$$\textsc{addBlk}_{\mathbf{S}}(S, Z) = \hat{\mathbb{S}}_k(S), \quad \text{where } k = \textsc{getSmlDpth}(S, Z)$$

and

$$\textsc{addBlk}_{\mathbf{S}}(S', Z) = \hat{\mathbb{S}}_{k'}(S'), \quad \text{where } k' = \textsc{getSmlDpth}(S', Z)$$

By Claim 3.16(a), we have that $k \leq k'$ or $\ell_{<k}(S) < \ell_{\leq k'}(S')$. So, by Lemma 3.5(a) (if $k \leq k'$) and by Lemma 3.8(a) (if $\ell_{<k}(S) < \ell_{\leq k'}(S')$), we have

$$\hat{\mathbb{S}}_k \, S \succeq \hat{\mathbb{S}}_{k'} \, S'$$

Therefore, $\textsc{addBlk}_{\mathbf{S}}(S, k) \succeq \textsc{addBlk}_{\mathbf{S}}(S', k')$.

The proof of part (b) is similar, using Claim 3.16(b) in place of Claim 3.16(a), and Lemma 3.8(b) instead of Lemmata 3.5(a) and 3.8(a).

Chapter 4

Analysis – Part II: Starting from a balanced partition

In this chapter and the next we study the balance of the partitions in an $\mathcal{S}$-process.[1] Specifically, in this chapter we consider the case where the initial partition of the $\mathcal{S}$-process is "very balanced": either it has $\varrho < 2$, or $\varrho = 2$ and most blocks are of intermediate size; we call such a partition *safe*. (Recall that $\varrho = i$ when the depths of all blocks belong to a set of $i + 1$ consecutive depths.) We show that if we start from a safe initial partition of size n and we use sufficiently large sampling-size functions, then, with high probability, in $\Theta(n)$ steps another safe partition is reached, and all intermediate partitions have $\varrho \leq 2$. In Chapter 5, we consider the complementary case where we start from a non-safe partition, and we provide an upper bound on the number of steps required to reach a safe partition, with high probability.

In Section 4.1, we describe the main result of the chapter. An outline of its proof is provided in Section 4.2. In Sections 4.3–4.5 we show some auxiliary results, which we use in Section 4.6 to derive the main result.

[1] Throughout Chapters 4 and 5, whenever we say "partition" we mean "sorted binary partition."

4.1 Statement of the main result: from a safe to a safe partition

In this chapter we consider $\mathcal{S}$-processes that have a "very balanced" initial partition S_0, and sampling-size functions

$$\lambda_+(k), \lambda_-(k) = \Theta(k)$$

that are "sufficiently large." We also assume that the processes' length N is "large enough." We elaborate on these requirements below. We do not impose any constraints on the strategy of the adversary, or the precision g. For any such $\mathcal{S}$-process of $|S_0| = n$, we show that, roughly speaking, with probability at least $1 - 1/n^{\Theta(1)}$, in $\Theta(n)$ steps we reach another "very balanced" partition, and all intermediate partitions have $\varrho \leq 2$ — i.e., in each of these partitions, the depths of all blocks belong to a set of 3 consecutive depths.

We quantify what we mean by a "very balanced" partition by introducing the class of *safe* partitions. A partition is *safe* if it satisfies one of the following two conditions:

- $\varrho < 2$
- $\varrho = 2$ and $\max\{\ell_\mu, \ell_\xi\} \leq 1/4 + \varepsilon$, where $\varepsilon = 1/16$.

(The threshold $1/4 + \varepsilon$ was chosen for convenience; any threshold within a certain range would work for our analysis.)

We now state the main result of this chapter formally. We use the following notation with respect to a given $\mathcal{S}$-process. For $t = 0, 1, \ldots, N$, we let

$$\mu_t = \mu(S_t) \qquad \xi_t = \xi(S_t) \qquad \varrho_t = \varrho(S_t)$$

Also, for $k \geq 0$, we let

$$\lambda(k) = \min\{\lambda_+(k),\ \lambda_-(k)\}$$

The big-oh notation below is with respect to $|S_0| \to \infty$.

Theorem 4.1. *For any long enough $\mathcal{S}$-process such that S_0 is safe, with probability*

$$1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}) \tag{4.1}$$

there is $\tau \in [c_1 2^{\mu_0} .. c_2 2^{\mu_0}]$, where c_1, c_2 are positive constants, such that

(i) S_τ is safe

(ii) for all $t < \tau$, $\varrho_t \leq 2$

(iii) for all $t \leq \tau$, $\xi_t \leq \xi_0 + 1$.

By "long enough" $\mathcal{S}$-process we mean that it has length $N \geq c_2 2^{\mu_0}$. (See the remark at the end of Section 3.5.2.)

Since S_0 is safe, $\mu_0 = \log|S_0| - \mathrm{O}(1)$. Hence, the endpoints of the interval to which τ belongs are both in $\Theta(|S_0|)$. Also, since $\lambda(k) \in \Theta(k)$, there is a constant $a > 0$ such that $\lambda(k) \geq ak$, for all large enough k. Therefore,

$$\mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}) \subseteq \mathrm{O}(|S_0|^{1-a\cdot(1/4-\varepsilon)\log e}) \subseteq \mathrm{O}(|S_0|^{1-a/4})$$

So, the big-oh term in (4.1) can be made smaller than any given polynomial of $1/|S_0|$, by using sufficiently large sampling-size functions.

Note that τ may not correspond to the *first* step when a safe partition is reached; i.e., it is possible that S_t is safe for some $0 < t < c_1 2^{\mu_0}$. By stipulating that $\tau \geq c_1 2^{\mu_0}$, instead of $\tau \geq 1$, we ensure that Theorem 4.1 argues about the balance of the partitions in the first $\Theta(|S_0|)$ steps of the $\mathcal{S}$-process, instead of the first $\mathrm{O}(|S_0|)$ steps. Note that we cannot provide similar guarantees for a number of steps τ that is $\omega(|S_0|)$, even if we drop (iii). Since the strategy of the adversary can be arbitrary, the system size may become very small after $|S_0|$ steps, in which case we cannot ensure that ϱ_t stays ≤ 2 with high probability (in $|S_0|$).

4.2 Outline of the proof

First, we provide an informal justification for a slightly simpler version of Theorem 4.1, where we require that $\tau \in [1..c2^{\mu_0}]$ instead of $\tau \in [c_1 2^{\mu_0}..c_2 2^{\mu_0}]$ — so, τ corresponds to the first step when a safe partition is reached. We also assume that $\varepsilon = 0$. In an $\mathcal{S}$-process, whenever an $\textsc{addBlk}_\mathbf{S}$ operation is applied to a partition of size n and $\mu = \Theta(\log n)$, the depth of the block split is smaller or equal to the depths of $\Omega(\log n)$ randomly probed blocks. Similarly, a $\textsc{rmBlk}_\mathbf{S}$ operation applied to such a partition merges a pair of blocks of depth greater or equal to the depths of $\Theta(\log n)$ randomly probed blocks. It is easy to see that, for that number of random probes, with high probability, the smallest random point chosen is $< 1/4$ and the largest is $> 3/4$. We will refer to this fact as Result 1. By Result 1, if $\ell_{\mu_0}(S_0) \geq 1/4$ and an $\textsc{addBlk}_\mathbf{S}$ operation occurs in the first step then, with high probability, a block of depth μ_0 is split. Likewise, if $\ell_{\xi_0}(S_0) \geq 1/4$ and a $\textsc{rmBlk}_\mathbf{S}$ operation occurs first, a pair of blocks of depth ξ_0 are merged. Thus, in both cases the resulting partition is safe. By Result 1, we also have that if $\max\{\ell_{\mu_0}(S_0), \ell_{\mu_0+2}(S_0)\} < 1/4$ (so, $\varrho_0 = 1$ or 2) and an $\textsc{addBlk}_\mathbf{S}$ operation occurs first then, with high probability, the depth of the block split is

μ_0 or $\mu_0 + 1$. Similarly, if $\max\{\ell_{\xi_0}(S_0), \ell_{\xi_0-2}(S_0)\} < 1/4$ a $\textsc{rmBlk}_{\mathbf{S}}$ operation merges blocks of depth ξ_0 or $\xi_0 - 1$. So, again, in both cases we result in a safe partition in a single step.

It remains to consider the two cases where either $\ell_{\mu_0}(S_0) < 1/4 = \ell_{\mu_0+2}(S_0)$ and an $\textsc{addBlk}_{\mathbf{S}}$ operation takes place in the first step, or $\ell_{\xi_0}(S_0) < 1/4 = \ell_{\xi_0-2}(S_0)$ and a $\textsc{rmBlk}_{\mathbf{S}}$ occurs. (Note that in both cases $\varrho_0 = 2$.) The two cases are symmetric, so we consider only the former. An adaptation of a well known bins-and-balls result [8] yields that, with high probability, the number of $\textsc{addBlk}_{\mathbf{S}}$ operations required until all blocks of depth μ_0 are split is $\mathrm{O}(n)$, provided that no blocks of depth $\leq \mu_0 + 1$ are merged during that time. In particular, we can show that, for a large enough λ_+, with high probability, all blocks of depth μ_0 are split before $\ell_{\mu_0}(S_t) + \ell_{\mu_0+1}(S_t)$ becomes smaller than $1/4$, given that no blocks of depth $\leq \mu_0 + 1$ are merged in the meantime; we will refer to this as Result 2. Note that, by Result 1, if $\ell_{\mu_0}(S_t) + \ell_{\mu_0+1}(S_t) \geq 1/4$ then, with high probability, an $\textsc{addBlk}_{\mathbf{S}}$ operation splits a block of depth $\leq \mu_0 + 1$. Combining this and Result 2, yields that, with high probability, in $\mathrm{O}(n)$ steps either (i) all blocks of depth μ_0 are split before any block of depth ξ_0 is split or blocks of depth $< \xi_0$ are merged; or (ii) a pair of blocks of depth $\leq \mu_0 + 1$ is merged before all blocks of depth μ_0 are split or a block of depth ξ_0 is split. If case (i) applies then the desired result holds. For (ii) note that, by Result 1, it must be $\ell_{\xi_0}(S_t) < 1/4$ for some previous step, so, a safe partition was reached then, and the desired result holds, as well.

The actual proof of Theorem 4.1 proceeds roughly as follows. First we reduce the set of different initial partitions we need to consider. Specifically, we show that for each n, there is a single safe partition of size n that is less balanced (with respect to the $\succeq$ relation) than all other safe partitions of the same size. We call this partition *borderline-safe*. (The details are described in Section 4.3.) We show that to prove the theorem it suffices to consider only initial partitions that are borderline-safe.

We then distinguish two different cases, depending on the (borderline-safe) initial partition S_0. The first case is when $\varrho_0 \in \{1, 2\}$ and $\min\{\ell_{\mu_0}(S_0), \ell_{\xi_0}(S_0)\} \geq 1/4$. In this setting, the probability of interest (i.e., that (i)–(iii) hold for some $\tau \in [c_1 2^{\mu_0} .. c_2 2^{\mu_0}]$) is bounded from below by the probability that all the blocks that are split in the first $\varepsilon 2^{\mu_0}$ steps have depth μ_0 and all the blocks merged have depth ξ_0. We compute this probability using the simple result we show in Section 4.4, which roughly corresponds to Result 1 we described above.

The complementary case is when $\varrho_0 = 2$ and exactly one of $\ell_{\mu_0}(S_0)$ or $\ell_{\xi_0}(S_0)$ is $\geq 1/4$ — recall that we only consider borderline-safe initial partition. Suppose $\ell_{\xi_0}(S_0) \geq 1/4$ (the other case is symmetric). The probability of interest is bounded from below by the probability that in $\Theta(|S_0|)$ steps: (a) either all blocks of depth μ_0 are split, or $\ell_{\xi_0}(S_t) \leq 1/4 + \varepsilon$; and (b) during these steps, no blocks of depth ξ_0 are split, and all blocks merged have depth ξ_0. We compute the probability of this joint event based on the result of Section 4.4 and a result we show in Section 4.5, which roughly corresponds to Result 2.

These results are integrated into the proof of Theorem 4.1 in Section 4.6.

4.3 Comparing the balance of safe partitions

Recall from Section 3.4 that relation $\succeq$ on $\mathbf{S}_n$ is a *partial* order. In this section, we prove that if we restrict the domain to the set of *safe* partitions of size n then $\succeq$ is a *total* order.

Let $\mathbf{F}_n$, for $n \in \mathbb{N}^*$, denote the set of all safe partitions of size n. We show that there is a single most balanced partition in $\mathbf{F}_n$, which we denote by $\hat{\Pi}_n$; i.e., $\hat{\Pi}_n \succeq S$, for all $S \in \mathbf{F}_n$. For this partition $\varrho \in \{0, 1\}$, while all other partitions in $\mathbf{F}_n$ have $\varrho = 2$. Also, all $S \in \mathbf{F}_n - \{\hat{\Pi}_n\}$ have the same μ, which we denote by $\hat{\mu}_n$. For each $S \in \mathbf{F}_n$, there is a *distinct* $d_S \in \mathbb{N}$ such that we can construct S from $\hat{\Pi}_n$ by applying to it d_S move-down operations; specifically, $\mathbb{V}_{\hat{\mu}_n \to \hat{\mu}_n+1}$ operations. The total ordering on $\mathbf{F}_n$ is based on this quantity: for every $S, S' \in \mathbf{F}_n$, if $d_S \leq d_{S'}$ then $S \succeq S'$.

So, first, we establish that there is a single *most balanced* partition in $\mathbf{F}_n$. For every $S \in \mathbf{S}_n$,

$$\sum_i s_i(S) = n \qquad \text{and} \qquad \sum_i \big(s_i(S)/2^i\big) = 1$$

It is easy to verify that there is a unique S such that $\varrho(S) \in \{0, 1\}$ and it satisfies the above pair of equations. (Specifically, $\varrho(S) = 0$ if n is a power of 2, and $\varrho(S) = 1$ otherwise.) Let $\hat{\Pi}_n$ denote this partition, and, for every $S \in \mathbf{S}_n$ of $\varrho \leq 2$, let

$$d_S = \begin{cases} 0, & \text{if } \varrho(S) \in \{0, 1\} \\ \min\{s_\mu(S),\ s_\xi(S)/2\}, & \text{if } \varrho(S) = 2 \end{cases}$$

It is straightforward to show that $\mathbb{V}^{d_S}_{\mu(S)+1\to\mu(S)}(S)$ is in $\mathbf{S}_n$ and it has $\varrho \in \{0, 1\}$. (We write $\mathbb{V}^i_{k\to k'}(S)$ to denote $\overbrace{\mathbb{V}_{k\to k'} \cdots \mathbb{V}_{k\to k'}}^{i}(S)$.) Therefore, for all $S \in \mathbf{S}_n$ of $\varrho \leq 2$ (and, thus, for all $S \in \mathbf{F}_n$),

$$\mathbb{V}^{d_S}_{\mu(S)+1\to\mu(S)}(S) = \hat{\Pi}_n \tag{4.2}$$

The above implies that $\hat{\Pi}_n \succeq S$, for all $S \in \mathbf{F}_n$; i.e., $\hat{\Pi}_n$ is the most balanced safe partition of size n.[2]

[2] In fact, more is true: $\hat{\Pi}_n$ is more balanced than any other partition in $\mathbf{S}_n$, safe or not.

The next lemma says that any partition that is more balanced than a safe partition is also safe.

Lemma 4.2. *If $S \succeq S'$ and S' is safe then S is safe.*

Proof. If $\varrho(S) < 2$ the lemma obviously holds. So, suppose that $\varrho(S) \geq 2$. By Lemma 3.4 then

$$\varrho(S) = \varrho(S') = 2 \qquad \mu(S) = \mu(S') = a \qquad \xi(S) = \xi(S') = a + 2 \tag{4.3}$$

for some $a \in \mathbb{N}$. So, by (4.2),

$$\mathbb{V}^{d_S}_{a+1\to a}(S) = \hat{\Pi}_{|S|} = \hat{\Pi}_{|S'|} = \mathbb{V}^{d_{S'}}_{a+1\to a}(S')$$

Since $S \succeq S'$, we have $d_S \leq d_{S'}$, so, by (3.1), the relation above yields

$$S = \mathbb{V}^{d_{S'}-d_S}_{a+1\to a}(S')$$

($d_S \leq d_{S'}$ because if it were $d_S > d_{S'}$ we would have $S' = \mathbb{V}^{d_S - d_{S'}}_{a+1\to a}(S) \succ S$.) Therefore, $\ell_a(S) \leq \ell_a(S')$ and $\ell_{a+2}(S) \leq \ell_{a+2}(S')$. Combining this with (4.3) and the assumption that S' is safe, yields S is also safe. ■

Since $\hat{\Pi}_n$ is the only partition in $\mathbf{S}_n$ that has $\varrho < 2$, we have that for all other $S \in \mathbf{F}_n$, $\varrho(S) = 2$. We now show that all these S have also the same μ (and, thus, the same $\xi = \mu + 2$). By (3.1), relation (4.2) implies that, for all $S \in \mathbf{F}_n - \{\hat{\Pi}_n\}$,

$$S = \mathbb{V}^{d_S}_{\mu(S)\to\mu(S)+1}(\hat{\Pi}_n) \tag{4.4}$$

So, $\mathbb{V}_{\mu(S)\to\mu(S)+1}(\hat{\Pi}_n) \succeq S$ (since $d_s > 0$), and, by Lemma 4.2,

$$\mathbb{V}_{\mu(S)\to\mu(S)+1}(\hat{\Pi}_n) \in \mathbf{F}_n \tag{4.5}$$

It is straightforward to show that for each n there is at most one k such that $\mathbb{V}_{k\to k+1}(\hat{\Pi}_n) \in \mathbf{F}_n$, and if such a k exists, $\mu(\hat{\Pi}_n) \leq k + 1 \leq \xi(\hat{\Pi}_n)$; we denote this k by $\hat{\mu}_n$. Combining this fact with (4.5), we obtain that, for all $S \in \mathbf{F}_n - \{\hat{\Pi}_n\}$,

$$\mu(S) = \hat{\mu}_n \tag{4.6}$$

and

$$\mu(\hat{\Pi}_n) \leq \mu(S) + 1 \leq \xi(\hat{\Pi}_n) \tag{4.7}$$

We can now show that relation $\succeq$ on $\mathbf{F}_n$ is a total order, as follows. Substituting $\mu(S)$ for $\hat{\mu}_n$ in (4.4) (because of (4.6)), we have that for all $S \in \mathbf{F}_n - \{\hat{\Pi}_n\}$,

$$S = \mathbb{V}^{d_S}_{\hat{\mu}_n \to \hat{\mu}_n+1}(\hat{\Pi}_n)$$

Let $T, T' \in \mathbf{F}_n - \{\hat{\Pi}_n\}$, and assume that $d_T \le d_{T'}$. Then, by the relation above,

$$T' = \mathbb{V}^{d_{T'}}_{\hat{\mu}_n \to \hat{\mu}_n+1}(\hat{\Pi}_n) = \mathbb{V}^{d_{T'}-d_T}_{\hat{\mu}_n \to \hat{\mu}_n+1}\,\mathbb{V}^{d_T}_{\hat{\mu}_n \to \hat{\mu}_n+1}(\hat{\Pi}_n) = \mathbb{V}^{d_{T'}-d_T}_{\hat{\mu}_n \to \hat{\mu}_n+1}(T) \preceq T$$

Combining this and the fact that $\hat{\Pi}_n \succeq S$, for all $S \in \mathbf{F}_n$, yields that

Lemma 4.3. *$\succeq$ on $\mathbf{F}_n$ is a total order.*

The above result implies that, for each n, there is a single partition in $\mathbf{F}_n$, denote $\check{\Pi}_n$, such that $\check{\Pi}_n \preceq S$, for all $S \in \mathbf{F}_n$; i.e., $\check{\Pi}_n$ is the *least balanced* safe partition of size n. We call $\check{\Pi}_n$ *borderline-safe*. We can compute $\check{\Pi}_n$ by observing that $d_{\check{\Pi}_n} > d_S$, for all $S \in \mathbf{F}_n - \{\check{\Pi}_n\}$. The next lemma states two simple properties of borderline-safe partitions that we will use in the proof of Theorem 4.1.

Lemma 4.4.

(a) For all $S \in \mathbf{F}_n$

$$\mu(\check{\Pi}_n) \le \mu(S) \le \mu(\check{\Pi}_n) + 1 \quad \textit{and} \quad \xi(\check{\Pi}_n) - 1 \le \xi(S) \le \xi(\check{\Pi}_n)$$

(b) For all $n \ge 5$, $\varrho(\check{\Pi}_n) \in \{1, 2\}$. Also, for all sufficiently large n,[3]
if $\varrho(\check{\Pi}_n) = 1$ then

$$\min\{\ell_\mu(\check{\Pi}_n),\, \ell_\xi(\check{\Pi}_n)\} \ge 1/4 + \varepsilon$$

and if $\varrho(\check{\Pi}_n) = 2$ then

$$\max\{\ell_\mu(\check{\Pi}_n),\, \ell_\xi(\check{\Pi}_n)\} = 1/4 + \varepsilon$$

Proof Sketch. Part (a) follows from (4.7) and Lemma 3.4. Part (b) can be shown by contradiction: If any of the relations did not hold then we could obtain a safe partition that is less balanced than $\check{\Pi}_n$ by applying to it operation $\mathbb{V}_{\hat{\mu}_n \to \hat{\mu}_n+1}$. ■

[3] Specifically, for all n such that $\hat{\mu}_n \ge -\log \varepsilon = 4$.

4.4 On the outcome of a single step

We now show a simple lemma we will use to argue about the outcome of a single step of an $\mathcal{S}$-process. Part (a) of the lemma describes a *sufficient* condition for an operation $\textsc{addBlk}_{\mathbf{S}}$ to split a block of depth smaller or equal to some given k. Part (b) is an analogous result for $\textsc{rmBlk}_{\mathbf{S}}$ operations.

Lemma 4.5. *For all $S \in \mathbf{S}_n$, $k \in \mathbb{N}$, and $Z = \langle z_1, \dots, z_m \rangle \in I^m$, for a large enough m,*

(a) if $\min\{z_i \,:\, i \leq \lambda_+(k+1)\} < \ell_{\leq k}(S)$ then $\textsc{addBlk}_{\mathbf{S}}(S, Z) \succeq \hat{\mathbb{S}}_k(S)$

(b) if $\max\{z_i \,:\, i \leq \lambda_-(\mu)\} \geq \ell_{<k}(S)$ then $\textsc{rmBlk}_{\mathbf{S}}(S, Z) \succeq \hat{\mathbb{M}}_k(S)$

Proof. Suppose that $\min\{z_i \,:\, i \leq \lambda_+(k+1)\} < \ell_{\leq k}$. Then,

$$\min\{\theta(S, z_i) \,:\, i \leq \lambda_+(k+1)\} \leq k$$

or, equivalently,

$$\min\{\theta(S, z_i) \,:\, i \leq \lambda_+(k')\} < k', \quad \text{for all } k' > k$$

Hence, by the definition of $\textsc{getSmlDpth}$,

$$\textsc{getSmlDpth}(S, Z) \leq k$$

By the definition of $\textsc{addBlk}_{\mathbf{S}}$, Lemma 3.5(a), and the inequality above, we have

$$\textsc{addBlk}_{\mathbf{S}}(S, Z) = \hat{\mathbb{S}}_{\textsc{getSmlDpth}(S,Z)}(S) \succeq \hat{\mathbb{S}}_k(S)$$

So, part (a) holds. The proof for part (b) is similar and is omitted. ■

From the lemma above it is immediate that if Z is a large enough random point-vector then

$$\mathbb{Pr}[\textsc{addBlk}_{\mathbf{S}}(S, Z) \succeq \hat{\mathbb{S}}_k(S)] \geq 1 - (1 - \ell_{\leq k})^{\lambda_+(k+1)}$$

and

$$\mathbb{Pr}[\textsc{rmBlk}_{\mathbf{S}}(S, Z) \succeq \hat{\mathbb{M}}_k(S)] \geq 1 - (\ell_{<k})^{\lambda_-(\mu)}$$

(For the definition of a random point-vector and what we mean by a “large enough” random point-vector see the end of Section 3.5.2, and the remark therein.)

4.5 On the outcome of a series of steps: A-times and R-times

In this section we prove a result that we will use to establish an upper bound on the number of $\textsc{addBlk}_{\mathbf{S}}$ operations that take place in an $\mathcal{S}$-process until all largest blocks of the initial partition are split (i.e., until $\mu_t = \mu_0 + 1$), when no blocks of depth μ_0 or $\mu_0 + 1$ are merged during that time. We also provide an analogous result that we use to bound the number of $\textsc{rmBlk}_{\mathbf{S}}$ operations until all smallest blocks are merged (i.e., until $\xi_t = \xi_0 - 1$), when no blocks of depth ξ_0 or $\xi_0 - 1$ are split. The first result holds for arbitrary initial partitions, while the second assumes that we start from an initial partition that has $\varrho \leq 2$.

Bounding the number of $\textsc{addBlk}_{\mathbf{S}}$ operations until $\mu_t = \mu_0 + 1$ is complicated by the fact that the order in which operations take place is not fixed; it is determined by an (online adaptive) adversary — described by the strategy of the adversary. What is more, we need to consider all possible such adversaries. Roughly speaking, we tackle this issue as follows. We consider the *stronger* adversarial model where the adversary has two additional abilities: (1) she knows in advance the sequence of random points that will be used in each $\textsc{addBlk}_{\mathbf{S}}$ operation — i.e., she knows the Y_t for the step when the first, second, third, etc., $\textsc{addBlk}_{\mathbf{S}}$ operation will take place (it is up to her to decide when these steps will occur in the overall sequence of $\textsc{addBlk}_{\mathbf{S}}$ and $\textsc{rmBlk}_{\mathbf{S}}$ operations); and (2) she can choose the sequence of points that will be used in each $\textsc{rmBlk}_{\mathbf{S}}$ operation — so, essentially, she decides the depth of the blocks that are merged in each $\textsc{rmBlk}_{\mathbf{S}}$ operation. The resulting process can be equivalently described as follows. We initially choose a sequence of independent uniformly-random vectors in I_g^{Λ}. The i-th vector in this sequence will be the Y_t that will be used in the i-th $\textsc{addBlk}_{\mathbf{S}}$ operation, if there is such an operation. The adversary then decides (off-line) the order in which operations occur and for each $\textsc{rmBlk}_{\mathbf{S}}$ operation she also decides the corresponding Y_t vector. For this model, we establish an upper bound on the number of $\textsc{addBlk}_{\mathbf{S}}$ operations required until $\mu_t = \mu_0 + 1$, for all possible choices of the adversary such that no blocks of depths μ_0 or $\mu_0 + 1$ are merged. The bound is probabilistic; the uncertainty results because of the random choice of the sample points used in each $\textsc{addBlk}_{\mathbf{S}}$ operation — not because of the randomness of the adversary. A similar approach is used to obtain an upper bound on the number of $\textsc{rmBlk}_{\mathbf{S}}$ operations required until $\xi_t = \xi_0 - 1$.

We begin by introducing some terminology. Let W be a fixed sequence of vectors, where all vectors have the same size and their elements are points in I; we denote the length of this sequence by l_w, and the size of each vector by h_w. I.e.,

$$W = \langle W_1, \ldots, W_{l_w} \rangle, \qquad \text{where } W_t = \langle W_{t,1}, \ldots, W_{t,h_w} \rangle \in I^{h_w}, \text{ for each } t = 1, \ldots, l_w \tag{4.8}$$

We assume that h_w is large enough that the operations we describe below are well defined. Let also $S \in \mathbf{S}$. Consider now a (fixed) sequence of partitions $\langle T_0, \ldots, T_{l_w} \rangle$ be such that: $T_0 = S$, and, for $t \geq 1$, T_t results from T_{t-1} by first merging zero or more pairs of sibling blocks of depth $\geq \mu(S) + 2$, and then applying a single ADDBLK$_\mathbf{S}$ operation using W_t as the sequence of sample points. More formally,

$$T_t = \begin{cases} S, & \text{if } t = 0 \\ \text{ADDBLK}_\mathbf{S}(T'_{t-1}, W_t), & \text{otherwise} \end{cases} \tag{4.9}$$

where

$$T'_t = \mathbb{M}_{k_1^t} \cdots \mathbb{M}_{k_{m_t}^t}(T_t), \quad \text{for some } m_t \geq 0, \text{ and } k_1^t, \ldots, k_{m_t}^t \geq \mu(S) + 2 \tag{4.10}$$

The conditions $k_j^t \geq \mu(S) + 2$ ensure that no blocks of depth $\mu(S)$ are merged or (new) blocks of depth $\mu(S)$ are created, as a result of the $\mathbb{M}$ operations applied to T_t; hence, for all t,

$$\mu(T_t) \geq \mu(S) \quad \text{and} \quad s_{\mu(S)}(T_{t+1}) \leq s_{\mu(S)}(T'_t) = s_{\mu(S)}(T_t) \tag{4.11}$$

We denote by $\mathcal{A}_{S,W}$ the set of all such sequences $\langle T_0, \ldots, T_{l_w} \rangle$. Given a $\langle T_0, \ldots, T_{l_w} \rangle \in \mathcal{A}_{S,W}$, let a be the time when the last block of depth $\mu(S)$ is split; i.e.,

$$a = \inf\{t \,:\, \mu(T_t) = \mu(S) + 1\}$$

(Note that $a = \infty$ if $\mu(T_t) = \mu(S)$, for all t.) The supremum of a over all sequences in $\mathcal{A}_{S,W}$ is called the *A-time of* $\langle S, W \rangle$.

Suppose now that instead of the fixed W we have a random point-array $\mathcal{Z}$. (For the definition of a random point-array see the end of Section 3.5.2, and the remark therein.) The A-time of $\langle S, \mathcal{Z} \rangle$ is then a random variable. The next lemma establishes a probabilistic upper bound on the A-time of $\langle S, \mathcal{Z} \rangle$; it bounds the probability that the A-time is at most equal to the number of largest blocks in S plus a term linear in $2^{\mu(S)}$ – the maximum possible number of such blocks.

Lemma 4.6. *Let τ be the A-time of $\langle S, \mathcal{Z} \rangle$, where $S \in \mathbf{S}$ and $\mathcal{Z} = \langle Z_1, Z_2, \ldots \rangle$ is a large enough random point-array, and let γ be a positive constant. Then,*

$$\mathbb{Pr}[\tau \leq s_\kappa(S) + \gamma 2^\kappa] = 1 - \mathrm{O}\big(2^\kappa e^{-\gamma(1 - \mathrm{O}(1/\ln \kappa))\lambda_+(\kappa+1)}\big)$$

where $\kappa = \mu(S)$.

As we explained in Section 3.5.2, the length of the $\mathcal{Z}$ is $|\mathcal{Z}| \geq s_\kappa(S) + \gamma 2^\kappa$, its height $|Z_1|$ is large enough that the A-time of $\langle S, \mathcal{Z}\rangle$ is well defined — i.e., the $\textsc{addBlk}_\mathbf{S}$ operations involved are well defined, and its precision is greater or equal to the maximum ξ of the partitions these $\textsc{addBlk}_\mathbf{S}$ operations are applied to — e.g., $\xi(S) + |\mathcal{Z}|$.

We will use the above result later in our analysis of an $\mathcal{S}$-process to bound from above the number of steps required until either all largest blocks of S_0 have been split or a pair of blocks of depth $\leq \mu_0 + 1$ is merged. We do so by observing that the number of $\textsc{addBlk}_\mathbf{S}$ operation until some of the above two events occurs is at most equal to the A-time of $\langle S_0, \mathcal{Z}\rangle$, where $\mathcal{Z}$ is, roughly speaking, the sequence of the Y_i that correspond to the steps where the $\textsc{addBlk}_\mathbf{S}$ operations occur.

Proof of Lemma 4.6. We describe a sequence of partitions that is a function of $\mathcal{Z}$, such that τ is bounded from above by a similar quantity τ^* defined for that sequence. We then prove the desired bound for τ by showing that this bound applies to τ^*.

Let W be defined as in (4.8), with l_w and h_w equal to the length $|\mathcal{Z}|$ of $\mathcal{Z}$ and its height $|Z_1|$, respectively. Consider the sequence of partitions $\langle T_0^*(W), \ldots, T_{l_w}^*(W)\rangle$, where

$$T_t^*(W) = \begin{cases} S, & \text{if } t = 0 \\ \mathbb{S}_\kappa(T_{t-1}^*(W)), & \text{if } t \neq 0 \text{ and } \min\{W_{t,i} \,:\, i \leq \lambda_+(\kappa+1)\} < \ell_\kappa(T_{t-1}^*(W)) \\ T_{t-1}^*(W), & \text{otherwise} \end{cases} \tag{4.12}$$

(Recall that $\kappa = \mu(S)$.) The above sequence resembles the sequence in $\mathcal{A}_{S,W}$ where no blocks are ever merged — i.e., the sequence described by (4.9) and (4.10) when $m_t = 0$ for all t. The only difference is that in the sequence of $T_t^*(W)$ splits occur only in steps t such that some of the first $\lambda_+(\kappa+1)$ elements of W_t belong to a block of depth κ of $T_{t-1}^*(W)$. Note that when this condition holds then, by Lemma 4.5, $T_t^*(W) = \textsc{addBlk}_\mathbf{S}(T_{t-1}^*(W), W_t)$.

Let

$$\tau^*(W) = \inf\{t \,:\, \mu(T_t^*(W)) = \kappa + 1\}$$

and $\tau(W)$ be the A-time of $\langle S, W\rangle$. Then,

Claim 4.7. *For all W, $\tau^*(W) \geq \tau(W)$.*

Proof. By contradiction. Suppose that $\tau^*(W) < \tau(W)$, for *some* W. Then, for this W, there is a sequence $\langle T_0, \ldots, T_{l_w}\rangle \in \mathcal{A}_{S,W}$ (defined as in (4.9)–(4.10)) such that

$$\tau^*(W) < a, \quad \text{where } a = \inf\{t \,:\, \mu(T_t) = \kappa + 1\}$$

In the rest of the proof we write T_t^* and τ^* to denote $T_t^*(W)$ and $\tau^*(W)$, respectively, for the above value of W. Let

$$t_0 = \inf\{t \,:\, \ell_\kappa(T_t^*) < \ell_\kappa(T_t)\}$$

Note that $t_0 < \infty$, since, by the assumption that $\tau^* < a$, $\tau^* < \infty$ and $\ell_\kappa(T_{\tau^*}^*) = 0 < \ell_\kappa(T_{\tau^*})$. By the definition of T_t we have that, for $t \geq 1$, $s_\kappa(T_{t-1}) - s_\kappa(T_t) \in \{0, 1\}$. Similarly, $s_\kappa(T_{t-1}^*) - s_\kappa(T_t^*) \in \{0, 1\}$. Combing the last two observations with the definition of t_0, yields the next three relations:

$$s_\kappa(T_{t_0-1}^*) = s_\kappa(T_{t_0-1}) \tag{4.13}$$

$$s_\kappa(T_{t_0}^*) = s_\kappa(T_{t_0-1}^*) - 1 \tag{4.14}$$

$$s_\kappa(T_{t_0}) = s_\kappa(T_{t_0-1}) \tag{4.15}$$

By (4.14) and the definition of $T_t^*(W)$,

$$\min\{W_{t_0,i} \,:\, i \leq \lambda_+(\kappa+1)\} < \ell_\kappa(T_{t_0-1}^*)$$

So, by (4.13),

$$\min\{W_{t_0,i} \,:\, i \leq \lambda_+(\kappa+1)\} < \ell_\kappa(T_{t_0-1}) = \ell_\kappa(T'_{t_0-1})$$

(T'_t was defined in (4.9).) Since $\mu(T'_{t_0-1}) \geq \kappa$, the above relation yields

$$\min\{\theta(T'_{t_0-1}, W_{t_0,i}) \,:\, i \leq \lambda_+(\kappa+1)\} = \kappa$$

which implies that

$$\textsc{getSmlDpth}(T'_{t_0-1}, W_{t_0}) = \kappa$$

Therefore,

$$T_{t_0} = \textsc{addBlk}_{\mathbf{S}}(T'_{t_0-1}, W_{t_0}) = \hat{\mathbb{S}}_{\textsc{getSmlDpth}(T'_{t_0-1}, W_{t_0})}\, T'_{t_0-1} = \mathbb{S}_\kappa\, T'_{t_0-1}$$

and, so, $s_\kappa(T_{t_0}) < s_\kappa(T_{t_0-1})$, which contradicts (4.15). ■ {of Claim 4.7}

By Claim 4.7, we have

$$\tau^*(\mathcal{Z}) \geq \tau(\mathcal{Z}) \equiv \tau$$

and, thus,

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq s_\kappa(S) + \gamma 2^\kappa] \leq \mathbb{Pr}[\tau \leq s_\kappa(S) + \gamma 2^\kappa] \tag{4.16}$$

We show the following lower bound for $\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq s_\kappa(S) + \gamma 2^\kappa]$. (The proof is described at the end of the section.)

Claim 4.8. $\mathbb{P}\mathrm{r}[\tau^*(\mathcal{Z}) \leq s_\kappa(S) + \gamma 2^\kappa] = 1 - \mathrm{O}\big(2^\kappa e^{-\gamma(1-\mathrm{O}(1/\ln\kappa))\lambda_+(\kappa+1)}\big)$

Combining this result with (4.16), yields the desired bound for $\mathbb{P}\mathrm{r}[\tau \leq s_\kappa(S) + \gamma 2^\kappa]$. ■

We now describe the analogous definitions and result for the case where we are interested in the time until all smallest blocks are merged (rather than the time until all largest blocks are split). Let $S \in \mathbf{S}$ be such that $\varrho(S) \leq 2$, and W be as before. We denote by $\mathcal{R}_{S,W}$ the class of all sequences of partitions $\langle T_0, \ldots, T_{l_w}\rangle$ such that

$$T_t = \begin{cases} S, & \text{if } t = 0 \\ \textsc{rmBlk}_\mathbf{S}(T'_{t-1}, W_t), & \text{if } t \neq 0 \text{ and } \textsc{rmBlk}_\mathbf{S}(T'_{t-1}, W_t) \succeq \hat{\mathbb{M}}_{\xi(S)-1}\, T'_{t-1} \\ T'_{t-1}, & \text{otherwise} \end{cases}$$

where

$$T'_t = \overbrace{\mathbb{S}_{\xi(S)-2} \cdots \mathbb{S}_{\xi(S)-2}}^{0 \text{ or more}}(T_t)$$

Note that the definition of a sequence in $\mathcal{R}_{S,W}$ is not completely symmetric to that of a sequence in $\mathcal{A}_{S,W}$. The former requires that $\varrho(S) \leq 2$, and it only allows $\textsc{rmBlk}_\mathbf{S}$ operations that merge blocks of depths $\xi(S)$ or $\xi(S) - 1$. These restrictions ensure that for all t, $\mu(T_t) \geq \xi(S) - 2$. The fact that only blocks of depth (at most) $\xi(S) - 2$ may be split, ensures that

$$\xi(T_t) \leq \xi(S) \quad \text{and} \quad s_{\xi(S)}(T_{t+1}) \leq s_{\xi(S)}(T'_t) = s_{\xi(S)}(T_t)$$

which is the analogue of (4.11). Given a $\langle T_0, \ldots, T_{l_w}\rangle \in \mathcal{R}_{S,W}$, let r be the time when the last pair of blocks of depth $\xi(S)$ is merged; i.e.,

$$r = \inf\{t \,:\, \xi(T_t) = \xi(S) - 1\}$$

The supremum of r taken over all sequences in $\mathcal{R}_{S,W}$ is called the *R-time of* $\langle S, W\rangle$.

The next result is the analogue of Lemma 4.6; the proof is very similar, so, we only sketch it.

Lemma 4.9. *Let τ be the R-time of $\langle S, \mathcal{Z}\rangle$, where $S \in \mathbf{S}$ is such that $\varrho(S) \leq 2$, and $\mathcal{Z} = \langle Z_1, Z_2, \ldots\rangle$ is a large enough random point-array, and let γ be a positive constant. Then,*

$$\mathbb{P}\mathrm{r}[\tau \leq (1/2)s_\kappa(S) + \gamma 2^{\kappa-1}] = 1 - \mathrm{O}\big(2^\kappa e^{-\gamma(1-\mathrm{O}(1/\ln\kappa))\lambda_-(\kappa)}\big)$$

where $\kappa = \xi(S)$.

Proof Sketch. As in the proof of Lemma 4.6, we describe a sequence of partitions such that τ is bounded from above by a similar quantity τ^* defined for that sequence, and then we prove the probabilistic bound for τ^*, instead. Let W be as in (4.8), with the same dimensions as $\mathcal{Z}$. The following is the analogue of definition (4.12):

$$T^*_t(W) = \begin{cases} S, & \text{if } t = 0 \\ \mathbb{M}_\kappa(T^*_{t-1}(W)), & \text{if } t \neq 0 \text{ and } \max\{W_{t,i} \,:\, i \leq \lambda_-(\kappa-2)\} \geq 1 - \ell_\kappa(T^*_{t-1}(W)) \\ T^*_{t-1}(W), & \text{otherwise} \end{cases}$$

(Here $\kappa = \xi(S)$.) Also let

$$\tau^*(W) = \inf\{t \,:\, \xi(T^*_t(W)) = \kappa - 1\}$$

and $\tau(W)$ be the R-time of $\langle S, W\rangle$. We can then show, as in Claim 4.7, that for all W, $\tau^*(W) \geq \tau(W)$. We can also show that

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq (1/2)s_\kappa(S) + \gamma 2^{\kappa-1}] = 1 - \mathrm{O}\big(2^\kappa e^{-\gamma(1-\mathrm{O}(1/\ln\kappa))\lambda_-(\kappa)}\big)$$

— the proof is symmetric to that of Claim 4.8. Combining these two results yields the desired bound for τ. ■

We will use the above result in Section 4.6 to show an upper bound for the number of steps required in an $\mathcal{S}$-process starting from a safe S_0 until: (1) all smallest blocks of S_0 have been merged, or (2) a block of depth $\geq \xi_0 - 1$ is split, or (3) a pair of blocks of depth $\xi_0 - 2$ is merged. We do so by observing that the number of rmBlks operation until one of the above three events occurs is at most equal to the R-time of $\langle S_0, \mathcal{Z}\rangle$, where $\mathcal{Z}$ is, roughly speaking, the sequence of the Y_i that correspond to the steps where the rmBlks operations occur.

Proof of Claim 4.8

We begin with a Chernoff-type bound for geometric random variables.

Lemma 4.10. *Let $Q_1, \ldots, Q_n$ be independent geometric random variables, such that, for all $1 \leq i \leq n$, $\mathbb{E}[Q_i] \leq 1/p$, where $0 < p < 1$. Then, for $Q = \sum_{i=1}^n (Q_i - 1)$, and any $\delta > 0$,*

$$\mathbb{Pr}\left[Q > (1+\delta)\frac{nq}{p}\right] < \left(\frac{(1+q\delta)^{1+q\delta}}{(1+\delta)^{(1+\delta)q}}\right)^{n/p} \tag{4.17}$$

where $q = 1 - p$. Also, for $0 < \delta \leq p$,

$$\mathbb{Pr}\left[Q > (1+\delta)\frac{nq}{p}\right] < \exp\left(-\frac{(p-\delta)\delta^2 nq}{2p}\right) \tag{4.18}$$

Proof. If X and Y are geometric random variables and $\mathbb{E}[X] \geq \mathbb{E}[Y]$ then X is stochastically larger than Y. So, to prove the lemma it suffices to consider only the case where

$$\mathbb{E}[Q_i] = \frac{1}{p}, \quad \text{for all } 1 \leq i \leq n$$

Then, for $0 < t < -\ln q$,

$$\mathbb{E}[\exp(t(Q_i - 1)] = \sum_{k=0}^{\infty} pq^k e^{tk} = \frac{p}{1 - qe^t} \tag{4.19}$$

Since

$$\exp(tQ) = \exp\Big(t \sum_{i=1}^{n} (Q_i - 1)\Big) = \prod_{i=1}^{n} \exp(t(Q_i - 1))$$

by the independence of the Q_i and (4.19) we have

$$\mathbb{E}[\exp(tQ)] = \prod_{i=1}^{n} \mathbb{E}[\exp(t(Q_i - 1))] = \Big(\frac{p}{1 - qe^t}\Big)^n \tag{4.20}$$

We can now bound $\mathbb{Pr}[Q > (1+\delta)nq/p]$ from above as follows. We have

$$\mathbb{Pr}\Big[Q > (1+\delta)\frac{nq}{p}\Big] = \mathbb{Pr}\Big[\exp(tQ) > \exp\Big(t(1+\delta)\frac{nq}{p}\Big)\Big]$$

and by applying Markov's inequality to the right-hand side, we obtain

$$\mathbb{Pr}\Big[Q > (1+\delta)\frac{nq}{p}\Big] < \frac{\mathbb{E}[\exp(tQ)]}{\exp(t(1+\delta)nq/p)}$$

So, by (4.20),

$$\mathbb{Pr}\Big[Q > (1+\delta)\frac{nq}{p}\Big] < \Big(\frac{p}{\exp(t(1+\delta)q/p) \cdot (1 - qe^t)}\Big)^n \tag{4.21}$$

Next, we compute the value of $t \in (0, -\ln q)$ that minimizes the expression on the right-hand side of (4.21). This is equivalent to computing the t that maximizes $\exp(t(1+\delta)q/p) \cdot (1 - qe^t)$. For that, we differentiate the last expression with respect to t and set the result to zero; solving for e^t yields

$$e^t = \frac{1+\delta}{1+q\delta}$$

(Note that $t = \ln \frac{1+\delta}{1+q\delta} < -\ln q$.) Substituting this value for e^t in (4.21), yields (4.17).

The bound (4.18) can be derived from (4.17) as follows. We can rewrite the latter as

$$\mathbb{Pr}\Big[Q > (1+\delta)\frac{nq}{p}\Big] < \exp\Big(\frac{n}{p}(1+q\delta)\ln(1+q\delta) - \frac{n}{p}(1+\delta)q\ln(1+\delta)\Big) \tag{4.22}$$

Using the fact that, for all $x > 0$, $x - \frac{1}{2}x^2 \le \ln(1+x) \le x - \frac{1}{2}x^2 + \frac{1}{3}x^3$, we obtain

$$\ln(1+q\delta) \le q\delta - \frac{1}{2}(q\delta)^2 + \frac{1}{3}(q\delta)^3 \quad \text{and} \quad \ln(1+\delta) \ge \delta - \frac{1}{2}\delta^2$$

Applying the above to the right-hand side of (4.22) yields

$$\mathbb{Pr}\left[Q > (1+\delta)\frac{nq}{p}\right] < \exp\left(-\frac{\delta^2 nq}{2p}\left(p - \delta + \frac{1}{3}q^2\delta(1-2q\delta)\right)\right)$$

This, together with the fact that, for $\delta \le p$,

$$1 - 2q\delta \ge 1 - 2pq = 1 - 2p(1-p) = p^2 + (1-p)^2 > 0$$

yields (4.18). ■

We now proceed to prove Claim 4.8. We will write τ^* and T_t^* to denote $\tau^*(\mathcal{Z})$ and $T_t^*(\mathcal{Z})$, respectively. Recall that $\mathcal{Z} = \langle Z_1, Z_2, \ldots\rangle$ is a random point-array,

$$T_t^* = \begin{cases} S, & \text{if } t = 0 \\ \mathbb{S}_\kappa(T_{t-1}^*), & \text{if } t \ne 0 \text{ and } Z_t^{\min} < \ell_\kappa(T_{t-1}^*) \\ T_{t-1}^*, & \text{otherwise,} \end{cases}$$

where $\kappa = \mu(S)$ and $Z_t^{\min}$ is the minimum of the first $\lambda_+(\kappa+1)$ elements of Z_t, and

$$\tau^* = \inf\{t \,:\, \mu(T_t^*) = \kappa + 1\}$$

We will also denote by n_0 the maximum number of blocks of depth k in any partition, i.e.,

$$n_0 = 2^\kappa$$

For $1 \le i \le s_\kappa(S)$, let χ_i be the number of partitions in the sequence of T_t^* that have $s_\kappa = i$; i.e.,

$$\chi_i = |\{t \,:\, s_\kappa(T_t^*) = i\}|$$

Note that

$$\tau^* = \begin{cases} \sum_{i=1}^{s_\kappa(S)} \chi_i, & \text{if } \sum_{i=1}^{s_\kappa(S)} \chi_i \le |\mathcal{Z}| \\ \infty, & \text{otherwise} \end{cases}$$

Therefore, since $|\mathcal{Z}| \ge s_\kappa(S) + \gamma n_0$,

$$\mathbb{Pr}[\tau^* \le s_\kappa(S) + \gamma n_0] = \mathbb{Pr}\left[\sum_{i=1}^{s_\kappa(S)} \chi_i \le s_\kappa(S) + \gamma n_0\right] \tag{4.23}$$

Note also that conditioned on the values of $\chi_{s_\kappa(S)}, \ldots, \chi_{i+1}$, χ_i is stochastically *smaller* than a geometric random variable with expectation $1/p_i$, where

$$p_i = \mathbb{Pr}[Z_t^{\min} < i/n_0] = 1 - (1 - i/n_0)^{\lambda_+(\kappa+1)} \tag{4.24}$$

(χ_i is not *equal* in distribution to this random variable, because χ_i is bounded by $|\mathcal{Z}|$.) Hence, if $\tau_1, \ldots, \tau_{n_0}$ are independent geometric random variables, such that, for each i, $\mathbb{E}[\tau_i] = 1/p_i$, then $\sum_{i=1}^{s_\kappa(S)} \chi_i$ is stochastically smaller than $\sum_{i=1}^{s_\kappa(S)} \tau_i$. From this, the fact that $\sum_{i=1}^{s_\kappa(S)} (\tau_i - 1) \leq \sum_{i=1}^{n_0} (\tau_i - 1)$, and (4.23), we obtain

$$\mathbb{Pr}[\tau^* \leq s_\kappa(S) + \gamma n_0] \geq \mathbb{Pr}\left[\sum_{i=1}^{s_\kappa(S)} \tau_i \leq s_\kappa(S) + \gamma n_0\right] \geq \mathbb{Pr}\left[\sum_{i=1}^{n_0} (\tau_i - 1) \leq \gamma n_0\right] \tag{4.25}$$

In the remainder of the proof we compute a lower bound for $\mathbb{Pr}[\sum_{i=1}^{n_0} (\tau_i - 1) \leq \gamma n_0]$. For that, we break $\sum_{i=1}^{n_0} (\tau_i - 1)$ into four smaller sums, compute a probabilistic upper bound for each of them, and then combine the results.

Let

$$n_1 = \left\lfloor \frac{n_0 \ln \kappa}{\lambda_+(\kappa+1)} \right\rfloor, \quad n_2 = \left\lfloor \frac{n_0}{\lambda_+(\kappa+1)} \right\rfloor, \quad n_3 = \left\lfloor \frac{n_0}{\lambda_+(\kappa+1) \ln \kappa} \right\rfloor, \quad n_4 = 0$$

For each $j \in [1..4]$, let

$$G_j = \sum_{n_j < i \leq n_{j-1}} (\tau_i - 1)$$

In the next series of claims we derive probabilistic upper bounds for the G_j.

Claim 4.11.

$$\mathbb{Pr}\left[G_1 > \frac{\kappa+1}{\kappa(\kappa-1)} n_0\right] \leq \exp\left(-(1 - \mathrm{o}(1)) \frac{n_0}{2\kappa^3}\right)$$

Proof. By (4.24), for $n_1 < i \leq n_0$,

$$p_i \geq 1 - (1 - (n_1+1)/n_0)^{\lambda_+(\kappa+1)} \geq 1 - \left(1 - \frac{\ln \kappa}{\lambda_+(\kappa+1)}\right)^{\lambda_+(\kappa+1)} \geq 1 - e^{-\ln \kappa} = 1 - 1/\kappa$$

So, by Lemma 4.10 (applied for $p = 1 - 1/\kappa$ and $\delta = 1/\kappa < p$),

$$\mathbb{Pr}\left[G_1 > \frac{\kappa+1}{\kappa(\kappa-1)} (n_0 - n_1)\right] < \exp\left(-\frac{(1 - 2/\kappa)(n_0 - n_1)/\kappa^3}{2(1 - 1/\kappa)}\right) = \exp\left(-(1 - \mathrm{o}(1)) \frac{n_0}{2\kappa^3}\right)$$

The desired result is immediate from the above relation. ■

The proof of the next result is very similar to that of Claim 4.11 and is omitted.

Claim 4.12.

$$\mathbb{Pr}\left[G_2 > \frac{e+1}{e(e-1)}n_1\right] \le \exp\left(-(1-\mathrm{o}(1))\frac{n_1(e-2)}{2e^3(e-1)}\right)$$

Claim 4.13.

$$\mathbb{Pr}[G_3 > 2n_2 \ln\kappa] \le \exp\left(-(1-\mathrm{o}(1))\frac{n_2}{64\ln^2\kappa}\right)$$

Proof. Is also similar to that of Claim 4.11. By (4.24), for $n_3 < i \le n_2$,

$$\begin{aligned} p_i &\ge 1 - (1-(n_3+1)/n_0)^{\lambda_+(\kappa+1)} \ge 1 - \left(1 - \frac{1}{\lambda_+(\kappa+1)\ln\kappa}\right)^{\lambda_+(\kappa+1)} \\ &\ge \frac{1}{\ln\kappa} - \frac{1}{2\ln^2\kappa} \ge \frac{1}{2\ln\kappa} \end{aligned}$$

where the first relation in the second line was obtained using the fact that $(1-\epsilon)^k \le 1 - k\epsilon + (k\epsilon)^2/2$, when $k\epsilon \le 1$. By Lemma 4.10 then (applied for $p = 1/(2\ln\kappa)$ and $\delta = p/2$), we obtain

$$\begin{aligned} \mathbb{Pr}\left[G_3 > \frac{\big(1+\frac{1}{4\ln\kappa}\big)\big(1-\frac{1}{2\ln\kappa}\big)(n_2-n_3)}{\frac{1}{2\ln\kappa}}\right] &< \exp\left(-\frac{\frac{1}{(4\ln\kappa)^3}(n_2-n_3)\big(1-\frac{1}{2\ln\kappa}\big)}{\frac{2}{2\ln\kappa}}\right) \\ &= \exp\left(-(1-\mathrm{o}(1))\frac{n_2}{64\ln^2\kappa}\right) \end{aligned}$$

This, together with the fact that

$$\frac{\big(1+\frac{1}{4\ln\kappa}\big)\big(1-\frac{1}{2\ln\kappa}\big)}{\frac{1}{2\ln\kappa}} \le 2\ln\kappa$$

yields the desired result. ■

Claim 4.14. *For any $\delta > 0$,*

$$\mathbb{Pr}[G_4 > \delta n_0] \le n_3 \cdot \exp\left(-\Big(1-\frac{1}{2\ln\kappa}\Big)\delta\lambda_+(\kappa+1)\right)$$

Proof. The technique we used to prove Claims 4.11–4.13 does not result in a tight enough bound in this case; so, we use a different approach. By (4.24), for $i \le n_3$,

$$\begin{aligned} p_i &= \mathbb{Pr}[Z_t^{\min} < n_3/n_0] \cdot \mathbb{Pr}[Z_t^{\min} < i/n_0 \mid Z_t^{\min} < n_3/n_0] \\ &= p_{n_3}\,\mathbb{Pr}[Z_t^{\min} < i/n_0 \mid Z_t^{\min} < n_3/n_0] \end{aligned}$$

Let K_t be the number elements that are $< n_3/n_0$, among the first $\lambda_+(\kappa+1)$ elements of Z_t. Then,

$$p_i = p_{n_3} \sum_{j=1}^{\lambda_+(\kappa+1)} \big(\mathbb{Pr}[Z_t^{\min} < i/n_0 \mid K_t = j] \cdot \mathbb{Pr}[K_t = j \mid Z_t^{\min} < n_3/n_0]\big)$$

Note that, for $j \in [1..\lambda_+(\kappa+1)]$,

$$\mathbb{Pr}[Z_t^{\min} < i/n_0 \mid K_t = j] = 1 - (1 - i/n_3)^j \geq i/n_3$$

so,

$$p_i \geq p_{n_3} \sum_{j=1}^{\lambda_+(\kappa+1)} \big((i/n_3)\,\mathbb{Pr}[K_t = j \mid Z_t^{\min} < n_3/n_0]\big) = p_{n_3} i/n_3$$

Therefore, G_4 is stochastically smaller than the sum $\sum_{i=1}^{n_3}(\tau'_i - 1) = \sum_{i=1}^{n_3} \tau'_i - n_3$, where $\tau'_1, \ldots, \tau'_{n_3}$ are independent geometric random variables, such that, for each $1 \leq i \leq n_3$, $\mathbb{E}[\tau'_i] = n_3/(p_{n_3} i)$. So,

$$\mathbb{Pr}[G_4 > \delta n_0] \leq \mathbb{Pr}\Big[\sum_{1\leq i\leq n_3} \tau'_i > n_3 + \delta n_0\Big] \tag{4.26}$$

We can bound from above the right-hand side of (4.26) as follows. Consider the following balls-and-bins process (which is a variation of the Coupon Collector's Problem [60]). We start with n_3 empty bins. In each step, we flip a biased coin that has probability of heads p_{n_3}. If the outcome is heads we place a ball in one of the bins chosen independently and uniformly at random among all the bins (empty or not); if the outcome is tails then nothing happens. The process finishes when each bin contains at least one ball. It is straightforward to verify that the total number of steps J until the process finished is equal in distribution to $\sum_{1\leq i\leq n_3} \tau'_i$; so,

$$\mathbb{Pr}\Big[\sum_{1\leq i\leq n_3} \tau'_i > n_3 + \delta n_0\Big] = \mathbb{Pr}[J > n_3 + \delta n_0] \tag{4.27}$$

Consider now a fixed bin, and let τ'' be the earliest step when a ball is placed in that bin. Since in each step a ball is placed in that bin independently with probability p_{n_3}/n_3, the probability that the bin is still empty after $n_3 + \lfloor \delta n_0 \rfloor$ steps is

$$\begin{aligned}\mathbb{Pr}[\tau'' > n_3 + \delta n_0] = \Big(1 - \frac{p_{n_3}}{n_3}\Big)^{n_3 + \lfloor \delta n_0 \rfloor} &\leq \exp\Big(-p_{n_3}\Big(1 + \frac{\lfloor \delta n_0 \rfloor}{n_3}\Big)\Big) \\ &\leq \exp\Big(-p_{n_3}\frac{\delta n_0}{n_3}\Big) \end{aligned} \tag{4.28}$$

Note that, by (4.24),

$$p_{n_3} = 1 - \Big(1 - \frac{n_3}{n_0}\Big)^{\lambda_+(\kappa+1)} \geq \lambda_+(\kappa+1)\frac{n_3}{n_0}\Big(1 - \lambda_+(\kappa+1)\frac{n_3}{2n_0}\Big) \geq \lambda_+(\kappa+1)\frac{n_3}{n_0}\Big(1 - \frac{1}{2\ln\kappa}\Big)$$

where the second relation was obtained using the fact that $(1-\epsilon)^k \leq 1 - k\epsilon + (k\epsilon)^2/2$, when $k\epsilon \leq 1$. By applying this to (4.28), we obtain

$$\mathbb{Pr}[\tau'' > n_3 + \delta n_0] \leq \exp\Big(-\Big(1 - \frac{1}{2\ln\kappa}\Big)\delta\lambda_+(\kappa+1)\Big)$$

Since the probability that at least one bin (of the n_3) is empty after $n_3 + \lfloor \delta n_0 \rfloor$ steps is at most $n_3 \cdot \mathbb{Pr}[\tau'' > n_3 + \delta n_0]$,

$$\mathbb{Pr}[J > n_3 + \delta n_0] \leq n_3 \cdot \exp\left(-\left(1 - \frac{1}{2\ln\kappa}\right)\delta\lambda_+(\kappa+1)\right)$$

Combining this with (4.26) and (4.27) yields the desired result. ■

We now combine Claims 4.11–4.14 to derive a lower bound for $\mathbb{Pr}[\sum_{i=1}^{n_0}(\tau_i - 1) \leq \gamma n_0]$. Let

$$\delta = \gamma - \frac{\kappa+1}{\kappa(\kappa-1)} - \frac{n_1(e+1)}{n_0 e(e-1)} - \frac{2n_2\ln\kappa}{n_0} \tag{4.29}$$

Since $\lambda_+(k) = \Theta(k)$, it is easy to see that

$$\delta = \gamma - \mathrm{O}\left(\frac{\ln\kappa}{\kappa}\right) \tag{4.30}$$

and, thus, $\delta > 0$, for all large enough κ. By (4.29) and the fact that $\sum_{i=1}^{n_0}(\tau_i - 1) = \sum_{i=1}^{4} G_i$,

$$\begin{aligned}\mathbb{Pr}\left[\sum_{i=1}^{n_0}(\tau_i - 1) \leq \gamma n_0\right] &\geq \mathbb{Pr}\left[\left\{G_1 \leq \frac{\kappa+1}{\kappa(\kappa-1)}n_0\right\} \cap \left\{G_2 \leq \frac{e+1}{e(e-1)}n_1\right\} \cap \right.\\ &\qquad \left. \left\{G_3 \leq 2n_2\ln\kappa\right\} \cap \left\{G_4 \leq \delta n_0\right\}\right]\\ &\geq 1 - \mathbb{Pr}\left[G_1 > \frac{\kappa+1}{\kappa(\kappa-1)}n_0\right] - \mathbb{Pr}\left[G_2 > \frac{e+1}{e(e-1)}n_1\right]\\ &\quad - \mathbb{Pr}[G_3 > 2n_2\ln\kappa] - \mathbb{Pr}[G_4 > \delta n_0]\end{aligned}$$

So, by Claims 4.11–4.14 and the fact that $\frac{n_0}{2\kappa^3}$, $\frac{n_1(e-2)}{2e^3(e-1)}$, and $\frac{n_2}{64\ln^2\kappa}$ are all $\omega\big(\lambda_+(\kappa+1)\big)$,

$$\mathbb{Pr}\left[\sum_{i=1}^{n_0}(\tau_i - 1) \leq \gamma n_0\right] = 1 - \mathrm{O}(n_3 e^{-\left(1-\frac{1}{2\ln\kappa}\right)\delta\lambda_+(\kappa+1)})$$

and, by (4.30),

$$\mathbb{Pr}\left[\sum_{i=1}^{n_0}(\tau_i - 1) \leq \gamma n_0\right] = 1 - \mathrm{O}(n_3 e^{-(1-\mathrm{O}(1/\ln\kappa))\gamma\lambda_+(\kappa+1)})$$

This, together with (4.25), yields the desired result.

4.6 Proof of Theorem 4.1

We begin by reducing the set of initial partitions we need to consider. Specifically, we show that to prove Theorem 4.1 it suffices to show the following variation of it, which considers only $\mathcal{S}$-process that have a borderline-safe initial partition. Recall from Section 4.3 that the borderline-safe partition of size n, denoted $\check{\Pi}_n$, is the (unique) safe partition such that $\check{\Pi}_n \preceq S$, for all $S \in \mathbf{F}_n$.

Lemma 4.15. *For any long enough $\mathcal{S}$-process such that $S_0 = \check{\Pi}_n$, with probability*

$$1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}) \tag{4.31}$$

there is $\tau \in [\varepsilon 2^{\mu_0}..8 \cdot 2^{\mu_0}]$ such that:

(i) S_τ is safe

(ii) for all $t \leq \tau$, $\mu_0 \leq \mu_t \leq \mu_0 + 1$ and $\xi_0 - 1 \leq \xi_t \leq \xi_0$.

Before we prove this lemma, we show that it implies Theorem 4.1. Consider an arbitrary (long enough) $\mathcal{S}$-process such that $S_0 \in \mathbf{F}_n$. We define a second $\mathcal{S}$-process as a function of the first one, as follows. (We use primed notation to denote the quantities associated with the second $\mathcal{S}$-process.) We let

$$S'_0 = \check{\Pi}_n \qquad N' = N \qquad \lambda'_{+/-} = \lambda_{+/-} \qquad g' = g$$

and, for each $1 \leq t < N'$,

$$U'_t = U_t \qquad V'_t = V_t \qquad Y'_t = Y_t$$

Clearly, the above define a valid $\mathcal{S}$-process. By Lemma 3.14 (and induction) we have that, for all t,

$$S'_t \preceq S_t \tag{4.32}$$

By Lemma 4.15, we have that, with some probability $p = 1 - \mathrm{O}(2^{\mu'_0} e^{-(1/4-\varepsilon)\lambda(\mu'_0)})$, there is $\tau' \in [\varepsilon 2^{\mu'_0}..8 \cdot 2^{\mu'_0}]$ such that:

(i′) $S'_{\tau'}$ is safe, and

(ii′) for all $t \leq \tau'$, $\mu'_0 \leq \mu'_t \leq \mu'_0 + 1$ and $\xi'_0 - 1 \leq \xi'_t \leq \xi'_0$.

By Lemma 4.4(a), $\mu'_0 \leq \mu_0 \leq \mu'_0 + 1$, so,

$$p = 1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

and the range where τ' takes on values is

$$[\varepsilon 2^{\mu'_0}..8 \cdot 2^{\mu'_0}] \subseteq [(\varepsilon/2) \cdot 2^{\mu_0}..8 \cdot 2^{\mu_0}]$$

We now show that if (i$'$) and (ii$'$) hold (for some τ') then (i)–(iii) of Theorem 4.1 hold, for $\tau = \tau'$. By (4.32) and Lemma 4.2, (i$'$) implies (i). By (4.32) and Lemma 3.4, we have that, for all $t \leq \tau'$,

$$\mu'_t \leq \mu_t \quad \text{and} \quad \xi_t \leq \xi'_t$$

So, if (ii$'$) holds then (ii) holds, since

$$\varrho_t = \xi_t - \mu_t \leq \xi'_t - \mu'_t \leq \xi'_0 - \mu'_0 \leq 2$$

Also, if (ii$'$) holds then (iii) holds, since

$$\xi_t \leq \xi'_t \leq \xi'_0 \leq \xi_0 + 1$$

where the last relation holds because of Lemma 4.4(a). Combining all the above yields that, with probability at least $p = 1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$, there is $\tau \in [(\varepsilon/2) \cdot 2^{\mu_0} .. 8 \cdot 2^{\mu_0}]$ such that (i)–(iii) hold.

Proof of Lemma 4.15

For simplicity of exposition, we will assume (without loss of generality) that the length of the $\mathcal{S}$-process is larger than $8 \cdot 2^{\mu_0}$; in particular, $N \geq 16 \cdot 2^{\mu_0}$. (If this is not the case, we can "extend" the process to the desired length.) Let $\mathcal{E}$ denote the event whose probability we want to bound, i.e.,

$$\text{"(i) and (ii) hold, for some } \tau \in [\varepsilon 2^{\mu_0} .. 8 \cdot 2^{\mu_0}]\text{"}$$

We distinguish three different cases, depending on $S_0 = \check{\Pi}_n$.

Case 1: $\min\{\ell_{\mu_0}(S_0), \ell_{\xi_0}(S_0)\} \geq 1/4$.
We establish a lower bound for $\mathbb{Pr}[\mathcal{E}]$ by identifying a collection of "good" events such that, if all these events occur, then $\mathcal{E}$ also occurs. Then we bound from below the probability of each of these good events, and show that the probability of their intersection is at least as in (4.31). Roughly speaking, these events say that all the blocks that are split in the first $\varepsilon 2^{\mu_0}$ steps have depth μ_0, and all blocks merged have depth ξ_0.

Note that, since the result we want to show is asymptotic, as $n \to \infty$, we can assume that n is larger than any given constant. So, by Lemma 4.4(b),

$$\varrho_0 \in \{1, 2\} \tag{4.33}$$

We begin with some definitions we will use to describe the good events. For $i \geq 1$, let η_i be the step when the i-th $\textsc{addBlk}_{\mathbf{S}}$ operation occurs; i.e.,

$$\eta_i = \inf\{t \,:\, |\{j \leq t \,:\, V_j = +\}| = i\}$$

Similarly, let η'_i be the step when the i-th $\textsc{rmBlk}_{\mathbf{S}}$ operation takes place; i.e.,

$$\eta'_i = \inf\{t \,:\, |\{j \leq t \,:\, V_j = -\}| = i\}$$

We assume, without loss of generality, that $\eta_i, \eta'_i < \infty$, for all i in the range of steps we are interested in; i.e., for $i \leq 8 \cdot 2^{\mu_0}$. (We can always achieve that by appropriately modifying the adversary for $t > 8 \cdot 2^{\mu_0}$ — since we assumed $N \geq 16 \cdot 2^{\mu_0}$.) For those i, we define

$$Z_i = \langle Z_{i,1}, Z_{i,2}, \ldots \rangle = Y_{\eta_i} \qquad Z'_i = \langle Z'_{i,1}, Z'_{i,2}, \ldots \rangle = Y_{\eta'_i}$$

Note that $\langle Z_1, \ldots, Z_{8 \cdot 2^{\mu_0}} \rangle$ is a random point-array, and so is $\langle Z'_1, \ldots, Z'_{8 \cdot 2^{\mu_0}} \rangle$.

We are now ready to describe the good events. Let $\mathcal{E}_{1,i}$ be the event:

$$\text{“}\min\{Z_{i,j} \,:\, j \leq \lambda_+(\mu_0 + 1)\} < 1/4 - \varepsilon\text{”}$$

and $\mathcal{E}_{2,i}$ be the event:

$$\text{“}\max\{Z'_{i,j} \,:\, j \leq \lambda_-(\mu_0)\} \geq 3/4 + \varepsilon\text{”}$$

Since Z_i is a random point-vector,

$$\mathbb{Pr}[\mathcal{E}_{1,i}] = 1 - (1 - 1/4 + \varepsilon)^{\lambda_+(\mu_0+1)} \geq 1 - e^{-(1/4-\varepsilon)\lambda_+(\mu_0+1)} \tag{4.34}$$

Similarly,

$$\mathbb{Pr}[\mathcal{E}_{2,i}] \geq 1 - e^{-(1/4-\varepsilon)\lambda_-(\mu_0)} \tag{4.35}$$

Now let,

$$\mathcal{E}' = \bigcap_{i=1}^{\kappa} (\mathcal{E}_{1,i} \cap \mathcal{E}_{2,i})$$

where

$$\kappa = \varepsilon 2^{\mu_0}$$

(Note that κ is an integer, for large n.) Then, by (4.34) and (4.35),

$$\begin{aligned}\mathbb{Pr}[\mathcal{E}'] = 1 - \mathbb{Pr}\left[\bigcup_{i=1}^{\kappa} (\bar{\mathcal{E}}_{1,i} \cup \bar{\mathcal{E}}_{2,i})\right] &\geq 1 - \sum_{i=1}^{\kappa} \big(\mathbb{Pr}[\mathcal{E}_{1,i}] + \mathbb{Pr}[\mathcal{E}_{2,i}]\big) \\ &\geq 1 - \kappa(e^{-(1/4-\varepsilon)\lambda_+(\mu_0+1)} + e^{-(1/4-\varepsilon)\lambda_-(\mu_0)}) \\ &= 1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}) \end{aligned} \tag{4.36}$$

We complete the proof by showing that $\mathcal{E}'$ implies $\mathcal{E}$. Suppose that $\mathcal{E}'$ occurs. We show that, in the fist κ steps, all $\textsc{addBlk}_\mathbf{S}$ operations split blocks of depth μ_0 and all $\textsc{rmBlk}_\mathbf{S}$ operations merge blocks of depth ξ_0. Assume (for contradiction) that some block of depth $\neq \mu_0$ is split, or some pair of blocks of depth $\neq \xi_0$ are merged during the first κ steps. Let t_0 be the earliest step when this happens. Then, since all blocks split in the first $t_0 - 1$ steps have depth μ_0, and all blocks merged have depth ξ_0, and (by (4.33)) $\varrho_0 > 0$, we have

$$\mu_{t_0-1} \geq \mu_0 \quad \text{and} \quad \xi_{t_0-1} \leq \xi_0$$

Also, since at most $t_0 - 1$ blocks of depth μ_0 (and length $2^{-\mu_0}$) are split and no blocks of depth μ_0 are merged,

$$\ell_{\mu_0}(S_{t_0-1}) \geq \ell_{\mu_0}(S_0) - (t_0 - 1)2^{-\mu_0} \geq 1/4 - (\kappa - 1)2^{-\mu_0} > 1/4 - \varepsilon \tag{4.37}$$

(thus, $\mu_{t_0-1} = \mu_0$). Likewise,

$$\ell_{\xi_0}(S_{t_0-1}) \geq \ell_{\xi_0}(S_0) - (t_0 - 1)2^{-\xi_0+1} \geq 1/4 - (\kappa - 1)2^{-\mu_0} > 1/4 - \varepsilon \tag{4.38}$$

(and $\xi_{t_0-1} = \xi_0$). Now, if an $\textsc{addBlk}_\mathbf{S}$ operation occurs in step t_0 then, by (4.37) and the event $\bigcap_{i=1}^{\kappa} \mathcal{E}_{1,i}$,

$$\min\{Y_{t_0,j} \,:\, j \leq \lambda_+(\mu_0 + 1)\} < \ell_{\mu_0}(S_{t_0-1})$$

Thus, by Lemma 4.5(a), a block of depth μ_0 is split in step t_0. Similarly, if a $\textsc{rmBlk}_\mathbf{S}$ operation occurs in step t_0, instead, then by (4.38), event $\bigcap_{i=1}^{\kappa} \mathcal{E}_{2,i}$, and Lemma 4.5(b), the blocks merged have depth ξ_0. So, in either case we have a contradiction. Therefore, in the fist κ steps, all $\textsc{addBlk}_\mathbf{S}$ operations split blocks of depth μ_0 and all $\textsc{rmBlk}_\mathbf{S}$ operations merge blocks of depth ξ_0. From this, and the fact that S_0 is safe with $\varrho_0 > 0$, it is immediate that (i) and (ii) hold for $\tau = \kappa$. Therefore, if $\mathcal{E}'$ occurs then $\mathcal{E}$ occurs, and, thus,

$$\mathbb{Pr}[\mathcal{E}] \geq \mathbb{Pr}[\mathcal{E}']$$

Combining this and (4.36) yields the desired result.

Case 2: $\ell_{\mu_0}(S_0) < 1/4$.

As in Case 1, we identify a number of good events (events $\mathcal{E}_1$–$\mathcal{E}_3$) such that if they all occur then $\mathcal{E}$ also occurs. Then, we establish a lower bound for the probability of the intersection of these good events, thus, obtaining a lower bound for the probability of $\mathcal{E}$, as well. Note that, by Lemma 4.4(b) and the case hypothesis, we have that (for all large enough n)

$$\varrho_0 = 2 \tag{4.39}$$

and

$$\ell_{\xi_0}(S_0) = 1/4 + \varepsilon \tag{4.40}$$

Intuitively, the good events we describe say that: each of the first $\Theta(2^{\mu_0})$ ADDBLK$_{\mathbf{S}}$ operations splits a block of depth μ_0 or $\mu_0 + 1$, if the partition it is applied to has $\ell_{\leq \mu_0+1} \geq 1/4 - \varepsilon$ (event $\mathcal{E}_1$); each of the first $\Theta(2^{\mu_0})$ RMBLK$_{\mathbf{S}}$ operations merges blocks of depth $\geq \xi_0$, if the partition it is applied to has $\ell_{\geq \xi_0} \geq 1/4 - \varepsilon$ (event $\mathcal{E}_2$); and, in $\Theta(2^{\mu_0})$ steps, either all blocks of depth μ_0 have been split or some pair of blocks of depth $< \xi_0$ is merged (event $\mathcal{E}_3$).

More precisely, for $i \geq 1$, we define η_i, η'_i, Z_i, Z'_i, and events $\mathcal{E}_{1,i}$ and $\mathcal{E}_{2,i}$ as in Case 1. We define events $\mathcal{E}_1$ and $\mathcal{E}_2$ as

$$\mathcal{E}_1 = \bigcap_{i=1}^{\kappa} \mathcal{E}_{1,i} \qquad \mathcal{E}_2 = \bigcap_{i=1}^{\kappa'} \mathcal{E}_{2,i}$$

where

$$\kappa = s_{\mu_0}(S_0) + 2^{\mu_0} \qquad \kappa' = 2^{\mu_0+1}$$

We let $\mathcal{E}_3$ be the event:

$$\text{“}a \leq \kappa\text{”}$$

where a is the A-time of $\big\langle S_0, \langle Z_1, \ldots, Z_\kappa \rangle \big\rangle$. We now compute lower bounds for the probabilities of these events and their intersection. By (4.34) and (4.35), we have

$$\mathbb{Pr}[\mathcal{E}_1] \geq 1 - \kappa e^{-(1/4-\varepsilon)\lambda_+(\mu_0+1)}$$

and

$$\mathbb{Pr}[\mathcal{E}_2] \geq 1 - \kappa' e^{-(1/4-\varepsilon)\lambda_-(\mu_0)}$$

Since $\langle Z_1, \ldots, Z_\kappa \rangle$ is a random point-array, by applying Lemma 4.6 (for $\gamma = 1$), we obtain

$$\mathbb{Pr}[\mathcal{E}_3] = 1 - \mathrm{O}\big(2^{\mu_0} e^{-(1-\mathrm{O}(1/\ln \mu_0))\lambda_+(\mu_0+1)}\big)$$

Combining the above bounds, we get

$$\mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3] = 1 - \mathrm{O}\big(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}\big) \tag{4.41}$$

In Claim 4.16, below, we show that if all of $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ occur then so does $\mathcal{E}$; thus,

$$\mathbb{Pr}[\mathcal{E}] \geq \mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3]$$

This, together with (4.41), yields the desired lower bound for $\mathbb{Pr}[\mathcal{E}]$.

Claim 4.16. *If $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ occurs then $\mathcal{E}$ occurs.*

Proof. Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ occurs. We define the following quantities:

- J is the earliest step when blocks of depth $< \xi_0$ are merged. ($J = \infty$ if no such step exists.)
- $K = \min\{\eta_\kappa,\, J - 1\}$.
- M is the number of $\textsc{addBlk}_{\mathbf{S}}$ operations performed in the first K steps.
- M' is the number of $\textsc{addBlk}_{\mathbf{S}}$ operations in the first K steps that split blocks of depth $\neq \mu_0$.

Note that $K < \infty$, because $\eta_\kappa < \infty$. Indeed, since fewer than $\kappa + |S_0|$ $\textsc{rmBlk}_{\mathbf{S}}$ operations take place before κ $\textsc{addBlk}_{\mathbf{S}}$ operations occur,

$$\eta_\kappa < 2\kappa + |S_0| \leq 8 \cdot 2^{\xi_0} < N \tag{4.42}$$

Since $K < J$,

$$\text{all the blocks merged in the first } K \text{ steps have depth} \geq \xi_0 \tag{4.43}$$

So, since, by (4.39),

$$\xi_0 = \mu_0 + 2$$

$\langle S_0, S_{\eta_1}, \ldots, S_{\eta_M} \rangle$ is a prefix of some sequence in $\mathcal{A}_{S_0, \langle Z_1, \ldots, Z_\kappa \rangle}$. But, by event $\mathcal{E}_3$, in each of these sequences, all (the $s_{\mu_0}(S_0)$) blocks of depth μ_0 are split during the first κ $\textsc{addBlk}_{\mathbf{S}}$ operations. So, the number of $\textsc{addBlk}_{\mathbf{S}}$ operations, among the first $M \leq \kappa$ such operations, that split blocks of depth $\neq \mu_0$ is

$$M' \leq \kappa - s_{\mu_0}(S_0) = 2^{\mu_0} \tag{4.44}$$

For all $t \leq K$, then, we have

$$\ell_{\leq \mu_0 + 1}(S_t) \geq \ell_{\leq \mu_0 + 1}(S_0) - M' 2^{-(\mu_0 + 1)} \geq \bigl(1 - \ell_{\xi_0}(S_0)\bigr) - 1/2 = 1/4 - \varepsilon \tag{4.45}$$

by (4.40). Combining this and event $\mathcal{E}_1$, we obtain that for all $i \leq M$,

$$\min\{Z_{i,j} \,:\, j \leq \lambda_+(\mu_0 + 1)\} < \ell_{\leq \mu_0 + 1}(S_{\eta_i})$$

which, by Lemma 4.5(a), yields

$$\text{all the blocks split in the first } K \text{ steps have depth} \leq \mu_0 + 1 \tag{4.46}$$

Also, by (4.45), for all $t \leq K$, $\ell_{\leq \mu_0+1}(S_t) > 0$, so,

$$\mu_t \leq \mu_0 + 1 \tag{4.47}$$

At this point we distinguish two cases, depending on which of η_κ or $J-1$ is smaller.

Case A: $K = \eta_\kappa$.
Then, $M = \kappa$. So, by (4.44), $M - M' \geq s_{\mu_0}(S_0)$, which, together with (4.43), yields

$$\mu(S_K) \geq \mu_0 + 1 \tag{4.48}$$

Combining this with (4.43), (4.46), and (4.47), yields that (i) and (ii) hold for $\tau = \eta_\kappa$. Also, $\eta_\kappa \geq \kappa > \varepsilon 2^{\xi_0}$ and, by (4.42), $\eta_\kappa < 8 \cdot 2^{\xi_0}$. Therefore, $\mathcal{E}$ occurs.

Case B: $K = J - 1 < \eta_\kappa$.
We show that

$$\ell_{\xi_0}(S_K) < 1/4 - \varepsilon \tag{4.49}$$

Suppose (for contradiction) that $\ell_{\xi_0}(S_K) \geq 1/4 - \varepsilon$. Then, by (4.43) and (4.46), the number of rmBlks operations performed in the first K steps is (at most)

$$\frac{1}{2}\big(s_{\xi_0}(S_0) - s_{\xi_0}(S_K)\big) + M' \leq \left(\left(\frac{1}{4}+\varepsilon\right) - \left(\frac{1}{4}-\varepsilon\right)\right)2^{\xi_0-1} + 2^{\mu_0} = (1+4\varepsilon)2^{\mu_0} < \kappa'$$

where the second relation holds because of (4.40) and (4.44). Combining that, event $\mathcal{E}_2$, the assumption that $\ell_{\xi_0}(S_K) \geq 1/4 - \varepsilon$, and Lemma 4.5(b), we obtain that the pair of blocks merged in step J have depth ξ_0, which contradicts the definition of J. So, (4.49) holds.
Combining (4.49), (4.43), (4.46), and (4.47), yields that (i) and (ii) hold, for $\tau = J-1$. Also, by (4.40) and (4.49),

$$J - 1 > \varepsilon 2^{\xi_0}$$

since $\varepsilon 2^{\xi_0}$ is the minimum number of rmBlks operations required to reduce ℓ_{ξ_0} by 2ε, and

$$J - 1 < \eta_\kappa \leq 8 \cdot 2^{\xi_0}$$

by (4.42). Hence, $\mathcal{E}$ occurs. ■

Case 3: $\ell_{\xi_0}(S_0) < 1/4$.
The proof in this case is similar to that of Case 2 and is omitted.

Chapter 5

Analysis – Part III: Starting from an unbalanced partition

In this chapter we continue the study of $\mathcal{S}$-processes, which we started in Chapter 4. So far, we have looked at the case where the $\mathcal{S}$-process begins from a safe initial partition. Here we consider the complementary case, where the initial partition is not safe, and we provide an upper bound on the number of steps required to reach a safe partition with high probability.[1]

In Section 5.1, we describe the main result of this chapter. An outline of its proof is given in Section 5.2. In Section 5.3, we introduce some definitions. Sections 5.4–5.7 contain the various steps of the proof. We combine all these steps in Section 5.8.

5.1 Statement of the main result: from a non-safe to a safe partition

In this chapter we consider $\mathcal{S}$-processes that start from an *arbitrary* non-safe initial partition S_0; specifically, we do not impose any restrictions on how "unbalanced" S_0 may be. As in Chapter 4, we assume that the sampling-size functions are $\lambda_+(d), \lambda_-(d) = \Theta(d)$, with the constants involved being sufficiently large; the processes' length N is large enough; and the strategy of the adversary and the precision g can be arbitrary. For any such $\mathcal{S}$-process, we show that, with probability $1 - (1/2^{\xi_0})^{\Theta(1)}$, a safe partition is reached within $\mathrm{O}(\xi_0 2^{\xi_0})$ steps.

[1] Throughout this chapter whenever we say "partition" we mean "sorted binary partition."

The formal statement of this result is as follows.

Theorem 5.1. *Consider an $\mathcal{S}$-process such that, for all $k \geq 0$,*

$$\lambda_+(k) \geq \max\{8(\ln 2)k,\, \beta\} \quad \textit{and} \quad \lambda_-(k) \geq \max\{8k,\, \beta\} \tag{5.1}$$

where β is a sufficiently large constant. If the $\mathcal{S}$-process is long enough then, with probability

$$1 - \mathrm{O}(\xi_0^{\mathrm{O}(1)} 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda(\xi_0)}) \tag{5.2}$$

there is $\tau \leq c\xi_0 2^{\xi_0}$, where c is a positive constant, such that

(i) S_τ is safe

(ii) for all $t \leq \tau$, $\xi_\tau \leq \xi_0 + 2$.

By long enough $\mathcal{S}$-process we mean that it has length $N \geq c\xi_0 2^{\xi_0}$. (See the remark at the end of Section 3.5.2.) ε is the constant 1/16 from the definition of a safe partition.

Recall that in the statement of Theorem 4.1, in Section 4.1, both the probability (4.1) and the range for τ are expressed in terms of μ_0, which is roughly the same as $\log|S_0|$ and ξ_0, since S_0 is safe. In Theorem 5.1, the corresponding quantities (i.e., the probability (5.2) and the bound for τ) are expressed in terms of ξ_0. In this case, however, ξ_0 may be much larger than $\log|S_0|$, depending on how unbalanced S_0 is; in the extreme, $\xi_0 = |S_0| - 1$. Similarly to Theorem 4.1, the big-oh term in (5.2) can be made smaller than any given polynomial of $2^{-\xi_0}$, by using sufficiently large λ_+, λ_-.

5.2 Outline of the proof

Recall that the proof of Theorem 4.1 depends critically on the fact that, for any partition S with $\mu, \xi = \Theta(\log|S|)$, an $\textsc{addBlk}_\mathbf{S}$ or $\textsc{rmBlk}_\mathbf{S}$ operation applied to S involves $\Theta(\log|S|)$ random probes. More precisely, the theorem is based on the fact that each $\textsc{addBlk}_\mathbf{S}$ operation splits a block of depth d only if d is smaller or equal to the depths of $\Theta(d)$ randomly probed blocks; and each $\textsc{rmBlk}_\mathbf{S}$ operation merges a pair of sibling blocks that are smaller or equal to $\Theta(\mu) = \Theta(\xi)$ randomly probed blocks. Theorem 5.1, however, considers partitions S for which it may not be true that $\mu, \xi = \Theta(\log|S|)$. For such a partition, an $\textsc{addBlk}_\mathbf{S}$ operation that splits a block of depth d also involves $\Theta(d)$ random probes; so, as before it has a strong tendency to split larger blocks and, thus, "improve" the balance of S. However, the number of random probes executed in an $\textsc{rmBlk}_\mathbf{S}$ operation, that is $\Theta(\mu)$, may be significantly smaller than ξ — in the worst case, $\xi = \Theta(|S|)$ and $\mu = \mathrm{O}(1)$. As a result, rmBlk$_\mathbf{S}$ operations may tend to improve balance "at a slower rate" than if $\Theta(\xi)$ random probes were used, or even to "deteriorate" it.

The basic intuition behind Theorem 5.1 is that addBlk$_\mathbf{S}$ operations tend to improve balance faster than rmBlk$_\mathbf{S}$ operations tend to deteriorate it. Roughly speaking, addBlk$_\mathbf{S}$ operations almost always split larger blocks, while rmBlk$_\mathbf{S}$ operations merge smallest blocks at a rate that is $\mathrm{O}(\xi)$ lower than if $\Theta(\xi)$ random probes were used. If $\Theta(\xi)$ probes were used for each rmBlk$_\mathbf{S}$ operation then it would take $\mathrm{O}(2^{\xi_0})$ steps to reach a safe partition starting from an arbitrary S_0. So, in an $\mathcal{S}$-process a safe partition is reached within at most $\mathrm{O}(\xi_0 2^{\xi_0})$ steps.

Informally, the proof proceeds as follows. We identify a number of classes of progressively "more balanced" partitions, where the last class is that of safe partitions. (We describe these classes and some of their properties in Section 5.3.) For each of these classes we establish a probabilistic upper bound on the number of steps required to reach a partition from this class starting form a partition in the previous class. Combining these results yields the bound of Theorem 5.1. Below we give a more detailed exposition. For simplicity we assume that the size of the current partition $|S_t|$ remains roughly the same.

First we compute and upper bound for the number of steps required to get from an arbitrary S_0, to a partition S_0' where almost all blocks have depths $\xi(S_0')$ or $\xi(S_0')-1$. Combining the facts that: (1) while $\ell_{\leq \xi_t - 2}(S_t)$ is not very small, almost all addBlk$_\mathbf{S}$ operations split blocks of depth $\leq \xi_t - 2$; and (2) with high probability, the number of rmBlk$_\mathbf{S}$ operation required to merge all smallest blocks of a partition S, if no blocks of depth $> \xi - 2$ are split, is $\mathrm{O}(\xi 2^{\xi})$ (by the Coupon Collector's Problem [60]), we obtain a bound of $\mathrm{O}(\xi_0 2^{\xi_0})$ steps for the number of steps to get from S_0 to S_0'. S_0' differs from a safe partition in that the distribution of depths may have a long thin "tail" to the left; i.e., $\ell_{\leq \xi - 2}(S_0')$[2] is small but it may be $\mu(S_0') \ll \xi(S_0')$. Next we compute and upper bound for the number of steps required until this tail becomes short, specifically, until we reach a partition S_0'' that has $\mu = \xi - \mathrm{o}(\xi)$ and still $\ell_{\leq \xi - 2}$ is small. The intuition is that most rmBlk$_\mathbf{S}$ operations will merge some of the numerous small blocks, so addBlk$_\mathbf{S}$ operations will quickly shorten the tail (in at most a linear number of steps). To get to a safe partition we still have to eliminate the short thin tail of S_0''. Note that this tail does not essentially affect the outcome of rmBlk$_\mathbf{S}$ operations, since now $\Theta(\mu_t) = \Theta(\xi_t)$. Hence, a safe partition is reached in at most a linear number of additional steps.

[2] Recall that $\ell_{\leq \xi - 2}(S_0') \equiv \ell_{\leq \xi(S_0') - 2}(S_0')$, by the convention we introduced at the end of Section 3.2.

We describe the above three steps in Sections 5.4, 5.5, and 5.7 respectively. We combine the first two in Section 5.6, and we put all the pieces together in Section 5.8.

5.3 Tailed, left-heavy, and almost-safe partitions

In this section, we define various classes of partitions, and describe some properties of them. Each partition, depending on the value of $\ell_{\leq \xi - 2}$, is classified as:
thick-tailed if

$$\ell_{\leq \xi-2} > 1/4 + \varepsilon$$

thin-tailed if

$$\ell_{\leq \xi-2} < 1/4 - \varepsilon$$

normal-tailed if

$$|\ell_{\leq \xi-2} - 1/4| \leq \varepsilon$$

where $\varepsilon = 1/16$ (as in the definition of a safe partition). A partition S is *left-heavy* if $S \succeq S'$ for some S' such that:

$$\xi(S') = \xi(S) \quad \text{and} \quad S' \text{ is either thick-tailed or normal-tailed}$$

Obviously, a partition that is thick-tailed or normal-tailed is also left-heavy. The converse, however, is not always true; there are thin-tailed partitions that are left-heavy. For example, the thin-tailed partition S with

$$\mu = 4 \qquad \xi = 6 \qquad \ell_4 = 1/4 - 2\varepsilon \qquad \ell_5 = 3\varepsilon/2$$

is left-heavy, because the partition $S' = \mathbb{V}_{4 \to 5}(S) = \mathbb{S}_5\, \mathbb{M}_5(S)$ is normal-tailed, $\xi(S') = \xi(S)$, and $S \succeq S'$. The next lemma states two useful results about left-heavy partitions. Part (a) states a sufficient condition for S to be left-heavy; and part (b) says that if S is not left heavy then it is more balanced than some left-heavy partition of $\xi = \xi(S) + 1$.

Lemma 5.2. *For all $S \in \mathbf{S}$ such that $\xi(S) \geq 6$,*[3]

(a) if $\ell_{\leq \xi-1}(S) > 1/4 + \varepsilon$ then S is left-heavy.

(b) if S is not *left-heavy then $S \succeq S'$ for some S' such that $\xi(S') = \xi(S) + 1$ and S' is either thick-tailed or normal-tailed.*

[3] In fact, part (a) holds even when $\xi(S) = 4$ or 5, and part (b) holds even when $\xi(S) = 3, 4$ or 5.

Proof.

(a) Let S' be the partition obtained from S by splitting $1/3$ of the blocks of depth $\xi(S)-1$ (into blocks of depth $\xi(S)$) and merging the other $2/3$ (into blocks of depth $\xi(S)-2$). More precisely,

$$S' = \mathbb{V}^k_{d-2\to d-1}(S)$$

where

$$d=\xi(S) \qquad k=\lfloor s_{d-1}(S)/3\rfloor$$

Recall that operation $\mathbb{V}_{d-2\to d-1}$ splits a block of depth $d-1$, and merges two sibling blocks of depth $d-1$. So, it reduces s_{d-1} by 3, and increases s_{d-2} and s_d by 1 and 2, respectively. Therefore,

$$\begin{aligned}\ell_{\le d-2}(S') &= \ell_{\le d-2}(S) + k2^{-(d-2)}\\ &\ge \ell_{\le d-2}(S) + (s_{d-1}(S)/3 - 2/3)\cdot 2^{-(d-2)}\\ &= \ell_{\le d-2}(S) + (2/3)\cdot \ell_{d-1}(S) - (2/3)\cdot 2^{-(d-2)}\\ &\ge (2/3)\cdot \ell_{\le d-1}(S) - (2/3)\cdot 2^{-(d-2)}\end{aligned}$$

Since $d>4$, the assumption that $\ell_{\le d-1}(S) > 1/4+\varepsilon$ is equivalent to

$$\ell_{\le d-1}(S) \ge 1/4+\varepsilon+2^{-(d-1)}$$

so,

$$\ell_{\le d-2}(S') \ge (2/3)\cdot(1/4+\varepsilon-2^{-(d-1)}) = 1/4-\varepsilon+(1/3)\cdot(1/16-2^{-(d-2)}) \ge 1/4-\varepsilon$$

where the last inequality holds because $d\ge 6$. This, together with the facts that $\xi(S')=\xi(S)$ and $S\succeq S'$, yields that S is left-heavy.

(b) Since S is not left-heavy, part (a) yields $\ell_{\le\xi-1}(S)\le 1/4+\varepsilon$; so (since $\xi(S)>1$), $s_\xi(S)\ge 3$. Let T be the partition obtained from S by splitting one block of depth $\xi(S)$ and merging two blocks of depth $\xi(S)$; i.e.,

$$T = \mathbb{V}_{\xi(S)-1\to\xi(S)}(S)$$

Note that $\xi(T)=\xi(S)+1$, and

$$\ell_{\le\xi-1}(T) = 1-2\cdot 2^{-\xi(T)} > 1/4+\varepsilon$$

so, by part (a), T is left-heavy. Hence, there is a partition S' such that $S'\preceq T\preceq S$, and S' is either thick-tailed or normal-tailed, and $\xi(S')=\xi(T)=\xi(S)+1$. ■

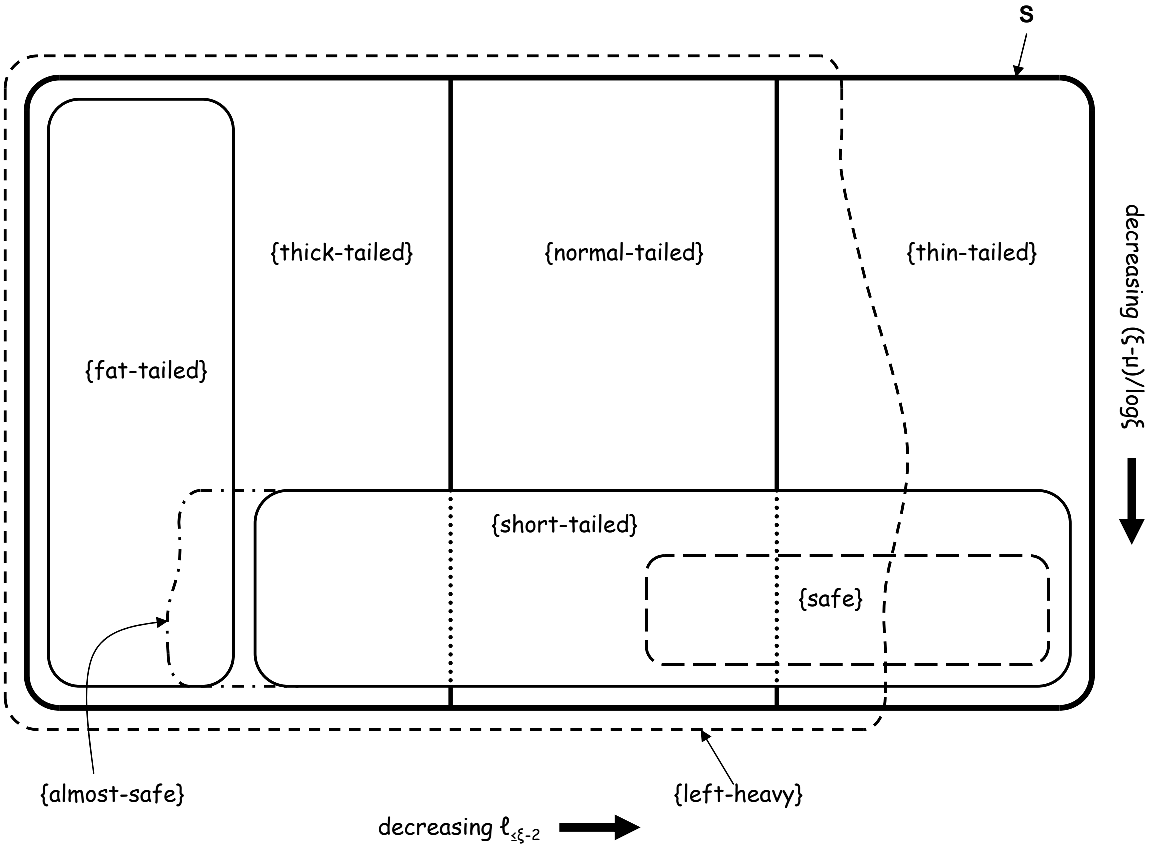

Figure 5.1: Relationship between the various classes of partitions.

We say that a partition is *fat-tailed* if

$$\ell_{\leq \xi-2} > 1/4 + 2\varepsilon$$

So, every fat-tailed partition is also thick-tailed. A partition is called *short-tailed* if it is not fat-tailed (i.e., $\ell_{\leq \xi-2} \leq 1/4 + 2\varepsilon$) and

$$\mu \geq \xi - 2\log \xi$$

A partition S is *almost-safe* if $S \succeq S'$ for some short-tailed S'. Note that we do not explicitly require that $\xi(S') = \xi(S)$, as in the definition of a left-heavy partition. However, the next lemma shows that this condition is implicit in the definition of an almost-safe partition.

Lemma 5.3. *If $S \in \mathbf{S}$ is almost-safe then $S \succeq S'$ for some short-tailed S' such that $\xi(S') = \xi(S)$.*

Proof Sketch. We show that for any short-tailed $T \in \mathbf{S}$ such that $T \preceq S$ and $\xi(T) > \xi(S)$, $T' = \hat{\mathbb{S}}_{\xi-2}\,\mathbb{M}_{\xi}(T)$ is also short-tailed and $T' \preceq S$. Then, we construct S' starting from any short-tailed partition $\preceq S$ and iteratively applying to it the pair of operations $\hat{\mathbb{S}}_{\xi-2}\,\mathbb{M}_{\xi}$, until the resulting partition has $\xi = \xi(S)$. ■

5.4 From a thick-tailed to a normal-tailed partition

In this section and the next, we prove two results that we use in Section 5.6 to establish a probabilistic upper bound on the number of steps required in an $\mathcal{S}$-process until we reach an almost-safe partition, starting from an arbitrary initial partition. Informally, here we show that if we start from a thick-tailed initial partition and λ_+ is sufficiently large then, with high probability, it takes at most $\mathrm{O}(\xi_0 2^{\xi_0})$ steps until either we reach a normal-tailed partition, or all smallest blocks of S_0 have been merged. The formal statement of this result is as follows.

Lemma 5.4. *For any long enough $\mathcal{S}$-process such that S_0 is thick-tailed, with probability*

$$1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4+\varepsilon)\lambda_+(\xi_0)})$$

there is $\tau \leq c\xi_0 2^{\xi_0}$, where c is a positive constant, such that

(i) all the blocks that are split in the first τ steps have depths $\leq \xi_0 - 2$, and

(ii) S_τ is normal-tailed, or S_τ is left-heavy and $\xi_\tau < \xi_0$.

The proof of this lemma is quite straightforward. Roughly speaking, by the Coupon Collector's Problem [60] it takes $\mathrm{O}(\xi_0 2^{\xi_0})$ steps until all smallest blocks of S_0 are merged, provided that no blocks of depth ξ_0 or $\xi_0 - 1$ are split during that time. But, by the result in Section 4.4, (for large λ_+) with high probability no blocks of depth $> \xi_0 - 2$ are split when $\ell_{\leq \xi_0 - 2}(S_t) > 1/4 + \varepsilon$. Therefore, in $\mathrm{O}(\xi_0 2^{\xi_0})$ steps, either all blocks of depth ξ_0 have been merged, or $\ell_{\leq \xi_0 - 2}(S_t)$ is decreased to $1/4 + \varepsilon$, and, thus, a normal-tailed partition is reached.

The actual proof is similar in structure to that of Theorem 4.1. We begin by introducing a variation of the concept of R-times.

5.4.1 $\tilde{R}$-times

Recall from Section 4.5 that the R-time of $\langle S, W\rangle$, where $\varrho(S) \leq 2$, is, roughly speaking, the maximum number of rmBlks operations required to merge all smallest blocks of S, when the sample points used are those in W. A rmBlks operation takes effect only if it merges blocks of depth $\xi(S)$ or $\xi(S) - 1$, and any number of blocks of depth $\leq \xi(S) - 2$ may be split between two $\textsc{rmBlk}_{\mathbf{S}}$ operations. Here we consider the variation where *all* $\textsc{rmBlk}_{\mathbf{S}}$ operations take effect, regardless of the depth of the blocks they merge. Also, we allow S to be an arbitrary partition; i.e, we drop the requirement that $\varrho(S) \leq 2$.

More formally, let $S \in \mathbf{S}$ and W be as in (4.8), i.e.,

$$W = \langle W_1, \ldots, W_{l_w} \rangle, \qquad \text{where } W_t = \langle W_{t,1}, \ldots, W_{t,h_w} \rangle \in I^{h_w}, \text{ for each } t = 1, \ldots, l_w \tag{5.3}$$

and h_w is large enough that the operations we describe below are well defined. We denote by $\tilde{\mathcal{R}}_{S,W}$ the class of all sequences of partitions $\langle T_0, \ldots, T_{l_w} \rangle$ such that

$$T_t = \begin{cases} S, & \text{if } t = 0 \\ \textsc{rmBlk}_{\mathbf{S}}(T'_{t-1}, W_t), & \text{otherwise} \end{cases}$$

where

$$T'_t = \mathbb{S}_{k_1^t} \cdots \mathbb{S}_{k_{m_t}^t}(T_t), \quad \text{for some } m_t \geq 0, \text{ and } k_1^t, \ldots, k_{m_t}^t \leq \xi(S) - 2$$

The $\tilde{R}$*-time of* $\langle S, W \rangle$ is the supremum of

$$\inf\{t \,:\, \xi(T_t) = \xi(S) - 1\}$$

taken over all $\langle T_0, \ldots, T_{l_w} \rangle \in \tilde{\mathcal{R}}_{S,W}$.

Note that the definitions of $\tilde{\mathcal{R}}_{S,W}$ and the $\tilde{R}$-time of $\langle S, W \rangle$ are completely symmetric to that of $\mathcal{A}_{S,W}$ and the A-time of $\langle S, W \rangle$, respectively.

The next result is the analogue of Lemmata 4.6 and 4.9.

Lemma 5.5. *Let τ be the $\tilde{R}$-time of $\langle S, \mathcal{Z} \rangle$, where $S \in \mathbf{S}$ and $\mathcal{Z} = \langle Z_1, Z_2, \ldots \rangle$ is a large enough random point-array, and let $\gamma > 0$. (γ may be a function of S.) Then,*

$$\mathbb{P}\mathrm{r}[\tau \leq \gamma 2^{\xi(S)}] = 1 - \mathrm{O}(|S| e^{-2\gamma})$$

Proof. The proof is similar to the proofs of Lemmata 4.6 and 4.9. Let

$$\kappa = \xi(S)$$

For each W as in (5.3), where $l_w = |\mathcal{Z}|$ and $h_w = |Z_1|$, we define the sequence of partitions $\langle T_0^*(W), \ldots, T_{l_w}^*(W) \rangle$ by

$$T_t^*(W) = \begin{cases} S, & \text{if } t = 0 \\ \mathbb{M}_\kappa(T_{t-1}^*(W)), & \text{if } t \neq 0 \text{ and } W_{t,1} \geq 1 - \ell_\kappa(T_{t-1}^*(W)) \\ T_{t-1}^*(W), & \text{otherwise} \end{cases}$$

We also let

$$\tau^*(W) = \inf\{t \,:\, \xi(T^*_t(W)) = \kappa - 1\}$$

and $\tau(W)$ be the $\tilde{R}$-time of $\langle S, W\rangle$.

The next two results are the analogues of Claims 4.7 and 4.8, respectively. The proof for the first is very similar to that of Claim 4.7 and it is omitted.

Claim 5.6. *For all W, $\tau^*(W) \geq \tau(W)$.*

Claim 5.7. $\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq \gamma 2^\kappa] = 1 - \mathrm{O}(|S| e^{-2\gamma})$

Proof. Consider the following coupon collection process. There are $2^{\kappa-1}$ types of coupons. Initially, coupons of all but $s_\kappa(S)/2$ types have already been collected, and in each step a new coupon is chosen at random. Each random coupon is equally likely to be of any of the $2^{\kappa-1}$ types, and the random choices are independent. We are interested in the number of steps J required until coupons of all types have been collected.

Clearly, for all $i \leq |\mathcal{Z}|$, $\mathbb{Pr}[J = i] = \mathbb{Pr}[\tau^*(\mathcal{Z}) = i]$; so, since $|\mathcal{Z}| \geq \gamma 2^\kappa$,

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq \gamma 2^\kappa] = \mathbb{Pr}[J \leq \gamma 2^\kappa] \tag{5.4}$$

In each step of the above process, a coupon of a fixed type is collected with probability $1/2^{\kappa-1}$. Thus, the probability that a coupon of this type is not selected in the first $\lfloor \gamma 2^\kappa \rfloor$ steps is

$$(1 - 1/2^{\kappa-1})^{\lfloor \gamma 2^\kappa \rfloor} \leq e \cdot e^{-2\gamma}$$

Therefore, the probability that coupons of all types have been collected in the first $\lfloor \gamma 2^\kappa \rfloor$ steps is

$$\mathbb{Pr}[J \leq \gamma 2^\kappa] \geq 1 - (s_\kappa(S)/2) \cdot e \cdot e^{-2\gamma} = 1 - \mathrm{O}(|S| e^{-2\gamma})$$

This and (5.4) yields the desired result. ■ {of Claim 5.7}

Combining Claims 5.6 and 5.7, we can obtain the desired lower bound for $\mathbb{Pr}[\tau \leq \gamma 2^\kappa]$. ■

5.4.2 Proof of Lemma 5.4

As in the proof of Lemma 4.15, we identify two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, such that if both these events occur then the event we are interested in (i.e., that (i) and (ii) hold for some $\tau \leq c\xi_0 2^{\xi_0}$) also occurs; then, we compute a lower bound for the probability of $\mathcal{E}_1 \cap \mathcal{E}_2$, instead. Roughly speaking, $\mathcal{E}_1$ says that each of the first $\Theta(\xi_0 2^{\xi_0})$ addBlks operations splits a block of depth

$\leq \xi_0 - 2$, if the partition it is applied to has $\ell_{\leq \xi_0 - 2} > 1/4 + \varepsilon$; and $\mathcal{E}_2$ says that, in $\mathrm{O}(\xi_0 2^{\xi_0})$ steps, either all blocks of depth ξ_0 have been merged or some block of depth $> \xi_0 - 2$ is split.

As in the proof of Lemma 4.15, we will assume (without loss of generality) that N is larger than the upper bound for τ — large enough that the definitions we describe later on in the proof are valid. (We will implicitly make this assumption in the proofs of many subsequent results in this chapter, as well.)

We define η_i, η'_i, Z_i, and Z'_i as in the proof of Lemma 4.15; i.e., η_i is the step when the i-th $\textsc{addBlk}_{\mathbf{S}}$ operation occurs, η'_i is the step when the i-th $\textsc{rmBlk}_{\mathbf{S}}$ operation occurs, and

$$Z_i = \langle Z_{i,1}, Z_{i,2}, \ldots \rangle = Y_{\eta_i} \qquad Z'_i = \langle Z'_{i,1}, Z'_{i,2}, \ldots \rangle = Y_{\eta'_i}$$

Let

$$\kappa' = \lambda_+(\xi_0 - 1) \cdot 2^{\xi_0 - 1} \qquad \kappa = \kappa' + 2^{\xi_0 - 1}$$

Let $\mathcal{E}_1$ be the event:

$$\text{“for all } i \leq \kappa,\ \min\{Z_{i,j} \,:\, j \leq \lambda_+(\xi_0 - 1)\} \leq 1/4 + \varepsilon\text{”}$$

and $\mathcal{E}_2$ be the event:

$$\text{“}r \leq \kappa'\text{”}$$

where r is the $\tilde{R}$-time of $\langle S_0, \langle Z'_1, \ldots, Z'_{\kappa'} \rangle\rangle$. (Again, without loss of generality, we assume that $\eta_i < \infty$ for all $i \leq \kappa$, and $\eta'_i < \infty$ for all $i \leq \kappa'$; we can always achieve that by appropriately modifying the adversary for $t > \kappa + \kappa'$.) Since the Z_i are random point-vectors,

$$\mathbb{Pr}[\mathcal{E}_1] \geq 1 - \kappa \cdot (1 - \tfrac{1}{4} - \varepsilon)^{\lambda_+(\xi_0 - 1)} \geq 1 - \kappa e^{-(1/4+\varepsilon)\lambda_+(\xi_0 - 1)}$$

Also, since $\langle Z'_1, \ldots, Z'_{\kappa'} \rangle$ is a random point-array, applying Lemma 5.5 (for $\gamma = \kappa' 2^{-\xi_0}$) yields

$$\mathbb{Pr}[\mathcal{E}_2] = 1 - \mathrm{O}(|S_0| e^{-\lambda_+(\xi_0 - 1)})$$

Therefore,

$$\mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1 - \mathrm{O}(\kappa e^{-(1/4+\varepsilon)\lambda_+(\xi_0 - 1)}) = 1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4+\varepsilon)\lambda_+(\xi_0 - 1)})$$

Combining this with Claim 5.8 that we show below, yields the desired result.

Claim 5.8. *If $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs then (i) and (ii) hold for some $\tau < \kappa + \kappa'$.*

Proof. The proof is similar to that of Claim 4.16. Let

- J be the earliest step when a block of depth $> \xi_0 - 2$ is split,
- $K = \min\{\eta'_{\kappa'},\ J-1\}$, and
- M be the total number of $\textsc{addBlk}_{\mathbf{S}}$ operations performed in the first K steps.

Since the number of $\mathbb{S}$ operations required to split all blocks in S_0 of depth $\leq \xi_0 - 2$ into blocks of depth $> \xi_0 - 2$ is smaller than $2^{\xi_0 - 1}$, and the number of $\textsc{rmBlk}_{\mathbf{S}}$ operations in the first $K \leq \eta'_{\kappa'}$ steps is at most κ',

$$M < 2^{\xi_0 - 1} + \kappa' = \kappa \tag{5.5}$$

Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs. We distinguish two cases depending on which of $\eta'_{\kappa'}$ or $J-1$ is smaller.

If $K = \eta'_{\kappa'}$ then $\langle S_0, S_{\eta'_1}, \ldots, S_{\eta'_{\kappa'}} \rangle \in \tilde{\mathcal{R}}_{S, \langle Z'_1, \ldots, Z'_{\eta'_{\kappa'}} \rangle}$; so, by $\mathcal{E}_2$,

$$\xi_{\eta'_{\kappa'}} \leq \xi_0 - 1$$

If $K = J - 1 < \eta'_{\kappa'}$, instead, then $J = \eta_{M+1} \leq \eta_\kappa$, by (5.5). So, $\mathcal{E}_1$ yields $\min\{Y_{J,j} : j \leq \lambda_+(\xi_0 - 1)\} \leq 1/4 + \varepsilon$, and, by Lemma 4.5(a),

$$\ell_{\leq \xi_0 - 2}(S_{J-1}) \leq 1/4 + \varepsilon$$

(since otherwise the block split in step J would have depth $\leq \xi_0 - 2$).

Combining the above two cases, we have that

$$\xi_K \leq \xi_0 - 1 \quad \text{or} \quad \ell_{\leq \xi_0 - 2}(S_K) \leq 1/4 + \varepsilon$$

Let

$$\tau = \min\{t : \xi_t \leq \xi_0 - 1 \text{ or } \ell_{\leq \xi_0 - 2}(S_t) \leq 1/4 + \varepsilon\}$$

Since $\ell_{\leq \xi_0 - 2}(S_0) > 1/4 - \varepsilon$ and $\tau \leq K < J$, we have that either

$$\xi(S_\tau) = \xi_0 \quad \text{and} \quad \ell_{\leq \xi_0 - 2}(S_\tau) = 1/4 + \varepsilon$$

or

$$\xi(S_\tau) = \xi_0 - 1 \quad \text{and} \quad \ell_{\leq \xi_0 - 2}(S_\tau) > 1/4 + \varepsilon$$

So, if $\xi(S_\tau) = \xi_0$ then S_τ is normal-tailed, while if $\xi(S_\tau) = \xi_0 - 1$ then, by Lemma 5.2(a), S_τ is left-heavy; thus, condition (ii) holds. Since $\tau \leq K$, condition (i) is also true. Finally,

$$\tau \leq K \leq M + \kappa' < \kappa + \kappa'$$

by (5.5). ■

5.5 From a normal-tailed to a short-tailed partition

Roughly speaking, we show that in an $\mathcal{S}$-process that starts from a normal-tailed partition and has sufficiently large sampling-size functions, with high probability, it takes at most $\mathrm{O}(|S_0|)$ steps until one of the following three events happens: we reach a short-tailed partition, or we reach a fat-tailed partition, or all smallest blocks of S_0 have been merged. The formal statement of this result is as follows.

Lemma 5.9. *Consider an $\mathcal{S}$-process such that S_0 is normal-tailed and, for all $k \geq 2$,*

$$\lambda_+(k) \geq 8(\ln 2)k \quad \textit{and} \quad \lambda_-(k) \geq 8k \tag{5.6}$$

If the $\mathcal{S}$-process is long enough then, with probability

$$1 - \mathrm{O}(2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0)}) \tag{5.7}$$

there is $\tau \leq c2^{\xi_0}$, where c is a positive constant, such that

(i) all the blocks that are split in the first τ steps have depths $\leq \xi_0 - 2$, and

(ii) S_τ is short-tailed, or S_τ is fat-tailed, or S_τ is left-heavy and $\xi_\tau < \xi_0$.

The proof of this result is based on the observation that for a normal-tailed partition with $\mu \ll \xi$, (1) the number of $\textsc{addBlk}_{\mathbf{S}}$ operations required to increase μ by one is smaller than the number of $\textsc{rmBlk}_{\mathbf{S}}$ operations required to merge a single pair of blocks of depth $\leq \xi - 2$ (since $\ell_{\leq \xi-2} \leq 1/4 + \varepsilon$); and, (2) the $\textsc{addBlk}_{\mathbf{S}}$ operations split blocks of depth $\leq \xi - 2$, with high probability (since $\ell_{\leq \xi-2} \geq 1/4 - \varepsilon$). So, starting from a normal-tailed S_0, if $\textsc{addBlk}_{\mathbf{S}}$ operations occur "sufficiently often" then a short-tailed partition is reached, while if mostly $\textsc{rmBlk}_{\mathbf{S}}$ operations take place then either a fat-tailed partition is reached or all smallest blocks are merged.

We begin by revisiting the notion of A-times and showing a related result.

5.5.1 More on A-times

Let $S \in \mathbf{S}$ and W be as in (4.8), i.e.,

$$W = \langle W_1, \ldots, W_{l_w} \rangle, \qquad \text{where } W_t = \langle W_{t,1}, \ldots, W_{t,h_w} \rangle \in I^{h_w}, \text{ for each } t = 1, \ldots, l_w \tag{5.8}$$

Recall from Section 4.5 that the A-time of $\langle S, W \rangle$ is, roughly speaking, the maximum number of $\textsc{addBlk}_{\mathbf{S}}$ operations required to split all largest blocks of S, when the sample points used are those in W, and any number of pairs of sibling blocks of depth $\geq \mu(S)+2$ may be merged between two ADDBLK$_{\mathbf{S}}$ operations. We define the *A-time of* $\langle \kappa, \kappa', W \rangle$, where $\kappa, \kappa' \in \mathbb{N}$, as the supremum of the A-time of $\langle S, W \rangle$ taken over all S such that

$$\mu(S) = \kappa \quad \text{and} \quad \xi(S) = \kappa'$$

The next lemma provides an upper bounds on the A-time of $\langle \kappa, \kappa', \mathcal{Z} \rangle$, where $\mathcal{Z}$ is a random point-array, that holds with probability $\geq 1/2$.

Lemma 5.10. *Let τ be the A-time of $\langle \kappa, \kappa', \mathcal{Z} \rangle$, where $\kappa, \kappa' \in \mathbb{N}$ and $\mathcal{Z} = \langle Z_1, Z_2, \ldots \rangle$ is a large enough random point-array. Then,*

$$\mathbb{Pr}\left[\tau \leq 2^{\kappa+1} + \frac{2}{\sigma}\Big(2 + \ln\frac{2}{\sigma}\Big)\right] \geq \frac{1}{2}, \quad \textit{where } \sigma = \frac{\lambda_+(\kappa+1)}{2^\kappa}$$

Proof. The proof is similar to that of Lemma 4.6. For each W as in (5.8), where $l_w = |\mathcal{Z}|$ and $h_w = |Z_1|$, we define the sequence $\langle T_0^*(W), \ldots, T_{l_w}^*(W) \rangle$ as in (4.12), letting S be the partition that has

$$\mu(S) = \xi(S) = \kappa$$

We also let

$$\tau^*(W) = \inf\{t \,:\, \mu(T_t^*(W)) = \kappa + 1\}$$

and $\tau(W)$ be the A-time of $\langle \kappa, \kappa', W \rangle$. Then,

Claim 5.11. *For all W, $\tau^*(W) \geq \tau(W)$.*

The proof of Claim 5.11 is similar to that of Claim 4.7 and is omitted. The next result is the analogue of Claim 4.8. Notice, however, that the bound we show below holds even for small values of κ, while the bound in Claim 4.8 is asymptotic, as $\kappa \to \infty$.

Claim 5.12. $\mathbb{Pr}\left[\tau^*(\mathcal{Z}) \leq 2^{\kappa+1} + \frac{2}{\sigma}\big(2 + \ln\frac{2}{\sigma}\big)\right] \geq \frac{1}{2}$

Proof. Let

$$b = 2^{\kappa+1} + \frac{2}{\sigma}\Big(2 + \ln\frac{2}{\sigma}\Big)$$

Similarly to (4.25), we have that

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq b] \geq \mathbb{Pr}\left[\sum_{i=1}^{2^\kappa} \tau_i \leq b\right] \tag{5.9}$$

for $\tau_1, \ldots, \tau_{2^\kappa}$ independent geometric random variables such that, for each i, $\mathbb{E}[\tau_i] = 1/p_i$, where

$$p_i = 1 - (1 - i/2^\kappa)^{\lambda_+(\kappa+1)} \geq 1 - e^{-i\sigma}$$

By applying Markov's inequality to the right-hand side of (5.9) we obtain

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq b] \geq 1 - \frac{1}{b}\,\mathbb{E}\Big[\sum_{i=1}^{2^\kappa} \tau_i\Big] \tag{5.10}$$

We now bound from above the expected value of $\sum_{i=1}^{2^\kappa} \tau_i$.

$$\mathbb{E}\Big[\sum_{i=1}^{2^\kappa} \tau_i\Big] = \sum_{i=1}^{2^\kappa} \frac{1}{p_i} \leq \sum_{i=1}^{2^\kappa} \frac{1}{1-e^{-i\sigma}} = 2^\kappa + \sum_{i=1}^{2^\kappa} \frac{e^{-i\sigma}}{1-e^{-i\sigma}}$$

Since

$$\sum_{i=2}^{2^\kappa} \frac{e^{-i\sigma}}{1-e^{-i\sigma}} \leq \int_1^\infty \frac{e^{-x\sigma}}{1-e^{-x\sigma}}dx = \Big[\frac{1}{\sigma}\ln(1-e^{-x\sigma})\Big]_1^\infty = \frac{1}{\sigma}\ln\frac{1}{1-e^{-\sigma}}$$

we have

$$\mathbb{E}\Big[\sum_{i=1}^{2^\kappa} \tau_i\Big] \leq 2^\kappa + \frac{e^{-\sigma}}{1-e^{-\sigma}} + \frac{1}{\sigma}\ln\frac{1}{1-e^{-\sigma}} \tag{5.11}$$

We distinguish two cases depending on the value of σ.
If $\sigma \geq 1$ then

$$\ln\frac{1}{1-e^{-\sigma}} = \ln\Big(1+\frac{e^{-\sigma}}{1-e^{-\sigma}}\Big) \leq \frac{e^{-\sigma}}{1-e^{-\sigma}}$$

and

$$\frac{e^{-\sigma}}{1-e^{-\sigma}} \leq \frac{e^{-1}}{1-e^{-1}} \leq \frac{2}{e} \leq \frac{2}{1+\sigma}$$

Therefore, (5.11) yields

$$\mathbb{E}\Big[\sum_{i=1}^{2^\kappa} \tau_i\Big] \leq 2^\kappa + \frac{e^{-\sigma}}{1-e^{-\sigma}}\Big(1+\frac{1}{\sigma}\Big) \leq 2^\kappa + \frac{2}{1+\sigma}\Big(1+\frac{1}{\sigma}\Big) = 2^\kappa + \frac{2}{\sigma} < \frac{b}{2}$$

If $\sigma < 1$, instead, then

$$e^{-\sigma} \leq 1 - \sigma + \frac{\sigma^2}{2} \leq 1 - \frac{\sigma}{2}$$

so, (5.11) yields

$$\mathbb{E}\Big[\sum_{i=1}^{2^\kappa} \tau_i\Big] \leq 2^\kappa + \frac{1-\sigma/2}{\sigma/2} + \frac{1}{\sigma}\ln\frac{2}{\sigma} = 2^\kappa + \frac{1}{\sigma}(2+\ln\frac{2}{\sigma}) - 1 < \frac{b}{2}$$

Therefore, in both cases $\mathbb{E}[\sum_{i=1}^{2^\kappa} \tau_i] < b/2$. Combining this and (5.10) yields the desired bound for $\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq b]$. ■ {of Claim 5.12}

The lemma now follows by combining Claims 5.11 and 5.12. ■

5.5.2 Proof of Lemma 5.9

We describe three events, $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, such that if they all occur then (i) and (ii) hold for some $\tau \leq c2^{\xi_0}$, and we show that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ occurs with the probability in (5.7). Roughly speaking, $\mathcal{E}_1$ says that each of the first $\Theta(2^{\xi_0})$ $\textsc{addBlk}_{\mathbf{S}}$ operations splits a block of depth $\leq \xi_0 - 2$, if the partition it is applied to has $\ell_{\leq \xi_0 - 2} \geq 1/4 - 2\varepsilon$; for each depth $d \leq \xi_0 - 2\xi_0$, $\mathcal{E}_2$ states a threshold for the maximum number of $\textsc{rmBlk}_{\mathbf{S}}$ operations, among the first $\Theta(2^{\xi_0})$ such operations, that are applied to partitions of $\mu \geq d$ and $\ell_{\leq \xi_0 - 2} \leq 1/4 + 2\varepsilon$, and merge blocks of depth $\leq \xi_0 - 2$; and, for each d, $\mathcal{E}_3$ states a threshold for the maximum number of $\textsc{addBlk}_{\mathbf{S}}$ operations that are applied to partitions of $\mu = d$, provided that the thresholds specified by $\mathcal{E}_2$ are not exceeded.

For $i \geq 1$, we define η_i, η'_i, Z_i, and Z'_i as in the proof of Lemma 5.4. Also, for $d \geq 1$, we let η_i^d be the i-th step when an $\textsc{addBlk}_{\mathbf{S}}$ operation is applied to a partition of $\mu = d$. Formally,

$$\eta_i^d = \inf\{t \,:\, |\{j \leq t \,:\, V_j = + \text{ and } \mu_{j-1} = d\}| = i\}$$

If $\eta_i^d < \infty$, we denote by Z_i^d the corresponding sequence of sample points, i.e.,

$$Z_i^d = Y_{\eta_i^d}$$

(Without loss of generality, we assume that the adversary is such that $\eta_i^d < \infty$ for all d, i such that the definition of the events $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ below are valid.)

We now define the events $\mathcal{E}_1$ $\mathcal{E}_2$, and $\mathcal{E}_3$. Let

$$\kappa = \varepsilon 2^{\xi_0 - 2} \qquad \kappa' = \kappa + 2^{\xi_0}$$

$\mathcal{E}_1$ is the event:

$$\text{“for all } i \leq \kappa,\ \min\{Z_{i,j} \,:\, j \leq \lambda_+(\xi_0 - 1)\} < 1/4 - 2\varepsilon\text{”}$$

Define

$$Q_i'^d = \begin{cases} 1, & \text{if } \max\{Z'_{i,j} \,:\, j \leq \lambda_-(d)\} < 1/4 + 2\varepsilon \\ 0, & \text{otherwise} \end{cases}$$

and let $\mathcal{E}_{2,d}$ be the event:

$$\text{“}\sum_{i=1}^{\kappa'} Q_i'^d \leq \kappa'_d\text{”}$$

The values of the κ'_d will be specified later. Define also

$$Q_i^d = \begin{cases} 1, & \text{if } a_{d,i} \leq \zeta_d \\ 0, & \text{otherwise} \end{cases}$$

where $a_{d,i}$ is the A-time of

$$\left\langle d, \xi_0, \langle Z^d_{(i-1)\zeta_d+1}, Z^d_{(i-1)\zeta_d+2}, \ldots, Z^d_{i\zeta_d}\rangle\right\rangle$$

and let $\mathcal{E}_{3,d}$ be the event:

$$\text{“}\sum_{i=1}^{\kappa_d} Q^d_i \geq \kappa'_d + 1\text{”}$$

The values of the κ_d and ζ_d will also be determined later. We define the events $\mathcal{E}_2$ and $\mathcal{E}_3$ as

$$\mathcal{E}_2 = \bigcap_{d=2}^{\nu-1} \mathcal{E}_{2,d} \quad \text{and} \quad \mathcal{E}_3 = \bigcap_{d=2}^{\nu-1} \mathcal{E}_{3,d}$$

where

$$\nu = \xi_0 - \lfloor 2\log \xi_0 \rfloor$$

Next, we compute the probability of the above events. For $\mathcal{E}_1$, since the Z_i are random point-vectors, we have

$$\mathbb{Pr}[\mathcal{E}_1] \geq 1 - \kappa(1 - 1/4 + 2\varepsilon)^{\lambda_+(\xi_0-1)} \geq 1 - \kappa e^{-(1/4-2\varepsilon)\lambda_+(\xi_0-1)} \tag{5.12}$$

To establish lower bounds for $\mathbb{Pr}[\mathcal{E}_2]$ and $\mathbb{Pr}[\mathcal{E}_3]$ we will use the following version of the Chernoff's bounds. By $\mathrm{Bi}(n,p)$ we denote a binomial random variable with parameters n and p.

Lemma 5.13.

(a) $\mathbb{Pr}[\mathrm{Bi}(n,p) > enp] < e^{-np}$

(b) $\mathbb{Pr}[\mathrm{Bi}(n,p) < np/2] < e^{-np/8}$

Part (a) of the lemma follows from Theorem 4.1 in [60], and part (b) from Theorem 4.2 in [60].

First we bound $\mathbb{Pr}[\mathcal{E}_{2,d}]$. Note that, for each d, the $Q'^d_1, \ldots, Q'^d_{\kappa'_d}$ are independent, since the Z'_i are independent random point-vectors. Also, for each i,

$$\mathbb{Pr}[Q'^d_i = 1] = (1/4 + 2\varepsilon)^{\lambda_-(d)} \leq e^{-(3/4-2\varepsilon)\lambda_-(d)}$$

Thus,

$$\mathbb{Pr}[\mathcal{E}_{2,d}] = \mathbb{Pr}\left[\mathrm{Bi}(\kappa', \mathbb{Pr}[Q'^d_i = 1]) \leq \kappa'_d\right] \geq \mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \kappa'_d] \tag{5.13}$$

where

$$p'_d = e^{-(3/4-2\varepsilon)\lambda_-(d)}$$

(The second relation in (5.13) holds because $\mathrm{Bi}(n,p)$ is stochastically smaller than $\mathrm{Bi}(n,p')$, if $p' \geq p$). We set

$$\kappa'_d = \lfloor \max\{e\kappa' p'_d,\, \xi_0 \log \xi_0\} \rfloor$$

If $e\kappa' p'_d \geq \xi_0 \log \xi_0$ then

$$\mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \kappa'_d] = \mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq e\kappa' p'_d]$$

so, by Lemma 5.13(a),

$$\mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \kappa'_d] > 1 - e^{-\kappa' p'_d} \geq 1 - e^{-\xi_0 \log \xi_0 / e}$$

If $e\kappa' p'_d < \xi_0 \log \xi_0$, instead, then

$$\mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \kappa'_d] = \mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \xi_0 \log \xi_0] \geq \mathbb{Pr}[\mathrm{Bi}(\kappa', \xi_0 \log \xi_0/(e\kappa')) \leq \xi_0 \log \xi_0]$$

so, again, by Lemma 5.13(a),

$$\mathbb{Pr}[\mathrm{Bi}(\kappa', p'_d) \leq \kappa'_d] > 1 - e^{-\xi_0 \log \xi_0 / e}$$

Applying the above results to (5.13), we obtain

$$\mathbb{Pr}[\mathcal{E}_{2,d}] \geq 1 - e^{-\xi_0 \log \xi_0 / e}$$

thus,

$$\mathbb{Pr}[\mathcal{E}_2] \geq 1 - \nu e^{-\xi_0 \log \xi_0 / e} \tag{5.14}$$

Next, we bound $\mathbb{Pr}[\mathcal{E}_{3,d}]$. We set

$$\zeta_d = \left\lceil 2^{d+1} + \frac{2^{d+1}}{\lambda_+(d+1)}\left(2 + \ln \frac{2^{d+1}}{\lambda_+(d+1)}\right)\right\rceil$$

Then, by Lemma 5.10,

$$\mathbb{Pr}[Q_i^d = 1] = \mathbb{Pr}[a_{d,i} \leq \zeta_d] \geq 1/2$$

Note that, for each d, $Q_1^d, \ldots, Q_{\kappa_d}^d$ are independent because their values are determined based on *non-overlapping* parts of $\langle Z_1^d, Z_2^d, \ldots\rangle$. So,

$$\mathbb{Pr}[\mathcal{E}_{3,d}] = \mathbb{Pr}\left[\mathrm{Bi}(\kappa_d,\, \mathbb{Pr}[Q_i^d = 1]) \geq \kappa'_d + 1\right] \geq \mathbb{Pr}[\mathrm{Bi}(\kappa_d, 1/2) \geq \kappa'_d + 1] \tag{5.15}$$

We now set

$$\kappa_d = 4(\kappa'_d + 1)$$

Then, by Lemma 5.13(b),

$$\mathbb{Pr}[\mathrm{Bi}(\kappa_d, 1/2) \geq \kappa'_d + 1] > 1 - e^{-(\kappa'_d+1)/4} = 1 - \mathrm{O}(e^{-\xi_0 \log \xi_0/4})$$

From this and (5.15),

$$\mathbb{Pr}[\mathcal{E}_{3,d}] = 1 - \mathrm{O}(e^{-\xi_0 \log \xi_0/4})$$

thus,

$$\mathbb{Pr}[\mathcal{E}_3] = 1 - \mathrm{O}(\nu e^{-\xi_0 \log \xi_0/4}) \tag{5.16}$$

Combining (5.12), (5.14), and (5.16), we obtain

$$\mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3] = 1 - \mathrm{O}(2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0-1)})$$

To establish Lemma 5.9, it remains to show the following result.

Claim 5.14. *If event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ occurs then (i) and (ii) hold, for some $\tau \leq \kappa + \kappa'$.*

Proof. Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ occurs. Since S_0 is normal-tailed, $\ell_{\leq \xi_0-2}(S_0) \geq 1/4 - \varepsilon$, and, so, for all $t \leq \eta_\kappa$,

$$\ell_{\leq \xi_0-2}(S_t) \geq \ell_{\leq \xi_0-2}(S_0) - \kappa 2^{\xi_0-2} = 1/4 - 2\varepsilon$$

From that, $\mathcal{E}_1$, and Lemma 4.5(a), we obtain that the first κ addBlks operations split blocks of depth $\leq \xi_0 - 2$. Therefore, (i) holds for $\tau = \eta_\kappa$, and, thus, for any $\tau \leq \eta_\kappa$. Note that the number of rmBlks operations during the first η_κ steps is at most $|S_0| + \kappa \leq 2^{\xi_0} + \kappa = \kappa'$; thus,

$$\eta_\kappa \leq \kappa + \kappa'$$

Therefore, to complete the proof of the claim it suffices to show that (ii) holds for some $\tau \leq \eta_\kappa$. We distinguish two cases.

Case A: $\ell_{\leq \xi_0-2}(S_t) > 1/4 + 2\varepsilon$, for some $t \leq \eta_\kappa$.
Let τ be the smallest such t. Clearly, $\xi_\tau \in \{\xi_0, \xi_0 - 1\}$. If $\xi_\tau = \xi_0$ then S_τ is fat-tailed; if $\xi_\tau = \xi_0 - 1$ then, by Lemma 5.2(a), S_τ is left-heavy. Therefore, (ii) holds.

Case B: $\ell_{\leq \xi_0-2}(S_t) \leq 1/4 + 2\varepsilon$, for all $t \leq \eta_\kappa$.
We show that the number of addBlks operations that are applied to partitions of $\mu < \nu$ in the first η_κ steps is *strictly* smaller than κ. So, since (by definition) exactly κ addBlks operations take place during the first η_κ steps, there is some $\tau < \eta_\kappa$ such that $\mu_\tau \geq \nu$ and, thus, S_τ is short-tailed. Let K'_d, for $d \geq 1$, be the number of rmBlks operations that are

applied to partitions of $\mu \geq d$ and merge blocks of depth $\leq \xi_0 - 2$, during the first η_κ steps. By the case hypothesis, $\mathcal{E}_2$, and Lemma 4.5(b), we have that for each $d \in [2..\nu - 1]$,

$$K'_d \leq \kappa'_d \tag{5.17}$$

Let K_d be the number of $\textsc{addBlk}_\mathbf{S}$ operations that are applied to partitions of $\mu = d$ in the first η_κ steps. Let also, for $i \geq 1$,

$$\Delta_i = [\eta^d_{(i-1)\zeta_d} + 1..\eta^d_{i\zeta_d}]$$

(We define $\eta^d_0 = 0$.) Note that exactly ζ_d $\textsc{addBlk}_\mathbf{S}$ operations are applied to partitions of $\mu = d$ during the steps in Δ_i. If no blocks of depth d or $d+1$ are merged during these steps, and $Q^d_i = 1$ then all blocks of depth $\leq d$ have been split by the last step in Δ_i; i.e., $\mu_t > d$, for $t = \eta^d_{i\zeta_d}$. Also if $\mu_{t_1} > d$ and $\mu_{t_2} = d$, for some $t_1 < t_2$, then, between steps $t_1 + 1$ and t_2, at least one $\textsc{rmBlk}_\mathbf{S}$ operation is applied to a partition of $\mu = d + 1$ and merges a pair of blocks of depth $d + 1$. By these observations, for each $d \in [2..\nu - 1]$, there are at most $K'_d + 1$ distinct i such that $\Delta_i \cap [1..\eta_\kappa] \neq \emptyset$ and $Q^d_i = 1$. However, by $\mathcal{E}_{3,d}$ and (5.17), $Q^d_i = 1$ for at least $\kappa'_d + 1 \geq K'_d + 1$ among the $i = 1, \ldots, \kappa_d$. Therefore,

$$K_d \leq \kappa_d \zeta_d$$

The total number of $\textsc{addBlk}_\mathbf{S}$ operations that are applied to partitions of $\mu < \nu$ during the first η_κ steps is then

$$\sum_{d=2}^{\nu-1} K_d \leq \sum_{d=2}^{\nu-1} (\kappa_d \zeta_d) = \sum_{d=2}^{\nu-1} (4(\kappa'_d + 1)\zeta_d) \tag{5.18}$$

By the definition of κ'_d and the assumption that, for all $k \geq 2$, $\lambda_-(k) \geq 8k$, we have

$$\kappa'_d \leq \xi_0 \log \xi_0 + e\kappa' p'_d = \xi_0 \log \xi_0 + e(1 + 4/\varepsilon)\kappa e^{-5\lambda_-(d)/8} \leq \xi_0 \log \xi_0 + 65e\kappa e^{-5d}$$

Also, by the definition of ζ_d and the assumption that, for all $k \geq 2$, $\lambda_+(k) \geq 8(\ln 2)k$,

$$\zeta_d \leq 2^{d+1}(1 + 1/4)$$

Applying the above two results to (5.18), yields

$$\begin{aligned}\sum_{d=2}^{\nu-1} K_d &\leq \sum_{d=2}^{\nu-1} \big(4(\xi_0 \log \xi_0 + 65e\kappa e^{-5d} + 1) \cdot 2^{d+1}(1 + 1/4)\big) \\ &= 10(\xi_0 \log \xi_0 + 1) \sum_{d=2}^{\nu-1} 2^d + 10 \cdot 65e\kappa \sum_{d=2}^{\nu-1} (2^d e^{-5d}) \\ &\leq 10(\xi_0 \log \xi_0 + 1)2^{\xi_0+1}/\xi_0^2 + 650e\kappa(2e^{-5})^2/(1 - 2e^{-5})\end{aligned}$$

In the last line, the first term is in $\Theta(2^{\xi_0} \log \xi_0 / \xi_0)$, and the second is $< \kappa/2$. So, for all sufficiently large ξ_0,

$$\sum_{d=2}^{\nu-1} K_d < \kappa$$

Since the number of $\textsc{addBlk}_{\mathbf{S}}$ operations applied to partitions of $\mu < \nu$ in the first η_κ steps is smaller that the total number κ of $\textsc{addBlk}_{\mathbf{S}}$ operations during these steps, there is some $\tau < \eta_\kappa$ such that $\mu_\tau \geq \nu$. This, together with the case hypothesis and the fact that $\xi_\tau \leq \xi_0$, yields that S_τ is short-tailed; thus, condition (ii) holds. ■

5.6 From an arbitrary to an almost-safe partition

Here we combine the results of the previous two sections to show that, roughly speaking, in an $\mathcal{S}$-process that starts from any initial partition and has sufficiently large sampling-size functions, it takes at most $\mathrm{O}(\xi_0 2^{\xi_0})$ steps, with high probability, until an almost-safe partition is reached. The formal statement of this result is as follows.

Lemma 5.15. *Consider an $\mathcal{S}$-process such that, for all $k \geq 0$,*

$$\lambda_+(k) \geq \max\{8(\ln 2)k,\, \beta\} \quad \textit{and} \quad \lambda_-(k) \geq \max\{8k, \beta\} \tag{5.19}$$

where β is a sufficiently large constant. If the $\mathcal{S}$-process is long enough then, with probability

$$1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0 - 2\log \xi_0)}) \tag{5.20}$$

there is $\tau \leq c\xi_0 2^{\xi_0}$, where c is a positive constant, such that

(i) S_τ is almost-safe, and

(ii) for all $t \leq \tau$,

$$\xi_t \leq \begin{cases} \xi_0, & \textit{if } S_0 \textit{ is left-heavy} \\ \xi_0 + 1, & \textit{otherwise} \end{cases}$$

We prove this result in two steps. First, using Lemmata 5.4 and 5.9, we show that if we start from a partition that is not thin-tailed then, with high probability, it takes at most $\mathrm{O}(\xi_0 2^{\xi_0})$ steps until we reach either a short-tailed partition or a partition that has $\xi < \xi_0$. This result is formally stated as follows.

Lemma 5.16. *For any long enough $\mathcal{S}$-process such that S_0 is either thick-tailed or normal-tailed, and* (5.6) *holds for all $k \geq 2$, with probability*

$$1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0)})$$

there is $\tau \leq c\xi_0 2^{\xi_0}$, where c is a positive constant, such that

(i) all the blocks that are split in the first τ steps have depths $\leq \xi_0 - 2$, and

(ii) S_τ is short-tailed, or S_τ is left-heavy and $\xi_\tau < \xi_0$.

We then use this result to establish Lemma 5.15. The main machinery that we use in the proofs of both lemmata (and that we have not used in any of the proofs we have described so far) is that, roughly speaking, we formulate the transitions between partitions of different types in the $\mathcal{S}$-process as a Markov-chain, and use this Markov-chain to establish bounds on the number of steps required until some partition of a desired type is reached.

Proof of Lemma 5.16

We assume that S_0 is *not* short-tailed — otherwise, the lemma holds trivially (for $\tau = 0$).

Informally, the proof proceeds as follows. We consider the sequence of times $\langle \tau_0 = 0, \tau_1, \tau_2, \ldots \rangle$, where τ_{i+1} is the earliest step after step τ_i when a normal-tailed partition is reached, if S_{τ_i} is thick-tailed; or a thick-tailed partition is reached,[4] if S_{τ_i} is normal-tailed. We focus on the prefix of this sequence until one of following two conditions is met: (a) we reach a short-tailed partition or a partition with $\xi = \xi_0 - 1$; or (b) a block of depth $> \xi_0 - 2$ is split. (Intuitively, (a) is the "good" outcome, and (b) the "bad.") Using Lemma 5.4 (Lemma 5.9) we bound from below the probability that from a normal-tailed (thick tailed) partition in step τ_i, we reach a thick-tailed (normal-tailed) partition in step τ_{i+1}, or condition (a) is met. Based on that, we compute a lower bound on the probability of the event that either condition (a) is met in $\mathrm{O}(\xi_0 2^{\xi_0})$ steps, or a larger number of steps take place without condition (b) being met (event $\mathcal{E}_1$). We also compute a lower bound on the probability of the event that a partition of $\xi = \xi_0 - 1$ is reached before that large number of steps takes place (event $\mathcal{E}_2$). Combining these two bounds we obtain the desired result.

We begin by introducing some useful notation. Let

$$\begin{aligned}
\mathbf{S}_{\text{nor}} &= \{S \in \mathbf{S} : S \text{ is normal-tailed but not short-tailed, and } \xi(S) = \xi_0\} \\
\mathbf{S}_{\text{thk}} &= \{S \in \mathbf{S} : S \text{ is thick-tailed but not short-tailed, and } \xi(S) = \xi_0\} \\
\mathbf{S}_{\text{fat}} &= \{S \in \mathbf{S} : S \text{ is fat-tailed and } \xi(S) = \xi_0\} \\
\mathbf{T} &= \{S \in \mathbf{S} : S \text{ is short-tailed and } \xi(S) \leq \xi_0, \text{ or } S \text{ is left-heavy and } \xi(S) < \xi_0\}
\end{aligned}$$

Note that $S_0 \in \mathbf{S}_{\text{thk}} \cup \mathbf{S}_{\text{nor}}$. Note also that if (i) holds then (ii) is equivalent to $S_\tau \in \mathbf{T}$.

[4] More precisely, a fat-tailed one.

We define the following infinite sequence of times $\langle \tau_0, \tau_1, \ldots \rangle$. We let $\tau_0 = 0$, and for each $i \geq 1$, we let τ_i be as follows, where c_0 is a positive constant we will determine later; c_1 is equal to the constant c of Lemma 5.4; and c_2 is equal to the constant c of Lemma 5.9.

- If $\tau_{i-1} = \infty$ or $S_{\tau_{i-1}} \in \mathbf{T}$ or $i > c_0\xi_0$ then $\tau_i = \infty$.
- Else if $S_{\tau_{i-1}} \in \mathbf{S}_{\text{thk}}$ then τ_i is the infimum of all $t \in [\tau_{i-1} + 1..\tau_{i-1} + c_1\xi_0 2^{\xi_0}]$ such that:
  - $S_t \in \mathbf{S}_{\text{nor}} \cup \mathbf{T}$, and
  - all the blocks that are split in steps $\tau_{i-1} + 1$ to t have depths $\leq \xi_0 - 2$.
- Else if $S_{\tau_{i-1}} \in \mathbf{S}_{\text{nor}}$ then τ_i is the infimum of all $t \in [\tau_{i-1} + 1..\tau_{i-1} + c_2 2^{\xi_0}]$ such that:
  - $S_t \in \mathbf{S}_{\text{fat}} \cup \mathbf{T}$, and
  - all the blocks that are split in steps $\tau_{i-1} + 1$ to t have depths $\leq \xi_0 - 2$.

Clearly, for the above sequence of τ_i, there is some index

$$K \leq c_0\xi_0$$

such that:

- for all $1 \leq i \leq K$, $S_{\tau_i} \in \mathbf{S}_{\text{nor}} \cup \mathbf{S}_{\text{fat}}$,
- for all $i \geq K + 2$, $\tau_i = \infty$, and
- either $\tau_{K+1} = \infty$, or $\tau_{K+1} < \infty$ and $S_{\tau_{K+1}} \in \mathbf{T}$.

Specifically, for all $i \in [1..K]$, if $S_0 \in \mathbf{S}_{\text{thk}}$ then for all even such i, $S_{\tau_i} \in \mathbf{S}_{\text{fat}}$, and for all odd i, $S_{\tau_i} \in \mathbf{S}_{\text{nor}}$; if $S_0 \in \mathbf{S}_{\text{nor}}$ the reverse is true — i.e., for all odd i, $S_{\tau_i} \in \mathbf{S}_{\text{fat}}$, and for all even i, $S_{\tau_i} \in \mathbf{S}_{\text{nor}}$. (Recall that we assumed $S_0 \in \mathbf{S}_{\text{thk}} \cup \mathbf{S}_{\text{nor}}$.) To distinguish between the two possible cases for $i = K + 1$, we define

$$Q = \begin{cases} 1, & \text{if } \tau_{K+1} < \infty \text{ (and, thus, } S_{\tau_{K+1}} \in \mathbf{T}) \\ 0, & \text{if } \tau_{K+1} = \infty \end{cases}$$

Note that all the blocks that are split in the first τ_{K+Q} steps have depths $\leq \xi_0 - 2$, and for all $i \in [1..K + Q]$,

$$\tau_i - \tau_{i-1} \leq \begin{cases} c_1\xi_0 2^{\xi_0}, & \text{if } S_{\tau_{i-1}} \in \mathbf{S}_{\text{thk}} \\ c_2 2^{\xi_0}, & \text{if } S_{\tau_{i-1}} \in \mathbf{S}_{\text{nor}} \end{cases} \tag{5.21}$$

Next, we describe two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, which imply the event that we are interested in (i.e., that (i) and (ii) hold for some $\tau \leq c\xi_0 2^{\xi_0}$). $\mathcal{E}_1$ is the event:

$$\text{“}Q = 1 \text{ or } K = \lfloor c_0\xi_0 \rfloor\text{”}$$

We can compute an upper bound for the probability that the complementary event: "$Q = 0$ and $K < \lfloor c_0 \xi_0 \rfloor$" occurs, as follows.

$$\mathbb{P}\mathrm{r}[\bar{\mathcal{E}}_1] = \mathbb{P}\mathrm{r}[\{Q = 0\} \cap \{K < \lfloor c_0 \xi_0 \rfloor\}] = \sum_{0 \le i < \lfloor c_0 \xi_0 \rfloor} \mathbb{P}\mathrm{r}[\{Q = 0\} \cap \{K = i\}]$$
$$\le \sum_{0 \le i < \lfloor c_0 \xi_0 \rfloor} \mathbb{P}\mathrm{r}[\{Q = 0\} \cap \{K = i\} \mid K \ge i] \tag{5.22}$$

Let $\mathcal{A}$ be the event:

$$\{\tau_i = t\} \cap \{S_t = T\} \cap \{K \ge i\}$$

where $i, t \in \mathbb{N}$ and $T \in \mathbf{S}$ are such that $\mathbb{P}\mathrm{r}[\mathcal{A}] > 0$. Conditioned on $\mathcal{A}$, $\langle S_{\tau_i}, S_{\tau_i + 1}, \ldots \rangle$ is the partition-sequence of an $\mathcal{S}$-process (of initial partition T, the same sampling-size functions and precision as the original $\mathcal{S}$-process, and length $N - t$). Also, since $\mathbb{P}\mathrm{r}[\mathcal{A}] > 0$, $T \in \mathbf{S}_{\text{thk}} \cup \mathbf{S}_{\text{nor}}$. So, if $T \in \mathbf{S}_{\text{thk}}$ then, by Lemma 5.4,

$$\mathbb{P}\mathrm{r}[\{Q = 0\} \cap \{K = i\} \mid \mathcal{A}] = \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4 + \varepsilon)\lambda_+(\xi_0)})$$

while if $T \in \mathbf{S}_{\text{nor}}$ then, by Lemma 5.9,

$$\mathbb{P}\mathrm{r}[\{Q = 0\} \cap \{K = i\} \mid \mathcal{A}] = \mathrm{O}(2^{\xi_0} e^{-(1/4 - 2\varepsilon)\lambda_+(\xi_0)})$$

Applying the above two results to (5.22), we obtain

$$\mathbb{P}\mathrm{r}[\bar{\mathcal{E}}_1] = \mathrm{O}(c_0 \xi_0^2 2^{\xi_0} e^{-(1/4 + \varepsilon)\lambda_+(\xi_0)} + c_0 \xi_0 2^{\xi_0} e^{-(1/4 - 2\varepsilon)\lambda_+(\xi_0)}) = \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4 - 2\varepsilon)\lambda_+(\xi_0)})$$

thus,

$$\mathbb{P}\mathrm{r}[\mathcal{E}_1] = 1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4 - 2\varepsilon)\lambda_+(\xi_0)}) \tag{5.23}$$

We now describe $\mathcal{E}_2$. For $i \ge 1$, we define η'_i and Z'_i as in the proof of Lemma 5.4. $\mathcal{E}_2$ is the event:

$$\text{“}r \le \kappa'\text{”}$$

where r is the $\tilde{R}$-time of $\langle S, \langle Z'_1, \ldots, Z'_{\kappa'} \rangle \rangle$, and

$$\kappa' = \lambda_+(\xi_0 - 1) \cdot 2^{\xi_0 - 1}$$

Since the Z'_i are independent random point-vectors, we can bound from below the probability that $\mathcal{E}_2$ occurs by applying Lemma 5.5 (for $\gamma = \kappa' 2^{-\xi_0}$):

$$\mathbb{P}\mathrm{r}[\mathcal{E}_2] = 1 - \mathrm{O}(|S_0| e^{-\lambda_+(\xi_0 - 1)}) \tag{5.24}$$

By (5.23) and (5.24),

$$\mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1 - \mathrm{O}\big(\xi_0 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0)}\big)$$

Combining this with Claim 5.17 that we show next, yields the desired result.

Claim 5.17. *If $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs then (i) and (ii) hold for some $\tau \leq c\xi_0 2^{\xi_0}$, where c is a constant > 0.*

Proof. Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs. We will show that

$$Q = 1 \quad \text{and} \quad \tau_{K+1} < c\xi_0 2^{\xi_0}$$

for some constant $c > 0$. From this, it is then immediate that (i) and (ii) hold for $\tau = \tau_{K+1} < c\xi_0 2^{\xi_0}$.

By $\mathcal{E}_2$, there is some step

$$\tau' \leq 2\kappa' + 2^{\xi_0 - 1} = 2^{\xi_0-1}\big(2\lambda_+(\xi_0-1)+1\big)$$

such that

$$\xi_{\tau'} < \xi_0 \quad \text{or} \quad \text{a block of depth} \geq \xi_0 - 1 \text{ is split in step } \tau'$$

(Because if all the blocks that are split in the first $2\kappa' + 2^{\xi_0-1}$ steps have depths $\leq \xi_0 - 2$ then at least κ' rmBlks operations are performed in these steps, and, thus, $\langle S_0, S_{\eta'_1}, \ldots, S_{\eta'_{\kappa'}}\rangle \in \tilde{\mathcal{R}}_{S,\langle Z'_1,\ldots,Z'_{\kappa'}\rangle}$. So, by $\mathcal{E}_2$, $\xi_{\tau'} < \xi_0$, for $\tau = \eta'_{\kappa'} \leq 2\kappa' + 2^{\xi_0-1}$.) Clearly, $\tau_K < \tau'$, therefore,

$$\tau_K < 2^{\xi_0-1}\big(2\lambda_+(\xi_0-1)+1\big) \tag{5.25}$$

We can now show that $Q = 1$ as follows. For all $i \in [2..K]$,

$$\tau_i - \tau_{i-1} \geq \varepsilon 2^{\xi_0-2}$$

since $\varepsilon 2^{\xi_0-2}$ is the minimum number of $\mathbb{S}$ operations required to get from a fat-tailed partition of $\xi = \xi_0$ to a normal-tailed partition of the same ξ (and, thus, it is also the minimum number of $\mathbb{M}$ operations to achieve the reverse result). Therefore,

$$\tau_K \geq (K-1)\varepsilon 2^{\xi_0-2}$$

Combining this and (5.25) yields

$$K < (2/\varepsilon)\cdot\big(2\lambda_+(\xi_0-1)+1\big)+1$$

Therefore,

$$K < \lfloor c_0 \xi_0 \rfloor$$

if c_0 is a sufficiently large constant (since $\lambda_+(k) = \Theta(k)$). From this and $\mathcal{E}_1$, we have that $Q = 1$. It remains to show that $\tau_{K+1} < c\xi_0 2^{\xi_0}$. Since $Q = 1$, by (5.21),

$$\tau_{K+1} \leq \tau_\kappa + \max\{c_1 \xi_0 2^{\xi_0},\, c_2 2^{\xi_0}\}$$

so, by (5.25),

$$\tau_{K+1} < 2^{\xi_0 - 1}\big(2\lambda_+(\xi_0 - 1) + 1\big) + \max\{c_1 \xi_0 2^{\xi_0},\, c_2 2^{\xi_0}\} \leq c\xi_0 2^{\xi_0}$$

for a sufficiently large constant c. ■

Proof of Lemma 5.15

Informally, the proof proceeds as follows. As in the proof of Lemma 5.16, we define a sequence of times $\langle \tau_0 = 0, \tau_1, \tau_2, \ldots \rangle$, such that we can apply Lemma 5.16 to each of the sequences of partitions $\langle S_{\tau_i}, S_{\tau_i + 1}, \ldots, S_{\tau_{i+1}} \rangle$. (In the proof of Lemma 5.16, the corresponding sequence of times was such that Lemma 5.4 or 5.9 could be applied to each such sequence of partitions.) Here, however, there is the complication that Lemma 5.16 can only be used when the starting partition S_{τ_i} is thick-tailed or normal-tailed (like Lemmata 5.4 and 5.9) but (unlike them) it does not ensure, with high probability, that the resulting $S_{\tau_{i+1}}$ will be thick-tailed or normal-tailed (or short-tailed). We overcome this problem as follows. If S_{τ_i} is thin-tailed then, instead of $\langle S_{\tau_i}, S_{\tau_i + 1}, \ldots \rangle$, we consider a (less balanced) partition sequence (denoted $\langle T_0^{i+1}, T_1^{i+1}, \ldots \rangle$), that is obtained if we replace S_{τ_i} by a thick-tailed or normal-tailed partition, denoted $h(S_{\tau_i})$, and then apply to it the same sequence of operations as in the original $\mathcal{S}$-process. $h(S_{\tau_i})$ is such that $h(S_{\tau_i}) \preceq S_{\tau_i}$ and $\xi(h(S_{\tau_i})) = \xi_{\tau_i}$ or $\xi_{\tau_i} + 1$. Roughly speaking, τ_{i+1} is the earliest step after step τ_i when (a) an almost-safe partition is reached, or (b) all blocks of depth $d = \xi_{\tau_i}$ (or $d = \xi(h(S_{\tau_i}))$, if S_{τ_i} is thin-tailed) have been merged, or (c) a block of depth d or $d-1$ is split. Outcomes (a) and (b) are the "good" ones, while (c) is the "bad" one. (We denote by G_i the indicator random variable of the good outcomes.) Next we express the event whose probability we want to bound in Lemma 5.15, in terms of an event (denoted $\mathcal{E}$) on the sequence of τ_i. Then we show that this new event is implied by two simpler events, $\mathcal{E}_1$ and $\mathcal{E}_2$. Intuitively, $\mathcal{E}_1$ says that all partition sequences that start from a sufficiently large partition have a good outcome ($G_i = 1$); and $\mathcal{E}_2$ says that most of the partition sequences that start from a smaller partition also have a good outcome. We then show that the probability that $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs is as in (5.20), which implies the desired result.

We begin with some definitions. We let h be a function that maps each partition S to a partition $h(S) \preceq S$ that is thick-tailed or normal-tailed, and has the smallest possible ξ. Formally, let $h : \mathbf{S} \to \mathbf{S}$ such that, for each $S \in \mathbf{S}$,

- if $\xi(S) < 6$ then $h(S) = S$,
- if $\xi(S) \geq 6$ then $h(S)$ is thick-tailed or normal-tailed, $h(S) \preceq S$, and

$$\xi(h(S)) = \begin{cases} \xi(S), & \text{if } S \text{ is left-heavy} \\ \xi(S) + 1, & \text{otherwise} \end{cases}$$

Note that, for the case where $\xi(S) \geq 6$, if S is left-heavy then, by the definition of a left-heavy partition, there is $h(S)$ with the required properties; if S is not left-heavy then $h(S)$ exists because of Lemma 5.2(b). If $\xi(S)$ very small, it is possible that there is no partition $\preceq S$ that is thick-tailed or normal-tailed; this is the reason why we treat partitions of $\xi < 6$ as a special case in the definition of h. An important observation for our analysis is that if $\xi(S) < 6$ and S is thin-tailed, then S is almost-safe.

We denote by $\mathcal{T}(S, t)$, for $t \in [0..N]$ and $S \in \mathbf{S}$, the sequence of partitions we obtain if we start from S and apply to it the same sequence of operations that are applied to partitions $S_t, S_{t+1}, \ldots$ in the $\mathcal{S}$-process. Formally, $\mathcal{T}(S, t) = \langle T_0, \ldots, T_{N-t} \rangle$, where

$$T_j = \begin{cases} S, & \text{if } j = 0 \\ \textsc{addBlk}_{\mathbf{S}}(T_{j-1}, Y_{t+j}), & \text{if } j \neq 1 \text{ and } V_{t+j} = + \\ \textsc{rmBlk}_{\mathbf{S}}(T_{j-1}, Y_{t+j}), & \text{if } j \neq 1 \text{ and } V_{t+j} = - \text{ and } |T_{j-1}| > 1 \\ T_{j-1}, & \text{if } j \neq 1 \text{ and } V_{t+j} = - \text{ and } |T_{j-1}| = 1 \end{cases}$$

The last case in the above definition ensures that we never attempt to apply a $\textsc{rmBlk}_{\mathbf{S}}$ operation to a partition of size 1; it is used only when $|S| < |S_t|$. A useful property of $\mathcal{T}(S, t)$, which follows from Lemma 3.14, is that if $S \preceq S_t$ then, for all j,

$$T_j \preceq S_{t+j}$$

Consider now the following finite sequence of times $\langle \tau_0, \ldots, \tau_\kappa \rangle$, for some sufficiently large κ (such that the arguments we make later on apply.) We let $\tau_0 = 0$, and, for $i \geq 1$, we define

$$\tau_i = \tau_{i-1} + \delta_i,$$

where δ_i is the smallest positive integer such that at least one of the conditions (a)–(d) below apply. For $j \geq 1$, let

$$\langle T_0^j, T_1^j, \ldots \rangle = \mathcal{T}(h(S_{\tau_{j-1}}),\, \tau_{j-1})$$

(a) $T_{\delta_i}^i$ is almost-safe.

(b) $T_{\delta_i}^i$ is left-heavy and $\xi(T_{\delta_i}^i) < \xi(T_0^i)$.

(c) A block of depth $\geq \xi(T_0^i) - 1$ is split in step δ_i of $\langle T_0^i, T_1^i, \ldots \rangle$.

(d) $\delta_i = \phi(\xi(T_0^i))$, for some function $\phi : \mathbb{N} \to \mathbb{N}$ that will be specified later.

Note that the sequence of the τ_i is determined from the sequence of partitions $\langle S_0, T_1^1, \ldots, T_{\delta_1}^1, T_1^2, \ldots, T_{\delta_2}^2, T_1^3, \ldots \rangle$, instead of directly from the partition-sequence of the $\mathcal{S}$-process — as in the proof of Lemma 5.16.

It is straightforward to show that the following facts hold, for all $1 \leq i \leq \kappa$:

(1) For all $j \leq \delta_i$, $T_j^i \preceq S_{\tau_{i-1}+j}$

(2) $1 \leq \delta_i \leq \phi(\xi(T_0^i))$

(3) For all $j < \delta_i$, $\xi(T_j^i) \leq \xi(T_0^i)$

(4) $\xi(h(T_{\delta_i}^i)) \leq \begin{cases} \xi(T_0^i) - 1, & \text{if condition (b) holds} \\ \xi(T_0^i) + 1, & \text{otherwise} \end{cases}$

(5) $\xi(T_{\delta_i}^i) \leq \xi(T_0^i)$, if condition (a) holds

(6) $\xi(T_0^{i+1}) \leq \xi(h(T_{\delta_i}^i))$

We denote by $\mathcal{E}$ the event that an almost-safe partition is reached (in the modified sequence of partitions) in $\mathrm{O}(\xi_0 2^{\xi_0})$ steps, and all intermediate partitions have $\xi \leq \xi_0$ (more correctly, $\xi \leq \xi(h(S_0))$). More formally, let

$$K = \inf\{i \,:\, \text{condition (a) holds}\}$$

$\mathcal{E}$ is the event:

$$\big\{K < \infty\big\} \cap \big\{\tau_K \leq c\xi_0 2^{\xi_0}\big\} \cap \big\{\xi(T_{\delta_K}^K) \leq \xi(h(S_0))\big\} \cap \bigcap_{i=1}^{K-1} \big\{\xi(h(T_{\delta_i}^i)) \leq \xi(h(S_0))\big\}$$

where c is a constant we will specify later.

We now argue that if $\mathcal{E}$ occurs then conditions (i) and (ii) of Lemma 5.15 hold, for some $\tau \leq c\xi_0 2^{\xi_0}$. Suppose that $\mathcal{E}$ occurs. Then, by (3) and (6), for all $i \leq K$ and $j \leq \delta_i$,

$$\xi(T_j^i) \leq \xi(h(S_0))$$

(The case $(i,j) = (K, \delta_K)$ is explicit in the definition of $\mathcal{E}$.) So, by (1) and Lemma 3.4, (ii) holds, for $\tau = \tau_K$. Also, by (1) and the fact that if $S \succeq S'$ and S' is almost-safe then so is S, we have that (i) holds for $\tau = \tau_K$, as well. Finally, by the definition of $\mathcal{E}$, $\tau_K \leq c\xi_0 2^{\xi_0}$. Consequently, to prove the lemma it suffices to show that $\mathbb{Pr}[\mathcal{E}]$ is equal to the probability in (5.20); we show this in the rest of the proof.

We describe two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, such that their intersection implies $\mathcal{E}$. For $1 \leq i \leq \kappa$, let

$$G_i = \begin{cases} 1, & \text{if (a) or (b) holds} \\ 0, & \text{otherwise} \end{cases}$$

Let also

$$A_1 \text{ be the set of the } \xi_0^2 \text{ smallest } i \text{ such that } \xi(T_0^i) \geq \xi_0 - 2\log\xi_0$$

and

$$A_2 \text{ be the set of the } \xi_0^2 \text{ smallest } i \text{ such that } \xi(T_0^i) < \xi_0 - 2\log\xi_0$$

(Without loss of generality, we assume that there are always enough i of each type to populate both A_1 and A_2.) $\mathcal{E}_1$ is the event:

$$\text{“for all } i \in A_1,\ G_i = 1$$

and $\mathcal{E}_2$ is the event:

$$\text{“}\sum_{i \in A_2} G_i \geq \frac{3}{4}\xi_0^2\text{”}$$

We now compute the probability of the above two events. Let $\mathcal{A}$ be the event:

$$\{\tau_{i-1} = t\} \cap \{S_t = S\} \cap \mathcal{H}$$

where $i, t \in \mathbb{N}$, $S \in \mathbf{S}$, and $\mathcal{H}$ is an event on the first t steps of the $\mathcal{S}$-process such that $\mathbb{Pr}[\mathcal{A}] > 0$. Then, conditioned on $\mathcal{A}$, the sequence $\langle T_0^i, T_1^i, \ldots \rangle$ is the partition-sequence of some $\mathcal{S}$-process (of the same sampling-size functions as the original $\mathcal{S}$-process). So, if we set

$$\phi(j) = \lceil c' j 2^j \rceil \tag{5.26}$$

where c' is the constant c of Lemma 5.16, then, by Lemma 5.16,

$$\mathbb{Pr}[G_i = 1 \mid \mathcal{A}] = 1 - \mathrm{O}(d2^d e^{-(1/4-2\varepsilon)\lambda_+(d)}) \tag{5.27}$$

where $d = \xi(S)$. From this and the fact that, if $d \geq \xi_0 - 2\log\xi_0$ then

$$\mathrm{O}(d2^d e^{-(1/4-2\varepsilon)\lambda_+(d)}) = \mathrm{O}\big((2^{\xi_0}/\xi_0) \cdot e^{-(1/4-2\varepsilon)\lambda_+(\xi_0 - 2\log\xi_0)}\big)$$

we obtain that

$$\mathbb{Pr}[\mathcal{E}_1] = 1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0 - 2\log\xi_0)}) \tag{5.28}$$

By (5.27), we also have that, for all $\mathcal{A}$ as above such that $d = \xi(S) \geq d_0$, for a large enough constant d_0,

$$\mathbb{Pr}[G_i = 1 \mid \mathcal{A}] \geq 7/8$$

We can make the above relation hold for $d < d_0$, as well, by choosing the constant β in (5.19) to be sufficiently large. From that, using Chernoff's bound (Theorem 4.2 in [60]), we can show that

$$\mathbb{Pr}[\mathcal{E}_2] \geq 1 - e^{-\xi_0^2/(2\cdot 7\cdot 8)}$$

Combining this and (5.28), yields

$$\mathbb{Pr}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1 - \mathrm{O}(\xi_0 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda_+(\xi_0 - 2\log\xi_0)})$$

To establish Lemma 5.15, it remains to show the following result.

Claim 5.18. *If $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs then $\mathcal{E}$ occurs.*

Proof. Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs. Let

$$J = \min\{\xi_0^2, K\}$$

For each $j \in \{0, 1\}$, we define

$$a_j = |\{i \leq J \,:\, \xi(T_0^i) \geq \xi_0 - 2\log\xi_0 \text{ and } G_i = j\}|$$
$$b_j = |\{i \leq J \,:\, \xi(T_0^i) < \xi_0 - 2\log\xi_0 \text{ and } G_i = j\}|$$

Clearly,

$$a_0 + a_1 + b_0 + b_1 = J \tag{5.29}$$

By (4) and (6), we have that for all $1 \leq i < K$,

$$\xi(T_0^{i+1}) \leq \begin{cases} \xi(T_0^i) - 1, & \text{if } G_i = 1 \\ \xi(T_0^i) + 1, & \text{otherwise} \end{cases} \tag{5.30}$$

From this it follows that

$$a_1 + b_1 \leq \xi(h(S_0)) + (a_0 + b_0) \tag{5.31}$$

Since $J \leq \xi_0^2$, we have, by $\mathcal{E}_1$, that $a_0 = 0$, and, by $\mathcal{E}_2$, that $b_0 \leq (1/4)\xi_0^2$. Combining these two results with (5.29) and (5.31), yields

$$J \leq \xi(h(S_0)) + 2(a_0 + b_0) \leq \xi(h(S_0)) + (1/2)\xi_0^2 < \xi_0^2$$

and, thus,

$$K < \xi_0^2 < \infty$$

Since $a_0 = 0$, we have, by (5.30), that for each d such that $\xi_0 - 2\log\xi_0 + 2 \leq d \leq \xi(h(S_0))$, there is at most one $i \leq K$ such that $\xi(T_0^i) = d$; so, by (2) and (5.26),

$$\tau_K \leq \sum_{j=\xi_0-\lfloor 2\log\xi_0\rfloor+2}^{\xi_0+1} \phi(d) + \xi_0^2 \cdot \phi(\xi_0 - 2\log\xi_0 + 1) \leq c\xi_0 2^{\xi_0}$$

for c sufficiently large. Since $a_0 = 0$, by applying (4) and (6) inductively, we obtain that, for all $i < K$,

$$\xi(h(T_{\delta_i}^i)) \leq \xi(h(S_0))$$

Finally, by (5), (6), and the above inequality, we have

$$\xi(T_{\delta_K}^K) \leq \xi(h(S_0))$$

Therefore, $\mathcal{E}$ occurs. ■

5.7 From an almost-safe to a safe partition

Informally, in this section we show that in an $\mathcal{S}$-process that starts from an almost-safe partition and has sufficiently large sampling-size functions, it takes at most $\mathrm{O}(|S_0|)$ steps, with high probability, until a safe partition is reached. The formal statement of this result is as follows.

Lemma 5.19. *For any long enough $\mathcal{S}$-process such that S_0 is almost-safe, with probability*

$$1 - \mathrm{O}(\log\xi_0 \cdot 2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\xi_0 - 2\log\xi_0)})$$

there is $\tau \leq c2^{\xi_0}$, where c is a positive constant, such that

(i) S_τ is safe

(ii) for all $t \leq \tau$, $\xi_t \leq \xi_0 + 1$.

The proof is similar to that of Lemma 5.16, and it is based on two results we describe in Sections 5.7.1 and 5.7.2. The first result is analogous to Theorem 4.1, but it concerns short-tailed partitions instead of safe ones. The second is a generalization of the upper bound for A-times we showed in Section 4.5.

5.7.1 From a short-tailed to a short-tailed partition

In this section, we consider a generalization of the class of short-tailed partitions, that consists of all $S \in \mathbf{S}$ such that

$$S \text{ is not fat-tailed} \quad \text{and} \quad \mu \geq \xi - \epsilon(\xi)$$

where $\epsilon(k) = \mathrm{o}(k)$. (Recall that the original definition of an almost-safe partition requires that $\mu \geq \xi - 2\log\xi$ instead of $\mu \geq \xi - \epsilon(\xi)$.) We show that if an $\mathcal{S}$-process starts from such a partition and has large enough sampling-size functions then, with high probability, in $\Theta(|S_0|)$ steps, it reaches another non fat-tailed partition that has $\xi \leq \xi_0 + 1$, and no blocks of depth $\leq \xi_0 - 2$ are merged during these steps — so, the "tail" of the initial partition does not get bigger. The formal statement of this result is as follows.

Lemma 5.20. *For any long enough $\mathcal{S}$-process such that S_0 is not fat-tailed and $\mu_0 \geq \xi_0 - \mathrm{o}(\xi_0)$, with probability*

$$1 - \mathrm{O}(2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

there is $\tau \in [c_1 2^{\xi_0} .. c_2 2^{\xi_0}]$, where c_1, c_2 are positive constants, such that

(i) S_τ is not fat-tailed

(ii) for all $t \leq \tau$,

- *if $\frac{1}{4} \leq \ell_{\xi_0}(S_0) \leq \frac{3}{4}$ then $\xi_t = \xi_0$ and $\frac{1}{4} - \varepsilon \leq \ell_\xi(S_t) \leq \frac{3}{4} + \varepsilon$*
- *if $\ell_{\xi_0}(S_0) > \frac{3}{4}$ then $\xi_t \in \{\xi_0, \xi_0 + 1\}$ and $\ell_{\xi_0+1}(S_t) \leq \frac{1}{4} + \varepsilon$ and $\ell_{\xi_0}(S_t) \geq \frac{3}{4} - \varepsilon$*
- *if $\ell_{\xi_0}(S_0) < \frac{1}{4}$ then $\xi_t \in \{\xi_0 - 1, \xi_0\}$ and $\ell_{\xi_0}(S_t) \leq \frac{1}{4} + \varepsilon$ and $\ell_{\xi_0-1}(S_t) \geq \frac{1}{4} - \varepsilon$; in particular, if no blocks of depth $\geq \xi_0 - 1$ are split in the first τ steps then $\xi_\tau = \xi_0 - 1$.*

(iii) for all $t \leq \tau$ and $d \in \mathbb{N}$,

- *if $V_t = +$ and $\ell_{\leq d}(S_{t-1}) \geq \frac{1}{4} - \varepsilon$ then the block split in step t has depth $\leq d$*
- *if $V_t = -$ and $\ell_{\geq d}(S_{t-1}) \geq \frac{1}{4} - \varepsilon$ then the blocks merged in step t have depth $\geq d$.*

The above result can be viewed as the analogue of Theorem 4.1 for the generalized class of short-tailed partitions. Note that Lemma 5.20 seems to provide more guarantees than Theorem 4.1 does. Even though it is not explicitly stated in the statement of Theorem 4.1, in its proof we show that similar guarantees apply. In fact, it is immediate from the proof of Theorem 4.1 that Lemma 5.20 holds when S_0 is safe. The proof of Lemma 5.20 is very similar to that of Theorem 4.1 (see Section 4.6). Intuitively, the tail of S_0 does not essentially affect the analysis because: (1) since the tail is not fat, S_0 is similar to a safe partition; and (2) since the tail is short, the number of random probes executed for each $\textsc{addBlk}_{\mathbf{S}}$ or $\textsc{rmBlk}_{\mathbf{S}}$ operation may be smaller that $\lambda_{+/-}(\log|S_t|)$, respectively, by only a negligible factor — of order $\mathrm{o}(1)$. The details of the proof are omitted.

5.7.2 A_k-times

Let $S \in \mathbf{S}$ and W be as in (4.8), i.e.,

$$W = \langle W_1, \dots, W_{l_w} \rangle, \qquad \text{where } W_t = \langle W_{t,1}, \dots, W_{t,h_w} \rangle \in I^{h_w}, \text{ for each } t = 1, \dots, l_w \tag{5.32}$$

Recall that the A-time of $\langle S, W \rangle$ is, roughly speaking, the maximum number of $\textsc{addBlk}_{\mathbf{S}}$ operations required to split all blocks of S of depth (at most) $k = \mu(S)$, when the sample points used are those in W, and any number of sibling blocks of depth $\geq k+2$ may be merged between $\textsc{addBlk}_{\mathbf{S}}$ operations. In this section, we consider a generalization of the above definition, where k is a parameter, and we prove a result analogous to that we showed in Section 4.5 for A-times.

We denote by $\mathcal{A}_{k,S,W}$, for $k \geq \mu(S)$, the set of all sequences in $\mathcal{A}_{S,W}$ such that no blocks of depth $\leq k+1$ are merged between $\textsc{addBlk}_{\mathbf{S}}$ operations. The *A_k-time of $\langle S, W \rangle$* is the supremum of

$$\inf\{t : \mu(T_t) > k\}$$

taken over all $\langle T_0, \dots, T_{l_w} \rangle \in \mathcal{A}_{k,S,W}$.

The next result is the analogue of Lemma 4.6. By $\mathrm{sp}_k(S)$ we denote the number of $\mathbb{S}$ operations we must apply to partition S to achieve $\mu > k$. It is straightforward to verify that

$$\mathrm{sp}_k(S) = \sum_{j \leq k} \left(s_j(S) \cdot (2^{k+1-j} - 1) \right) \tag{5.33}$$

Lemma 5.21. *Let τ be the A_k-time of $\langle S, \mathcal{Z} \rangle$, where $S \in \mathbf{S}$, $k \geq \mu(S)$, and $\mathcal{Z} = \langle Z_1, Z_2, \dots \rangle$ is a large enough random point-array, and let γ be a positive constant. Then,*

$$\mathbb{Pr}[\tau \leq \mathrm{sp}_k(S) + \gamma 2^{k+1}] = 1 - \mathrm{O}\big(2^k e^{-\gamma(1-\mathrm{O}(1/\ln k))\lambda_+(k+1)}\big)$$

Proof. (Similar to the proof of Lemma 4.6.) Let T be the partition such that

$$\mu(T) = k+1, \quad \xi(T) = k+2, \quad \text{and} \quad s_{k+1}(T) = \mathrm{sp}_k(S)$$

(By (5.33), $\mathrm{sp}_k(S) < 2^{k+1}$, so, T is well defined.) For each W as in (5.32), where $l_w = |\mathcal{Z}|$ and $h_w = |Z_1|$, we define the sequence of partitions $\langle T^*_0(W), \ldots, T^*_{l_w}(W)\rangle$ as

$$T^*_t(W) = \begin{cases} T, & \text{if } t = 0 \\ \mathbb{S}_{\mu(T)}(T^*_{t-1}(W)), & \text{if } t \neq 0 \text{ and } \min\{W_{t,i} \,:\, i \leq \lambda_+(\mu(T))\} < \ell_{\mu(T)}(T^*_{t-1}(W)) \\ T^*_{t-1}(W), & \text{otherwise} \end{cases} \tag{5.34}$$

Note that the above definition is almost identical to (4.12); it differs from that only in the starting partition, and in the condition of the middle case, where "$\lambda_+(\mu(T))$" is used instead of "$\lambda_+(\mu(T)+1)$." We let

$$\tau^*(W) = \inf\{t \,:\, \mu(T^*_t(W)) = \mu(T) + 1\}$$

and $\tau(W)$ be the A_k-time of $\langle S, W\rangle$. Then,

Claim 5.22. *For all W, $\tau^*(W) \geq \tau(W)$.*

Proof. (Similar to the proof of Claim 4.7.) Suppose for contradiction that $\tau^*(W) < \tau(W)$, for some W. Then, for this W, there is a sequence $\langle T_0, \ldots, T_{l_w}\rangle \in \mathcal{A}_{k,S,W}$ such that

$$\tau^*(W) < \inf\{t \,:\, \mu(T_t) > k\}$$

Below we write T^*_t to denote $T^*_t(W)$, for the above value of W. Let

$$t_0 = \min\{t \,:\, s_{k+1}(T^*_{t_0}) < \mathrm{sp}_k(T_{t_0})\}$$

(Clearly, there is such a t_0.) Then,

$$s_{k+1}(T^*_{t_0-1}) = \mathrm{sp}_k(T_{t_0-1}) \tag{5.35}$$

$$s_{k+1}(T^*_{t_0}) = s_{k+1}(T^*_{t_0-1}) - 1 \tag{5.36}$$

$$\mathrm{sp}_k(T_{t_0}) = \mathrm{sp}_k(T_{t_0-1}) \tag{5.37}$$

By (5.35) and (5.33),

$$\ell_{k+1}(T^*_{t_0-1}) = \sum_{j\leq k} \big(s_j(T_{t_0-1}) \cdot (2^{k+1-j} - 1)\big)/2^{k+1} < \sum_{j\leq k} \big(s_j(T_{t_0-1})/2^j\big) = \ell_{\leq k}(T_{t_0-1}) \tag{5.38}$$

The rest of the proof is analogous to the corresponding part of the proof of Claim 4.7, using relations (5.38), (5.36), and (5.37) in place of (4.13), (4.14) and (4.15), respectively.

■ {of Claim 5.22}

We remarked earlier that definition (5.34) is *almost* identical to (4.12). In fact, the two definitions become *exactly* the same if we make the following two changes to (4.12): we set $S = T$, and we let the underlying sampling-size function be $\lambda'_+(d) = \lambda_+(d-1)$ (instead of $\lambda_+(d)$). Claim 4.8 then yields

$$\mathbb{Pr}[\tau^*(\mathcal{Z}) \leq \mathrm{sp}_k(S) + \gamma 2^{k+1}] \geq 1 - \mathrm{O}\big(2^k e^{-\gamma(1-\mathrm{O}(1/\ln k))\lambda_+(k+1)}\big)$$

Combining this with Claim 5.22, yields the desired result. ■

5.7.3 Proof of Lemma 5.19

In the proof we describe below we assume that S_0 is short-tailed and that it is not safe. If S_0 is safe then the lemma holds trivially (for $\tau = 0$). If S_0 is not short-tailed then we consider the partition-sequence $\langle S'_0, \ldots, S'_N\rangle$, instead of $\langle S_0, \ldots, S_N\rangle$, where S'_0 is a short-tailed partition such that

$$S'_0 \preceq S_0 \quad \text{and} \quad \xi(S'_0) = \xi_0$$

and, for $t \geq 1$,

$$S'_t = \begin{cases} \textsc{addBlk}_{\mathbf{S}}(S'_{t-1}, Y_t), & \text{if } V_t = + \\ \textsc{rmBlk}_{\mathbf{S}}(S'_{t-1}, Y_t), & \text{if } V_t = - \end{cases}$$

(By Lemma 5.3, S'_0 exists.)[5] The corresponding result for the original partition-sequence can then be obtained by observing that (by Lemma 3.14) $S'_t \preceq S_t$, for all t, and applying Lemmata 4.2 and 3.4.

The proof has a similar structure as the proof of Lemma 5.16; here we use Lemma 5.20 in place of Lemmata 5.4 and 5.9, and Lemma 5.21 in place of Lemma 5.5. Roughly speaking, we consider a sequence of times $\langle \tau_0 = 0, \tau_1, \tau_2, \ldots\rangle$, such that we can apply Lemma 5.20 to each of the sequences $\langle S_{\tau_i}, S_{\tau_i+1}, \ldots, S_{\tau_{i+1}}\rangle$. Intuitively, τ_{i+1} is the earliest step $\tau_i + \tau$ such that $\tau \in \Theta(|S_{\tau_i}|)$ and τ satisfies conditions (i)–(iii) of Lemma 5.20; if no such a step exists then the sequence of τ_i stops (more correctly, $\tau_j = \infty$ for all $j > i$). Based on the sequence of τ_i, we describe two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, such that if they both occur then the event we are interested in also occurs; so, we compute a lower bound for the probability of $\mathcal{E}_1 \cap \mathcal{E}_2$, instead. Roughly, $\mathcal{E}_1$ says that unless a safe partition is reached, the length of the sequence of τ_i exceeds some $\Theta(\xi_0)$ threshold. $\mathcal{E}_2$ describes, for each depth k, a threshold on the maximum number of $\textsc{rmBlk}_{\mathbf{S}}$ operations required until all blocks of depth $< k$ have been split, provided that no blocks of depth $\leq k$ are merged during those steps; the thresholds are

[5] Note that $\langle S'_0, \ldots, S'_N\rangle$ is in fact the sequence $\mathcal{T}(0, S'_0)$, where $\mathcal{T}$ was defined in the proof of Lemma 5.15.

chosen based on Lemma 5.21. The intersection of $\mathcal{E}_1$ and $\mathcal{E}_2$ yields that $\textsc{addBlk}_{\mathbf{S}}$ operations increase μ_t much faster than they increase ξ_t, while $\textsc{rmBlk}_{\mathbf{S}}$ operations may only decrease ξ_t, and they do not affect smaller blocks. Therefore, a safe partition is finally reached.

More precisely, let

$$\mathbf{T} = \left\{ S \in \mathbf{S} \,:\, S \text{ is safe or } \xi(S) \geq \xi_0 + 2 \text{ or } \mu(S) < \mu_0 \right\}$$

We define the following infinite sequence of times $\langle \tau_0, \tau_1, \ldots \rangle$. We let $\tau_0 = 0$, and for each $i \geq 1$, we let τ_i be as follows, where c_0 is a positive constant we will determine later; and c_1, c_2 are the constants of Lemma 5.20.

- If $\tau_{i-1} = \infty$ or $S_{\tau_{i-1}} \in \mathbf{T}$ or $i > c_0 \log \xi_0$ then $\tau_i = \infty$.
- Otherwise, τ_i is the infimum of all $t \in [\tau_{i-1} + c_1 2^{\xi_{\tau_{i-1}}} .. \tau_{i-1} + c_2 2^{\xi_{\tau_{i-1}}}]$ such that (recall conditions (i)–(iii) of Lemma 5.20):

  (a) S_t is not fat-tailed.

  (b) For all $j \in [\tau_{i-1}..t]$,

  - if $\frac{1}{4} \leq \ell_\xi(S_{\tau_{i-1}}) \leq \frac{3}{4}$ then $\xi_j = \xi_{\tau_{i-1}}$ and $\frac{1}{4} - \varepsilon \leq \ell_\xi(S_j) \leq \frac{3}{4} + \varepsilon$
  - if $\ell_\xi(S_{\tau_{i-1}}) > \frac{3}{4}$ then $\xi_j \in \{\xi_{\tau_{i-1}}, \xi_{\tau_{i-1}} + 1\}$ and $\ell_{\xi_{\tau_{i-1}}+1}(S_j) \leq \frac{1}{4} + \varepsilon$ and $\ell_{\xi_{\tau_{i-1}}}(S_j) \geq \frac{3}{4} - \varepsilon$
  - if $\ell_\xi(S_{\tau_{i-1}}) < \frac{1}{4}$ then $\xi_j \in \{\xi_{\tau_{i-1}} - 1, \xi_{\tau_{i-1}}\}$ and $\ell_{\xi_{\tau_{i-1}}}(S_j) \leq \frac{1}{4} + \varepsilon$ and $\ell_{\xi_{\tau_{i-1}}-1}(S_j) \geq \frac{1}{4} - \varepsilon$; also, if no blocks of depth $\geq \xi_0 - 1$ are split in steps $\tau_{i-1} + 1..t$ then $\xi_\tau = \xi_0 - 1$.

  (c) For all $j \in [\tau_{i-1} + 1..t]$ and $d \in \mathbb{N}$,

  - if $V_j = +$ and $\ell_{\leq d}(S_{j-1}) > \frac{1}{4} - \varepsilon$ then the block split in step j has depth $\leq d$
  - if $V_j = -$ and $\ell_{\geq d}(S_{j-1}) \geq \frac{1}{4} - \varepsilon$ then the blocks merged in step j have depth $\geq d$.

Similarly to the sequence of τ_i we described in the proof of Lemma 5.16, there is some

$$K \leq c_0 \log \xi_0$$

such that: for all $0 \leq i \leq K$, S_{τ_i} is a non fat-tailed partition and $S_{\tau_i} \notin \mathbf{T}$; for all $i \geq K + 2$, $\tau_i = \infty$; and either $\tau_{K+1} = \infty$, or $\tau_{K+1} < \infty$ and $S_{\tau_{K+1}} \in \mathbf{T}$. Again we define

$$Q = \begin{cases} 1, & \text{if } \tau_{K+1} < \infty \text{ (and, thus, } S_{\tau_{K+1}} \in \mathbf{T}) \\ 0, & \text{if } \tau_{K+1} = \infty \end{cases}$$

Next we describe two events, $\mathcal{E}_1$ and $\mathcal{E}_2$, (similar to those described in the proof of Lemma 5.16) and compute the probability that they occur. We then show that their intersection implies the event we are interested in. $\mathcal{E}_1$ is the event:

$$\text{“}Q = 1 \text{ or } K = \lfloor c_0 \log \xi_0 \rfloor\text{”}$$

Similarly to (5.22),

$$\mathbb{Pr}[\bar{\mathcal{E}}_1] \leq \sum_{0 \leq i < \lfloor c_0 \log \xi_0 \rfloor} \mathbb{Pr}[\{Q = 0\} \cap \{K = i\} \mid K \geq i] \tag{5.39}$$

Let $\mathcal{A}$ be the event:

$$\{\tau_i = t\} \cap \{S_t = T\} \cap \{K \geq i\}$$

where $i, t \in \mathbb{N}$ and $T \in \mathbf{S}$ are such that $\mathbb{Pr}[\mathcal{A}] > 0$. Conditioned on $\mathcal{A}$, $\langle S_{\tau_i}, S_{\tau_i+1}, \ldots \rangle$ is the partition-sequence of an $\mathcal{S}$-process (of the same sampling-size functions as the original $\mathcal{S}$-process). Also, since $\mathbb{Pr}[\mathcal{A}] > 0$, T is not fat-tailed, $\mu(T) \geq \mu_0$, and $\xi(T) \leq \xi_0 + 1$. So, by Lemma 5.20,

$$\mathbb{Pr}[\{Q = 0\} \cap \{K = i\} \mid \mathcal{A}] = \mathrm{O}(2^{\xi(T)} e^{-(1/4-\varepsilon)\lambda(\mu(T))}) = \mathrm{O}(2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

Applying the above to (5.39), we get

$$\mathbb{Pr}[\bar{\mathcal{E}}_1] = \mathrm{O}(\log \xi_0 \cdot 2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

so,

$$\mathbb{Pr}[\mathcal{E}_1] = 1 - \mathrm{O}(\log \xi_0 \cdot 2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)}) \tag{5.40}$$

We now describe $\mathcal{E}_2$. For $i \geq 1$, we define η_i and Z_i as in the proof of Lemma 5.4. For $k \geq \mu_0$, we let $\mathcal{E}_{2,k}$ be the event:

$$\text{“}a_k \leq \kappa_k\text{”}$$

where a_k is the A_k-time of $\langle S_0, \langle Z_1, \ldots, Z_{\kappa_k} \rangle \rangle$ and

$$\kappa_k = \mathrm{sp}_k(S_0) + 2^{k-1}$$

We let

$$\mathcal{E}_2 = \bigcap_{k=\mu_0}^{\xi_0 - 2} \mathcal{E}_{2,k}$$

By Lemma 5.21 (applied for $\gamma = 1/4$), we obtain

$$\mathbb{Pr}[\mathcal{E}_{2,k}] = 1 - \mathrm{O}\big(2^k e^{-(1-\mathrm{O}(1/\ln k))\lambda_+(k+1)/4}\big)$$

so,

$$\mathbb{P}\mathrm{r}[\mathcal{E}_2] = 1 - \mathrm{O}\big((\xi_0 - \mu_0) \cdot 2^{\mu_0} e^{-(1-\mathrm{O}(1/\ln\mu_0))\lambda_+(\mu_0+1)/4}\big) \tag{5.41}$$

By (5.40) and (5.41), we have

$$\mathbb{P}\mathrm{r}[\mathcal{E}_1 \cap \mathcal{E}_2] = 1 - \mathrm{O}(\log\xi_0 \cdot 2^{\xi_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

Combining this with Claim 5.23 that we show next, yields the desired result.

Claim 5.23. *If $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs then (i) and (ii) hold for some $\tau \leq c2^{\xi_0}$, where c is a constant > 0.*

Proof Sketch. Informally, we proceed as follows. For $j \geq \mu_0$, let C_j be the step when the κ_j-th ADDBLKS operation occurs, and, for $j \leq \xi_0$, D_j be the earliest step when $\xi_t = j$ and $\ell_j(S_t) < 1/4 - \varepsilon$. By $\mathcal{E}_2$, $\xi_{C_j} > j$ if no blocks of depth $< j+2$ have been merged until step C_j; and, by $\mathcal{E}_1$, no blocks of depth $< j$ have been merged until step D_j. We let L be the largest depth j such that $C_j \leq D_{j+2}$, and we consider the latest step G among C_L and D_{L+3}. Then, $\mu_G > L$ and $\xi_G \leq L+3$. We argue that a safe partition is reached either before step G, or soon afterwards.

For $j \geq 1$, let

$$C_j = \begin{cases} \eta_{\kappa_j}, & \text{if } j \geq \mu_0 \\ 0, & \text{otherwise} \end{cases}$$

and

$$D_j = \inf\{t : \xi_t \leq j \text{ and } \ell_j(S_t) < 1/4 - \varepsilon\}$$

Note that

$$C_1 = \cdots = C_{\mu_0-1} = 0 < C_{\mu_0} \leq C_{\mu_0+1} \leq \cdots$$
$$D_1 \geq \cdots \geq D_{\xi_0} \geq 0 = D_{\xi_0+1} = D_{\xi_0+2} = \cdots$$

($C_j = C_{j+1}$, for some $j \geq \mu_0$, iff $C_j = \infty$; and $D_j = D_{j-1}$, for some $1 < j \leq \xi_0$, iff $D_j = \infty$.) Let

$$L = \max\{j : C_j \leq D_{j+2}\}$$

Note that $\mu_0 - 1 \leq L \leq \xi_0 - 2$. Let

$$G = \max\{C_L, D_{L+3}\}$$

Then,

$$G < C_{L+1} \quad \text{and} \quad G \leq D_{L+2} \tag{5.42}$$

Let also

$$F = \inf\{t \,:\, S_t \text{ is safe}\}$$

and J be such that

$$\tau_{J-1} < \min\{G, F\} \leq \tau_J, \quad \text{if } G > 0$$

and $J = 0$, if $G = 0$.

Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2$ occurs. Based on (5.42) and the fact that, for $1 \leq i \leq K + Q$, $\tau_i - \tau_{i-1} \geq c_1 2^{\xi_{\tau_{i-1}}}$, we can show (inductively) that

$$J - 1 \leq \frac{2}{c_1}(\xi_0 - L) \leq c_0 \log \xi_0 - 2$$

for a large enough c_0. Combining this, event $\mathcal{E}_1$, and (5.42), we can show that

$$J \leq K + Q$$

and that, for all $t \leq \min\{G, F\}$,

$$\xi_t \leq \xi_0 + 1 \quad \text{and} \quad \mu_t \geq \mu_0$$

So, if $F \leq G$ the lemma holds. Suppose now that $G < F$. Since $G \leq D_{L+2}$, no blocks of depth $< L + 2$ are split up to step G; hence, by $\mathcal{E}_2$, $\mu_G > L$. Also, since $D_{L+3} \leq G < C_{L+1}$, we can show that, by $\mathcal{E}_1$, $\xi_G \leq L + 2$. By considering each of the two cases: $C_L \geq D_{L+3}$ and $C_L < D_{L+3}$, we can show that in both cases either a safe partition is reached between steps $G + 1$ and τ_J, or

$$J < K + Q$$

and

$$\mu_{\tau_J} = L + 1 \qquad \xi_{\tau_J} = L + 3 \qquad \frac{1}{4} + \varepsilon < \ell_{L+1}(S_{\tau_J}) \leq \frac{1}{4} + 2\varepsilon \qquad \ell_{L+3}(S_{\tau_J}) < \frac{1}{4} - \varepsilon$$

In the latter case, a safe partition is reached by step τ_{J+1}. ■

5.8 Proof of Theorem 5.1

The proof of Theorem 5.1 is based on Lemmata 5.15 and 5.19, and it is similar to the proof of Lemma 5.15. Informally, we consider the sequence of times $\langle \tau_0 = 0, \tau_1, \tau_2, \ldots \rangle$, where τ_{i+1} is the earliest step after τ_i such that an almost-safe partition is reach, if S_{τ_i} is not almost-safe; or a safe partition is reach, if S_{τ_i} is almost-safe; or a maximum number of steps has occurred. The first two represent the "good" outcomes for τ_{i+1}, and the third represents the "bad" outcome. Lemmata 5.15 and 5.19 provide bounds for the probability of these outcomes (lower bounds for the probability of the good outcomes, and upper bound for the probability of the bad outcomes). We express the event whose probability we want to bound in Theorem 5.1, as the intersection of four events on the sequence of τ_i, denoted $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}'_1$, and $\mathcal{E}'_2$, and we establish lower bounds for their probability, instead. Roughly speaking, these events describe thresholds for the ratio of good over bad outcomes for the τ_i; each event concerns one of the four subsets of the τ_i such that S_{τ_i} is or is not almost-safe, and ξ_{τ_i} is or is not close to ξ_0.

More formally, consider the following finite sequence of times $\langle \tau_0, \ldots, \tau_\kappa \rangle$, for some sufficiently large κ (such that the arguments we make later on hold.) We let $\tau_0 = 0$, and, for each $i \geq 1$, we define τ_i as follows, where c_1 is equal to the constant c of Lemma 5.15, and c_2 is equal to the constant c of Lemma 5.19:

- if $S_{\tau_{i-1}}$ is *not* almost-safe then

$$\tau_i = \min\{t_i,\, \tau_{i-1} + c_1 \xi_{\tau_{i-1}} 2^{\xi_{\tau_{i-1}}}\}$$

  where

$$t_i = \inf\{t > \tau_{i-1} \,:\, S_{\tau_i} \text{ is almost-safe or } \xi_{\tau_i} = \xi_{\tau_{i-1}} + 2\}$$

- if $S_{\tau_{i-1}}$ is almost-safe then

$$\tau_i = \min\{t_i,\, \tau_{i-1} + c_2 2^{\xi_{\tau_{i-1}}}\}$$

  where

$$t_i = \inf\{t > \tau_{i-1} \,:\, S_{\tau_i} \text{ is safe or } \xi_{\tau_i} = \xi_{\tau_{i-1}} + 2\}$$

Let

$$\mathbf{A} = \{S \in \mathbf{S} \,:\, S \text{ is almost-safe}\}$$

$$\mathbf{L} = \{S \in \mathbf{S} \,:\, \xi(S) \geq \xi_0 - \log \xi_0\}$$

Let also

$$\mathbf{A}' = \mathbf{S} - \mathbf{A} \qquad \mathbf{L}' = \mathbf{S} - \mathbf{L}$$

For each $0 \leq i < \kappa$, we define the indicator random variables

$$G_i = \begin{cases} 1, & \text{if } \xi_{\tau_{i+1}} \leq \xi_{\tau_i} + 1 \text{ and } S_{\tau_{i+1}} \in \mathbf{A} \\ 0, & \text{otherwise} \end{cases}$$

and

$$H_i = \begin{cases} 1, & \text{if } \xi_{\tau_{i+1}} \leq \xi_{\tau_i} + 1 \text{ and } S_{\tau_{i+1}} \text{ is safe} \\ 0, & \text{otherwise} \end{cases}$$

Consider now the following four events. $\mathcal{E}_1$ is the event:

$$\text{“for all } i \in A_1,\ G_i = 1\text{”}$$

where

$$A_1 \text{ is the set of the } 3\lambda_+(\xi_0) \text{ smallest } i \text{ such that } S_{\tau_i} \in \mathbf{A}' \cap \mathbf{L},$$

$\mathcal{E}_2$ is the event:

$$\text{“for the smallest } i \text{ such that } S_{\tau_i} \in \mathbf{A} \cap \mathbf{L},\ H_i = 1\text{”}$$

$\mathcal{E}_1'$ is the event:

$$\text{“}\sum_{i \in A_1'} G_i \geq |A_1'| - 3\lambda_+(\xi_0)\text{”}$$

where

$$A_1' \text{ is the set of the } 6\lambda_+(\xi_0) \text{ smallest } i \text{ such that } S_{\tau_i} \in \mathbf{A}' \cap \mathbf{L}'$$

Finally, $\mathcal{E}_2'$ is the event:

$$\text{“}\textstyle\sum_{i \in A_2'} H_i > 0\text{”}$$

where

$$A_2' \text{ is the set of the } 3\lambda_+(\xi_0) \text{ smallest } i \text{ such that } S_{\tau_i} \in \mathbf{A} \cap \mathbf{L}'$$

(Without loss of generality, we assume that there are enough i of each type to populate the sets $\mathbf{A} \cap \mathbf{L}$ and A_2' — we can always achieve that by manipulating the strategy of the adversary for large t. However, we cannot make the same assumption for A_1 and A_1'. Indeed, for any given starting partition, we can devise a strategy of the adversary that guarantees that an almost-safe partition (of a given ξ) is reached in a bounded number of steps; but there is no strategy of the adversary such that a *non* almost-safe partition is always reached. So, instead, we let A_1 be the set of all the i such that $S_{\tau_i} \in \mathbf{A}' \cap \mathbf{L}$, if there are fewer than $3\lambda_+(\xi_0)$ such i; and similarly for A_1'.)

Next we compute the probability of the above events. Let $\mathcal{A}$ be the event:

$$\{\tau_i = t\} \cap \{S_t = S\} \cap \mathcal{H}$$

where $i, t \in \mathbb{N}$, $S \in \mathbf{S}$, and $\mathcal{H}$ is an event on the first t steps of the $\mathcal{S}$-process such that $\mathbb{Pr}[\mathcal{A}] > 0$. Conditioned on $\mathcal{A}$, the sequence $\langle S_{\tau_i}, S_{\tau_i+1}, \ldots \rangle$ is the partition-sequence of some

$\mathcal{S}$-process (of the same sampling-size functions as the original $\mathcal{S}$-process). So, by Lemma 5.15,

$$\mathbb{P}\mathrm{r}[G_i = 1 \mid \mathcal{A}] = 1 - \mathrm{O}(\xi(S)2^{\xi(S)}e^{-(1/4-2\varepsilon)\lambda_+(\xi(S)-2\log\xi(S))}) \tag{5.43}$$

If $S \in \mathbf{L}$, the above gives

$$\mathbb{P}\mathrm{r}[G_i = 1 \mid \mathcal{A}] = 1 - \mathrm{O}(2^{\xi_0}e^{-(1/4-2\varepsilon)\lambda_+(\xi_0-3\log\xi_0)})$$

and, thus,

$$\mathbb{P}\mathrm{r}[\mathcal{E}_1] = 1 - \mathrm{O}(\xi_0 2^{\xi_0}e^{-(1/4-2\varepsilon)\lambda_+(\xi_0-3\log\xi_0)}) \tag{5.44}$$

Similarly, for $S \in \mathbf{A}$, Lemma 5.19 yields

$$\mathbb{P}\mathrm{r}[H_i = 1 \mid \mathcal{A}] = 1 - \mathrm{O}(\log\xi(S) \cdot 2^{\xi(S)}e^{-(1/4-\varepsilon)\lambda(\xi(S)-2\log\xi(S))}) \tag{5.45}$$

So, for $S \in \mathbf{A} \cap \mathbf{L}$, the above gives

$$\mathbb{P}\mathrm{r}[\mathcal{E}_2] = 1 - \mathrm{O}(\log\xi_0 \cdot (2^{\xi_0}/\xi_0)e^{-(1/4-\varepsilon)\lambda(\xi_0-3\log\xi_0)}) \tag{5.46}$$

To compute a lower bound for $\mathbb{P}\mathrm{r}[\mathcal{E}_1']$ we observe that, by (5.43), for all $\mathcal{A}$ such that $\xi(S) \geq d_0$, for a large enough constant d_0,

$$\mathbb{P}\mathrm{r}[G_i = 1 \mid \mathcal{A}] \geq 3/4$$

We can make the above relation hold for $\xi(S) < d_0$, as well, by choosing the constant β in (5.1) to be sufficiently large. From that we can show, using Chernoff's bound (Theorem 4.2 in [60]), that

$$\mathbb{P}\mathrm{r}[\mathcal{E}_1'] \geq 1 - e^{-(3/4)6\lambda_+(\xi_0)\cdot(1/3)^2\cdot(1/2)} = 1 - e^{-\lambda_+(\xi_0)/4} \tag{5.47}$$

Similarly, by (5.45), we have that (for a large enough β)

$$\mathbb{P}\mathrm{r}[H_i = 1 \mid \mathcal{A}] \geq 1/3$$

and, thus,

$$\mathbb{P}\mathrm{r}[\mathcal{E}_2'] = 1 - (1 - 1/3)^{3\lambda_+(\xi_0)} \geq 1 - e^{-\lambda_+(\xi_0)} \tag{5.48}$$

By (5.44), (5.46), (5.47), and (5.48), we have

$$\mathbb{P}\mathrm{r}[\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_1' \cap \mathcal{E}_2'] = 1 - \mathrm{O}(\xi_0 2^{\xi_0}e^{-(1/4-2\varepsilon)\lambda(\xi_0-3\log\xi_0)})$$

Combining this with Claim 5.24 that we show next, yields the desired result.

Claim 5.24. *If $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_1' \cap \mathcal{E}_2'$ occurs then (i) and (ii) hold for some $\tau \leq c\xi_0 2^{\xi_0}$, where c is a constant > 0.*

Proof. (Similar to the proof of Claim 5.18.) Suppose that $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_1' \cap \mathcal{E}_2'$ occurs. Let

$$K = \inf\{i \,:\, S_{\tau_i} \text{ is safe}\} \quad \text{and} \quad J = \min\{K, \kappa\}$$

Define

$$D_1 = \{i < J \,:\, S_{\tau_i} \in \mathbf{A}' \cap \mathbf{L}\} \qquad D_2 = \{i < J \,:\, S_{\tau_i} \in \mathbf{A} \cap \mathbf{L}\}$$
$$D_1' = \{i < J \,:\, S_{\tau_i} \in \mathbf{A}' \cap \mathbf{L}'\} \qquad D_2' = \{i < J \,:\, S_{\tau_i} \in \mathbf{A} \cap \mathbf{L}'\}$$

Note that $\{D_1, D_2, D_1', D_2'\}$ is a partition of $\{0, \ldots, J-1\}$, so,

$$|D_1| + |D_2| + |D_1'| + |D_2'| = J \tag{5.49}$$

From $\mathcal{E}_2$ and $\mathcal{E}_2'$ it is immediate that

$$|D_2| \leq 1 \tag{5.50}$$
$$|D_2| + |D_2'| \leq 3\lambda_+(\xi_0) \tag{5.51}$$

Also, if any of the above relations holds as equality then $\max D_2 \cup D_2' + 1 = K = J$. The next two relations follow, respectively, from $\mathcal{E}_1$ and (5.51), and from $\mathcal{E}_1'$, (5.51), and the note after (5.51):

$$|D_1| \leq 3\lambda_+(\xi_0) \quad \text{and} \quad |D_1'| \leq 6\lambda_+(\xi_0) \tag{5.52}$$

By the definition of the τ_i, for all $i \in D_1' \cup D_2'$ and $t \in [\tau_i..\tau_{i+1}]$, since $\xi_{\tau_i} < \xi_0 - \log \xi_0$

$$\tau_{i+1} - \tau_i \leq \max\{c_1 \xi_{\tau_i} 2^{\xi_{\tau_i}}, c_2 2^{\xi_{\tau_i}}\} \leq c_3 2^{\xi_0} \quad \text{and} \quad \xi_t \leq \xi_{\tau_i} + 2 < \xi_0 - \log \xi_0 + 2 \tag{5.53}$$

Also, for all $i \in D_1 - \{0\}$, $\xi_{\tau_i} < \xi_0 - \log \xi_0 + 2$, since $i - 1 \in D_1' \cup D_2'$; so,

$$\tau_{i+1} - \tau_i \leq c_1 \xi_{\tau_i} 2^{\xi_{\tau_i}} \leq c_4 2^{\xi_0} \quad \text{and} \quad \xi_t \leq \xi_{\tau_i} + 1 < \xi_0 - \log \xi_0 + 3 \tag{5.54}$$

for all $t \in [\tau_i..\tau_{i+1}]$. If $0 \in D_1$,

$$\tau_1 - \tau_0 \leq c_1 \xi_0 2^{\xi_0} \quad \text{and} \quad \xi_t \leq \xi_0 + 1 \tag{5.55}$$

for all $t \leq \tau_1$. Finally, if $i \in D_2$ then $\xi_{\tau_i} \leq \xi_0 + 1$, so,

$$\tau_{i+1} - \tau_i \leq c_2 2^{\xi_{\tau_i}} \leq 2c_2 2^{\xi_0} \quad \text{and} \quad \xi_t \leq \xi_{\tau_i} + 1 \leq \xi_0 + 2 \tag{5.56}$$

for all $t \in [\tau_i..\tau_{i+1}]$. By (5.49), (5.51), and (5.52),

$$J \leq 12\lambda_+(\xi_0) < \kappa$$

for a large enough κ. So,

$$K = J < 12\lambda_+(\xi_0)$$

Combining (5.50), (5.51) and (5.52) with the bounds for $\tau_{i+1} - \tau_i$ from (5.53)–(5.56), yields

$$\tau_K \leq 9\lambda_+(\xi_0) \cdot c_3 2^{\xi_0} + 3\lambda_+(\xi_0) \cdot c_4 2^{\xi_0} + c_1\xi_0 2^{\xi_0} + 2c_2 2^{\xi_0} \leq c\xi_0 2^{\xi_0}$$

for a sufficiently large c. Finally, combining the bounds for the ξ_t from (5.53)–(5.56), yields

$$\xi_t \leq \xi_0 + 2$$

for all $t \leq \tau_K$. Therefore, (i) and (ii) hold for $\tau = \tau_K \leq c\xi_0 2^{\xi_0}$. ■

Chapter 6

Analysis – Part IV: Putting the pieces together

In this chapter we complete the analysis of our key-space partitioning scheme by combining the results of the previous three chapters. Specifically, using Theorem 3.10, which establishes a coupling of $\mathcal{B}$-processes and $\mathcal{S}$-processes, we show that Theorems 4.1 and 5.1, which we showed for $\mathcal{S}$-processes, readily apply to $\mathcal{B}$-processes, as well. Also, combining these two results we establish a lower bound on the expected fraction of the binary partitions in a $\mathcal{B}$-process that have $\varrho \leq 2$.

We derive the versions of Theorems 4.1 and 5.1 for $\mathcal{B}$-processes in Section 6.1, and the bound on the fraction of balanced binary partitions in a $\mathcal{B}$-process in Section 6.2.

6.1 From a safe/non-safe to a safe binary partition

Theorems 4.1 and 5.1, which we showed for $\mathcal{S}$-processes, carry over directly to $\mathcal{B}$-processes. So, starting from a safe B_0 (i.e., one such that $\mathrm{srt}(B_0)$ is safe), if λ_+ and λ_- are sufficiently large then, with high probability, all binary partitions reached in a number of $\Theta(|B_0|)$ steps have $\varrho \leq 2$, and the last of them is safe. If, instead, B_0 is not safe then a safe binary partition is reached in $\mathrm{O}(\xi(B_0)2^{\xi(B_0)})$ steps, with high probability. The proofs are straightforward applications of Theorem 3.10.

More formally, we extend the definition of "safe" to (non-sorted) binary partitions in the natural way. $B \in \mathbf{B}$ is *safe* if

$$\varrho(B) < 2, \text{ or } \varrho(B) = 2 \text{ and } \max\{\ell_\mu(B), \ell_\xi(B)\} \leq 1/4 + \varepsilon$$

where $\varepsilon = 1/16$. So, B is safe iff $\mathrm{srt}(B)$ is safe. By analogy to the previous two chapters, for a given a $\mathcal{B}$-process, we let

$$\mu_t = \mu(B_t), \quad \xi_t = \xi(B_t), \quad \text{and} \quad \varrho_t = \varrho(B_t)$$

for each $t \geq 0$. Also, for $k \geq 0$, we let

$$\lambda(k) = \min\{\lambda_+(k), \lambda_-(k)\}$$

The analogue of Theorem 4.1 for $\mathcal{B}$-processes is as follows.

Theorem 6.1. *For any $\mathcal{B}$-process such that B_0 is safe, with probability*

$$1 - \mathrm{O}(2^{\mu_0} e^{-(1/4-\varepsilon)\lambda(\mu_0)})$$

there is $\tau \in [c_1 2^{\mu_0} .. c_2 2^{\mu_0}]$, where c_1, c_2 are positive constants, such that

(i) B_τ is safe, and

(ii) for all $t \leq \tau$, $\varrho_t \leq 2$ and $\xi_t \leq \xi_0 + 1$.

Proof. Let $\mathcal{P}_\mathcal{B}$ be a $\mathcal{B}$-process such that B_0 is safe. By Theorem 3.10, there is an $\mathcal{S}$-process $\mathcal{P}_\mathcal{S}$ such that:

- $\mathcal{P}_\mathcal{S}$ has the same λ_+ and λ_- as $\mathcal{P}_\mathcal{B}$, $S_0 = \mathrm{srt}(B_0)$, and

  $$N > c_2 2^{\mu(S_0)}$$

  where c_2 is the constant of Theorem 4.1; and
- we can construct a coupling $\langle \hat{\mathcal{P}}_\mathcal{B}, \hat{\mathcal{P}}_\mathcal{S} \rangle$ of $\mathcal{P}_\mathcal{B}, \mathcal{P}_\mathcal{S}$ such that, for all $t \in [0..N]$, $\hat{S}_t \preceq \mathrm{srt}(\hat{B}_t)$.

By applying Theorem 4.1 to $\hat{\mathcal{P}}_\mathcal{S}$, we obtain that: with probability

$$1 - \mathrm{O}(2^{\mu(\hat{S}_0)} e^{-(1/4-\varepsilon)\lambda(\mu(\hat{S}_0))})$$

there is $\tau \in [c_1 2^{\mu(\hat{S}_0)} .. c_2 2^{\mu(\hat{S}_0)}]$ such that

(i′) $\hat{S}_\tau$ is safe, and

(ii′) for all $t \leq \tau$, $\varrho(\hat{S}_t) \leq 2$ and $\xi(\hat{S}_t) \leq \xi(\hat{S}_0) + 1$.

Combining the above result and the three simple facts

1. $\mu(\hat{S}_0) = \mu(\hat{B}_0)$ and $\xi(\hat{S}_0) = \xi(\hat{B}_0)$ — since $\hat{S}_0 = \mathrm{srt}(\hat{B}_0)$.
2. for all $t \in [1..N]$, $\xi(\hat{S}_t) \geq \xi(\hat{B}_t)$ and $\varrho(\hat{S}_t) \geq \varrho(\hat{B}_t)$ — by Lemma 3.4, since $\hat{S}_t \succeq \mathrm{srt}(\hat{B}_t)$.
3. for all $t \in [1..N]$, if $\hat{S}_t$ is safe then $\hat{B}_t$ is safe — by Lemma 4.2, since $\hat{S}_t \succeq \mathrm{srt}(\hat{B}_t)$.

yields: with probability

$$1 - \mathrm{O}(2^{\mu(\hat{B}_0)} e^{-(1/4-\varepsilon)\lambda(\mu(\hat{B}_0))})$$

there is $\tau \in [c_1 2^{\mu(\hat{B}_0)} .. c_2 2^{\mu(\hat{B}_0)}]$ such that

(i′) $\hat{B}_\tau$ is safe, and

(ii′) for all $t \leq \tau$, $\varrho(\hat{B}_t) \leq 2$ and $\xi(\hat{B}_t) \leq \xi(\hat{B}_0) + 1$.

This and the fact that $\hat{\mathcal{P}}_\mathcal{B}$ has the same distribution as $\mathcal{P}_\mathcal{B}$ yields the desired result. ■

The version of Theorem 5.1 for $\mathcal{B}$-processes is also almost identical to the original. The proof is similar to that of Theorem 6.1 and is omitted.

Theorem 6.2. *Consider any $\mathcal{B}$-process such that, for all $k \geq 0$,*

$$\lambda_+(k) \geq \max\{8(\ln 2)k,\, \beta\} \quad \textit{and} \quad \lambda_-(k) \geq \max\{8k,\, \beta\}$$

where β is a sufficiently large constant. Then, with probability

$$1 - \mathrm{O}(\xi_0^{\mathrm{O}(1)} 2^{\xi_0} e^{-(1/4-2\varepsilon)\lambda(\xi_0)})$$

there is $\tau \leq c\xi_0 2^{\xi_0}$, where c is a positive constant, such that

(i) B_τ is safe, and

(ii) for all $t \leq \tau$, $\xi_\tau \leq \xi_0 + 2$.

6.2 From a safe to an unbalanced to a safe binary partition

In this section, we combine Theorems 6.1 and 6.2 to show that all but a negligible fraction of the binary partitions in a $\mathcal{B}$-process have $\varrho \leq 2$ (provided that the sampling-size functions used are large enough). More specifically, we typically have long intervals during which all binary partitions have $\varrho \leq 2$, interrupted by much smaller intervals where $\varrho > 2$ for some binary partitions. We quantify that by establishing a lower bound on the expected value of the ratio τ/r, where τ is the number of steps required to get from a safe initial partition to a binary partition B with $\varrho > 2$, and r is the number of subsequent steps to get from B back to a safe binary partition. We show that $\mathbb{E}[\tau/r] = \Omega(|B_0|^{1-1/\gamma}/\log|B_0|)$, where γ is a constant that can be made arbitrarily large by choosing large enough λ_+, λ_-. This result holds regardless of the strategy of the adversary. In the special case where the adversary is not allowed to reduce the system size below some threshold $m \geq |B_0|^{1/\gamma}$, we have $\mathbb{E}[\tau/r] = \Omega(m^{\gamma-1}/\log m)$; so, if $m = \Theta(|B_0|)$ then $\mathbb{E}[\tau/r] = \Omega(|B_0|^{\gamma-1}/\log|B_0|)$. The formal statement of this result is as follows.

Theorem 6.3. *Consider any $\mathcal{B}$-process such that:*

(1) B_0 is safe,

(2) for all $k \geq 0$,

$$\lambda(k) \geq \max\{8k, \beta\}$$

where β is a sufficiently large constant, and

(3) for all $t \geq 0$,

$$\mathbb{Pr}[|B_t| \geq m] = 1$$

for some $m \in [1..|B_0|]$.

Let

$$\tau = \inf\{t : \varrho_t > 2\} \quad \textit{and} \quad r = \begin{cases} \inf\{j \geq 1 \,:\, B_{\tau+j} \textit{ is safe}\}, & \textit{if } \tau < \infty \\ 1, & \textit{otherwise} \end{cases}$$

Then,

$$\mathbb{E}[\tau/r] = \Omega\Big(\frac{n}{n^{1/\gamma}\log n} + \frac{m^\gamma}{m\log m}\Big)$$

where $n = |B_0|$ and $\gamma = (1/4 - \varepsilon) \cdot (\log e) \cdot \inf_k\{\lambda(k)/k\}$.

A "malicious" strategy of the adversary, i.e., one that results in asymptotically smallest $\mathbb{E}[\tau/r]$, is, roughly speaking, to quickly reduce the system size to $|B_0|^{1/\gamma}$ (or to m, if $m > |B_0|^{1/\gamma}$), and then keep the size roughly the same. On the opposite side, there are "helpful" strategies of the adversary, for which $\mathbb{E}[\tau/r]$ is unbounded. We can show that, when ADDBLK operations occur sufficiently more often than RMBLK operations then, with positive probability, *all* binary partitions have $\varrho \leq 2$.

We describe the proof of Theorem 6.3 in Section 6.2.2. Before that, in Section 6.2.1, we derive an upper bound on the average number of steps to reach a safe binary partition from a non-safe binary partition; i.e., an "expected-value" version of the "high-probability" bound of Theorem 6.2.

6.2.1 Expected time to reach a safe binary partition

Based on Theorem 6.2, we show an upper bound on the expected number of steps required in a $\mathcal{B}$-process until we reach a safe binary partition, starting from a non-safe initial partition. Specifically,

Lemma 6.4. *For any $\mathcal{B}$-process such that B_0 is not safe and condition (2) of Theorem 6.3 holds, if $r = \inf\{t : B_t \text{ is safe}\}$ then*

$$\mathbb{E}[r] = \mathrm{O}(\xi_0 2^{\xi_0})$$

Proof. Consider the following infinite sequence of times $\langle \tau_0, \tau_1, \ldots \rangle$. Let $\tau_0 = 0$, and, for $i \geq 1$,

$$\tau_i = \min\{t_i,\ \tau_{i-1} + c_1 \xi_{\tau_{i-1}} 2^{\xi_{\tau_{i-1}}}\}$$

where

$$t_i = \inf\{t \geq \tau_{i-1} : B_t \text{ is safe or } \xi_t = \xi_{\tau_{i-1}} + 3\}$$

and c_1 is the constant c of Theorem 6.2. Note that if B_{τ_i} is safe then $\tau_j = \tau_i$, for all $j > i$. We can write r in terms of the τ_i as

$$r = \sum_{i \geq 0} (\tau_{i+1} - \tau_i) \tag{6.1}$$

We will compute an upper bound on the expected value of $\sum_{i \geq 0}(\tau_{i+1} - \tau_i)$. Let

$$J = \inf\{i : B_{\tau_i} \text{ is safe}\}$$

For every $k \geq 0$,

$$\sum_{0 \leq i < k} \mathbb{E}[\tau_{i+1} - \tau_i] = \sum_{0 \leq i < k} \big(\mathbb{E}[\tau_{i+1} - \tau_i \mid J > i] \cdot \mathbb{Pr}[J > i] \big) \tag{6.2}$$

since $\tau_{i+1} - \tau_i = 0$ if $J \leq i$. We now establish upper bounds for $\mathbb{E}[\tau_{i+1} - \tau_i \mid J > i]$ and $\mathbb{Pr}[J > i]$. From the definition of τ_i it follows (by induction) that $\xi_{\tau_i} \leq \xi_0 + 3i$, and

$$\tau_{i+1} - \tau_i \leq c_1 \xi_{\tau_i} 2^{\xi_{\tau_i}} \leq c_1 (\xi_0 + 3i) 2^{\xi_0 + 3i}$$

So,

$$\mathbb{E}[\tau_{i+1} - \tau_i \mid J > i] \leq c_1 (\xi_0 + 3i) 2^{\xi_0 + 3i} \tag{6.3}$$

The probability that $J > i$ can be expressed as

$$\mathbb{Pr}[J > i] = \prod_{j=1}^{i} \mathbb{Pr}[J > j \mid J > j - 1] \tag{6.4}$$

For $j \geq 1$, conditioned on any fixed value B for $B_{\tau_{j-1}}$ and any event on the first τ_{j-1} steps of the $\mathcal{B}$-process, $\langle B_{\tau_{j-1}}, B_{\tau_{j-1}+1}, \ldots \rangle$ is the partition-sequence of some $\mathcal{B}$-process (of initial partition B and the same sampling-size functions as the original $\mathcal{B}$-process). Also if $J > j-1$, $B_{\tau_{j-1}}$ is not safe. So, by Theorem 6.2,

$$\mathbb{P}\mathrm{r}[J > j \mid J > j-1,\, B_{\tau_{j-1}} = B] = \mathrm{O}\big(\xi(B)^{\mathrm{O}(1)} 2^{\xi(B)} e^{-(1/4-2\varepsilon)\lambda(\xi(B))}\big)$$

Since, for all d, $\lambda(d) \geq 8d$, the right-hand side of the above relation is o(1). Thus, for large $|B|$,

$$\mathbb{P}\mathrm{r}[J > j \mid J > j-1,\, B_{\tau_{i-1}} = B] \leq 1/2^5$$

We can make the above relation hold for small $|B|$, as well, by choosing the constant β (in the condition on λ) to be sufficiently large. Therefore,

$$\mathbb{P}\mathrm{r}[J > j \mid J > j-1] \leq 1/2^5$$

By applying the above to the right-hand side of (6.4), we obtain

$$\mathbb{P}\mathrm{r}[J > i] \leq 1/2^{5i} \tag{6.5}$$

Applying now (6.3) and (6.5) to (6.2), yields

$$\sum_{0 \leq i < k} \mathbb{E}[\tau_{i+1} - \tau_i] \leq \sum_{0 \leq i < k} \big(c_1(\xi_0 + 3i)2^{\xi_0+3i}/2^{5i}\big) = c_1\xi_0 2^{\xi_0} \sum_{0 \leq i < k} \big(1/2^{2i} + 3i/(\xi_0 2^{2i})\big)$$
$$\leq 4c_1\xi_0 2^{\xi_0}$$

Letting $k \to \infty$ we obtain

$$\sum_{i \geq 0} \mathbb{E}[\tau_{i+1} - \tau_i] \leq 4c_1\xi_0 2^{\xi_0}$$

Therefore, (6.1) yields

$$\mathbb{E}[r] = \mathbb{E}\Big[\sum_{i \geq 0}(\tau_{i+1} - \tau_i)\Big] = \sum_{i \geq 0} \mathbb{E}[\tau_{i+1} - \tau_i] \leq 4c_1\xi_0 2^{\xi_0} \qquad \blacksquare$$

6.2.2 Proof of Theorem 6.3

If $\mathbb{E}[\tau/r] = \infty$ then the theorem obviously holds; so, we assume that $\mathbb{E}[\tau/r] < \infty$. Note then that, by Markov's inequality, $\mathbb{P}\mathrm{r}[\tau/r = \infty] = 0$ and, thus,

$$\mathbb{P}\mathrm{r}[\tau = \infty] = 0$$

We begin by using Lemma 6.4 to bound $\mathbb{E}[\tau/r]$ from below by the expectation of a quantity that depends only on τ and ξ_τ (and not on r). Conditioned on any fixed value for B_τ and any event $\mathcal{H}$ on the first τ steps of the $\mathcal{B}$-process, $\langle B_\tau, B_{\tau+1}, \ldots\rangle$ is the partition-sequence of some $\mathcal{B}$-process. So, by Lemma 6.4, for any k such that $\mathbb{Pr}[\{\xi_\tau = k\} \cap \mathcal{H}] > 0$,

$$\mathbb{E}[r \mid \xi_\tau = k,\, \mathcal{H}] \leq c' k 2^k$$

for some constant c'. By the convexity of function $1/x$ then and Jensen's inequality,

$$\mathbb{E}[1/r \mid \xi_\tau = k,\, \mathcal{H}] \geq 1/\,\mathbb{E}[r \mid \xi_\tau = k,\, \mathcal{H}] \geq \frac{1}{c' k 2^k}$$

Therefore,

$$\mathbb{E}[\tau/r] = \mathbb{E}\left[\mathbb{E}[\tau/r \mid \xi_\tau, \tau]\right] = \mathbb{E}\left[\tau \cdot \mathbb{E}[1/r \mid \xi_\tau, \tau]\right] \geq \mathbb{E}\left[\frac{\tau}{c' \xi_\tau 2^{\xi_\tau}}\right] \tag{6.6}$$

Next we derive a lower bound for τ, by partitioning $[0..\tau]$ into a number of smaller intervals, and applying Theorem 6.1 to each of these smaller intervals. More precisely, consider the following infinite sequence of times $\langle \tau_0, \tau_1, \ldots\rangle$. Let $\tau_0 = 0$, and, for each $i \geq 1$, let

$$\tau_i = \min\left\{\tau,\, \tau_{i-1} + c_2 2^{\xi_{\tau_{i-1}}},\, t_i\right\}$$

where

$$t_i = \begin{cases} \inf\{t \geq \tau_{i-1} + c_1 2^{\xi_{\tau_{i-1}}-2} \,:\, B_t \text{ is safe}\}, & \text{if } B_{\tau_{i-1}} \text{ is safe} \\ \inf\{t > \tau_{i-1} \,:\, B_t \text{ is safe}\}, & \text{otherwise} \end{cases}$$

and c_1, c_2 are the constants of Theorem 6.1. It is easy to see that there is an index $J < \infty$ such that the sequence of τ_i is strictly increasing for $i \leq J$, and for all $i \geq J$, $\tau_i = \tau$. ($J < \infty$ because we assumed that $\tau < \infty$ with probability 1.) We can show that for every $i < J$, and all t such that $\tau_i \leq t < \tau_{i+1}$,

$$\varrho_t \leq 2 \quad \text{and} \quad \xi_t \leq \begin{cases} \xi_{\tau_i} + 2, & \text{if } B_{\tau_i} \text{ is safe} \\ \xi_{\tau_i}, & \text{otherwise} \end{cases} \tag{6.7}$$

Also, if B_{τ_i} is safe then

$$\varrho_{\tau_{i+1}} \leq 2 \text{ and } \xi_{\tau_{i+1}} \leq \xi_{\tau_i} + 2, \text{ or } \varrho_{\tau_{i+1}} = 3 \text{ and } \xi_{\tau_{i+1}} \leq \xi_{\tau_i} + 3 \tag{6.8}$$

If B_{τ_i} is not safe (and $i < J$) then

$$\varrho_{\tau_{i+1}} \leq 2 \text{ and } \xi_{\tau_{i+1}} \leq \xi_{\tau_i}, \text{ or } \varrho_{\tau_{i+1}} = 3 \text{ and } \xi_{\tau_{i+1}} \leq \xi_{\tau_i} + 1 \tag{6.9}$$

We can express τ in terms of the τ_i as

$$\tau = \sum_{i \geq 0} (\tau_{i+1} - \tau_i)$$

So, if

$$D = \{i \, : \, S_{\tau_i} \text{ is safe and } i \neq J - 1\}$$

then

$$\tau \geq \sum_{i \in D} (\tau_{i+1} - \tau_i)$$

and, since $\tau_{i+1} - \tau_i \geq c_1 2^{\xi_{\tau_i} - 2}$ for all $i \in D$,

$$\tau \geq \sum_{i \in D} c_1 2^{\xi_{\tau_i} - 2}$$

Applying this to (6.6), yields

$$\mathbb{E}[\tau/r] \geq \mathbb{E}[R], \quad \text{where } R = \frac{c_1}{c' \xi_\tau 2^{\xi_\tau}} \sum_{i \in D} 2^{\xi_{\tau_i} - 2} \tag{6.10}$$

Next we compute a lower bound for $\mathbb{E}[R]$. We distinguish two cases, depending on whether $m \leq n^{1/\gamma}$ or $m > n^{1/\gamma}$.

Case A: $m \leq n^{1/\gamma}$.

Roughly speaking, we consider the event (denoted $\mathcal{E}$) that, for every (not very small) depth k, if $\xi_\tau = k$ then the minimum number of elements $i \in D$ such that $\xi_{\tau_i} \geq k$ exceeds some threshold ν_k. We show that this event implies $R = \Omega\left(n/(n^{1/\gamma} \log n)\right)$, and that it occurs with a constant probability.

Let

$$k_0 = \log n^{1/\gamma} = \frac{1}{\gamma} \log n$$

For each $k \geq k_0$ and $j \geq 1$, let $\pi_{k,j}$ be the index of the j-th smallest among the τ_i for which B_{τ_i} is safe and $\xi_{\tau_i} \geq k$; if no such τ_i exists then $\pi_{k,j} = \infty$. We denote by $\mathcal{E}_k$ the event:

"for all $i \in \{\pi_{k,1}, \ldots, \pi_{k,\nu_k}\}$ such that $i \neq \infty$, $B_{\tau_{i+1}}$ is safe"

where

$$\nu_k = \frac{n}{n^{1/\gamma} k_0} \cdot \frac{k}{4c_3} \tag{6.11}$$

and c_3 is a positive constant we will determine later — see (6.13). Let

$$\mathcal{E} = \bigcap_{k \geq k_0} \mathcal{E}_k$$

Let also

$$K = \xi_\tau - 3$$

If $K \geq k_0$ and $\mathcal{E}$ occurs then, using (6.7)–(6.9), it is easy to show that

$$|\{i \in D \,:\, \xi_{\tau_i} \geq K\}| \geq \nu_K$$

So,

$$R \geq \frac{\nu_K c_1 2^{K-2}}{c'(K+3)\cdot 2^{K+3}} \geq \frac{n}{n^{1/\gamma}k_0} \cdot \frac{c_1}{2^9 c_3 c'}$$

If $K < k_0$ and $\mathcal{E}_{\xi_0}$ occurs then

$$R \geq \frac{c_1 2^{\xi_0 - 2}}{c'(K+3)\cdot 2^{K+3}} \geq \frac{n}{n^{1/\gamma}k_0} \cdot \frac{c_1}{2^7 c'}$$

for large n. Therefore, if $\mathcal{E}$ occurs then (for any value of K)

$$R \geq c_4 \frac{n}{n^{1/\gamma}k_0}$$

where c_4 is a positive constant. So,

$$\mathbb{E}[R] \geq \mathbb{E}[R \mid \mathcal{E}] \cdot \mathbb{P}\mathrm{r}[\mathcal{E}] \geq c_4 \frac{n}{n^{1/\gamma}k_0}\, \mathbb{P}\mathrm{r}[\mathcal{E}] \tag{6.12}$$

We now compute a lower bound for $\mathbb{P}\mathrm{r}[\mathcal{E}]$. For each $i \geq 0$, conditioned on any fixed value for B_{τ_i} and any event $\mathcal{H}_i$ on the first τ_i steps of the $\mathcal{B}$-process, $\langle B_{\tau_i}, B_{\tau_i+1}, \ldots\rangle$ is the partition-sequence of some $\mathcal{B}$-process. So, by Theorem 6.1, for any k such that $\mathbb{P}\mathrm{r}[\{B_{\tau_i} \text{ is safe}\} \cap \{xi_{\tau_i} = k\} \cap \mathcal{H}_i] > 0$,

$$\begin{aligned}\mathbb{P}\mathrm{r}[B_{\tau_{i+1}} \text{ is safe} \mid B_{\tau_i} \text{ is safe}, \xi_{\tau_i} = k, \mathcal{H}_i] &\geq 1 - c_3 2^k e^{-(1/4-\varepsilon)\lambda(k)} \\ &\geq 1 - c_3 2^{k(1-\gamma)}\end{aligned} \tag{6.13}$$

where c_3 is a sufficiently large constant. (To obtain the second inequality we used the assumption that $\lambda(k) \geq 8k$.) Note that $\gamma > 1$, so, the above probability goes to 1, as $k \to \infty$. By (6.13), for all $k \geq k_0$,

$$\mathbb{P}\mathrm{r}[\mathcal{E}_k] \geq (1 - c_3 2^{k(1-\gamma)})^{\nu_k} \tag{6.14}$$

We can obtain a lower bound for the right-hand side of the above relation as follows. Note that

$$\nu_k c_3 2^{k(1-\gamma)} = \frac{1}{4} \cdot \frac{k}{k_0} \cdot 2^{(k-k_0)\cdot(1-\gamma)} = \frac{1}{4} \cdot \Big(\frac{k}{k_0} \cdot 2^{-(k-k_0)\cdot(\gamma-2)}\Big) 2^{-k+k_0}$$

Since $\gamma \geq 8(1/4-\varepsilon)\log e > 2$, if k_0 is large enough then $(k/k_0)\cdot 2^{-(k-k_0)\cdot(\gamma-2)}$ is a decreasing function of k, for $k \geq k_0$. So, for all $k \geq k_0$,

$$\nu_k c_3 2^{k(1-\gamma)} \leq \frac{1}{4}\cdot\Big(\frac{k_0}{k_0}\cdot 2^{-(k_0-k_0)\cdot(\gamma-2)}\Big)2^{-k+k_0} = \frac{1}{4}2^{-k+k_0}$$

So,

$$(1-c_3 2^{k(1-\gamma)})^{\nu_k} \geq 1-\nu_k c_3 2^{k(1-\gamma)} \geq 1-\frac{1}{4}2^{-k+k_0}$$

Combining this and (6.14) yields

$$\mathbb{Pr}[\mathcal{E}_k] \geq 1-\frac{1}{4}2^{-k+k_0}$$

Therefore,

$$\mathbb{Pr}[\mathcal{E}] \geq 1-\frac{1}{4}\sum_{k\geq k_0} 2^{-k+k_0} = \frac{1}{2}$$

By applying the above to (6.12), we obtain

$$\mathbb{E}[R] \geq c_4\frac{n}{2n^{1/\gamma}k_0}$$

hence, by (6.10),

$$\mathbb{E}[\tau/r] \geq c_4\frac{n}{2n^{1/\gamma}k_0}$$

Since $m \leq n^{1/\gamma}$, $\frac{n}{n^{1/\gamma}k_0} \geq \frac{m^\gamma}{m\log m}$, so, the above relation yields

$$\mathbb{E}[\tau/r] = \Omega\Big(\frac{n}{n^{1/\gamma}\log n}+\frac{m^\gamma}{m\log m}\Big)$$

Case B: $m > n^{1/\gamma}$.

This is very similar to Case A: In (6.11) we substitute $n^{1/\gamma}$ for m, thus,

$$\nu_k = \frac{m^\gamma}{m\log m}\cdot\frac{k}{4c_3}$$

and then we proceed in the same way. The details are omitted.

Chapter 7

Greedy routing in uniformly-augmented rings

In this chapter and the next, we study the complexity of a natural decentralized routing protocol, in a broad family of random networks. Specifically, the network model we consider is the directed ring that is augmented by adding links from each node to a number of randomly selected "long-range contacts" of the node, such that, for each node, the set of ring distances to its long-range contacts is chosen independently from the same distribution. (The distribution is a parameter of the model.) The routing protocol we consider is the greedy protocol with respect to the ring distance. This combination of network topology and routing scheme captures many designs proposed for P2P networks, and models for social networks. We show that for any network in this family with n nodes and *on average* ℓ long-range contacts per node, the expected number of steps for greedy routing is $\Omega((\log^2 n)/\ell a^{\log^* n})$, for some constant $a > 1$. This result improves an earlier lower bound of $\Omega((\log^2 n)/\ell \log\log n)$ by Aspnes *et al.* [6] and is very close to the upper bound of $\mathrm{O}((\log^2 n)/\ell)$ achieved in Kleinberg's (one-dimensional) "small-world" model [39], a particular instance of the model we study.

7.1 Introduction

Consider the following model of random graphs on the set of nodes $[0..n-1]$. We start with the nodes forming a *directed* ring, where each node u is connected to its *successor* node $(u+1) \bmod n$. We define the *ring distance* from node u to node v as the number of edges along the ring from u to v, i.e., $(v-u+n) \bmod n$. Then, from each node u, we add directed links to the nodes in a random set of nodes, called the *long-range contacts* of u, that are chosen as follows. Independently for each u, we choose Δ_u according to some probability distribution φ (the same for all u) on the powerset of $[2..n-1]$. (Note that $[2..n-1]$ is the set of all possible ring distances from a node to the remaining nodes, excluding the node's successor.) The long-range contacts of u are the nodes whose ring distance from u is in Δ_u, i.e., the nodes $(u+d) \bmod n$, for all $d \in \Delta_u$. We denote this model by $\mathcal{G}(\varphi)$, and we call the random graphs generated according to this model *uniformly-augmented rings*, or *augmented rings*, for short.

The above model captures a wide range of different graph topologies. Depending on the distribution φ used, the resulting construction may be deterministic (when one subset of $[2..n-1]$ has probability one, and all others have probability zero), or randomized. In the latter case, the number of long-range contacts per node may vary between nodes. Also, in general, the long-range contacts of the same node are *not* chosen independently of each other. An example of a deterministic augmented ring is the Chord ring [77], where

$$\varphi(\Delta) = 1, \qquad \text{for } \Delta = \left\{ \lfloor n/2^i \rfloor \;:\; 1 \le i \le \log n - 1 \right\}$$

An example of a randomized augmented ring is Kleinberg's one-dimensional model for "small-worlds" [39], where each node u has ℓ long-range contacts and the ring distance from u to each of them is chosen independently at random such that it is equal to j with probability $\propto 1/j^\alpha$, for a constant $\alpha \ge 0$.[1] E.g., for $\ell = 1$,

$$\varphi(\Delta) = \begin{cases} 1/(j^\alpha f), & \text{if } \Delta = \{j\}, \text{ for } j = 2, \ldots, n-1 \\ 0, & \text{otherwise} \end{cases}$$

where f is the normalizing constant $\sum_{j=2}^{n-1}(1/j^\alpha)$. A simple example where the out-degree of nodes vary and the long-range contacts of the same node are not chosen independently of each other is as follows. Each node u with probability $p \in (0,1)$ has no long-range contacts, and with probability $1-p$ it has two: node $(u+R_u) \bmod n$ and node $(u+R_u+Q_u) \bmod n$, where R_u and Q_u are chosen independently and uniformly at random from the set $[2..\lfloor n/2 \rfloor - 1]$.

A natural decentralized routing scheme for augmented rings is the following *greedy* protocol: A node forwards a message for destination t to its neighbor v (successor or long-range contact) that minimizes the remaining ring distance to t. As we discuss in Section 7.2, the combination of augmented rings and greedy routing provides an attractive model for the design of P2P networks, and it also captures models used for social networks. We investigate

[1] Strictly speaking, if $\ell > 1$ a node may have fewer than ℓ long-range contacts, since the ℓ distances to its long-range contacts are chosen independently *with replacement* from $[2..n-1]$.

the complexity of routing in this model. More precisely, we focus on the *expected delivery time*, that is the expected value of the average number of steps required to route a message between nodes, where the average is taken over all possible source–destination pairs, and the expectation is over the random construction of the graph. We establish a lower bound on the expected delivery time, as a function of the number of nodes n and the *expected* number ℓ of long-range contacts per node. This bound holds for *all* possible distributions φ.

In his seminal work on routing in social networks [39], Kleinberg described a simple instance of $\mathcal{G}(\varphi)$, where the expected delivery time for greedy routing is $\mathrm{O}((\log^2 n)/\ell)$. In this model, each node u has the same number $\ell \leq \log n$ of long-range contacts, and the ring distance from u to each of them is chosen independently according to the harmonic distribution. No augmented rings that achieve better than $\Theta((\log^2 n)/\ell)$ have been described yet. On the other hand, Aspnes *et al.* [6] proved that, for any distribution φ, the expected delivery time for greedy routing in $\mathcal{G}(\varphi)$ is $\Omega((\log^2 n)/\ell \log\log n)$.

We reduce the gap between the two results above by improving the lower bound to $\Omega((\log^2 n)/\ell a^{\log^* n})$, for some constant $a > 1$. Note that the quantity $a^{\log^* n}$ grows slower than any constant number of iterative applications of log to n — it is "practically" a constant. The proof of this result proceeds by deriving a recursive formula that bounds the expected delivery time for greedy routing in any augmented ring of a given size in terms of that in an exponentially smaller augmented ring. Our analysis suggests general structural properties of an asymptotically optimal augmented ring that are similar to those observed by Kleinberg for the class of graphs he studied. We conjecture that the lower bound can be further improved to $\Omega((\log^2 n)/\ell)$ — i.e., that Kleinberg's model is in fact asymptotically optimal for greedy routing in $\mathcal{G}(\varphi)$.

In the rest of this chapter, we discuss the advantages of the model we study and survey related work in Section 7.2, and we state our result formally in Section 7.3. The proof of this result is described in Chapter 8.

7.2 Discussion and related work

Our results are limited by two assumptions: First, we focus on uniformly-augmented rings; second, we focus on greedy routing. We now explain the benefits of these assumptions.

Despite its simplicity, $\mathcal{G}(\varphi)$ can describe a wide range of graphs, by suitable choice of the probability distribution φ used to determine the long-range contacts. For example, we can describe deterministic constructions (where one subset of $[2..n-1]$ has probability one, and all others have probability zero), as well as probabilistic ones. Moreover, we can describe both homogeneous probabilistic networks, where all nodes have the same number of long-range contacts, and heterogeneous ones where the number of long-range contacts can vary (a little or a lot) from node to node. The "uniformity" property, i.e., that each node uses the same distribution to determine its long-range contacts, is also beneficial. It implies that a node's position in the ring doesn't influence its choice of long-range contacts (i.e., their number, and their ring distances from the node.) Since nodes are effectively equivalent, it is harder for an adversary to disrupt the system by attacking critical nodes.

The advantages of greedy routing are well-known, and reflected by its popularity: Routing decisions are made locally and independently in each node. These decisions are also independent of the routing path up to the current node, so, messages need not store routing information other than the destination node. As a result of these two properties, greedy routing is easy to implement. Also, it is inherently fault-tolerant since as long as each node has *some* edge towards the destination, the message will reach it. *Bidirectional* greedy routing is a variation of the (unidirectional) version that we consider in this paper, where a node forwards a message for node t to its neighbor v that minimizes the "absolute ring distance" to t: $\min\{(t-v+n) \bmod n,\ (v-t+n) \bmod n\}$.[2]

In view of the advantages of the two assumptions underlying our analysis, it is not surprising that the designs of many P2P systems fall within the purview of these assumptions. There have been proposed deterministic designs with $\ell = \Theta(\log n)$ employing both unidirectional and bidirectional greedy routing (Chord [77, 26]); there are also probabilistic designs with ℓ ranging from 1 to $\Theta(\log n)$ using either version of greedy routing (Kleinberg's small-world networks [39], Symphony [52], Randomized-Chord [29, 84]). In all of these systems the expected delivery time is $\Theta((\log^2 n)/\ell)$. (For the tightness of this bound for the probabilistic designs above see [11, 53].)

On the lower bound side, the following facts are known about the number of steps required for *unidirectional* greedy routing in augmented rings. Xu [83] has shown that for any *deterministic* construction with $\ell = \Theta(\log n)$, the number of steps required is $\Omega(\log n)$ *in the worst-case* — i.e., for *some* source–destination pair routing takes $\Omega(\log n)$ steps. Aspnes *et al.* [6] have shown that, for *any* distribution φ, the expected delivery time is $\Omega((\log^2 n)/\ell \log\log n)$. We improve this bound to $\Omega((\log^2 n)/\ell a^{\log^* n})$, for some constant

[2] To ensure that in augmented rings bidirectional greedy routing can always reach the destination, we slightly modify our model by requiring that, in addition to its successor, each node u is also connected to its *predecessor* $(u-1+n) \bmod n$ in the ring.

$a > 1$. For *bidirectional* greedy routing in augmented rings, Aspnes *et al.* [6] showed that under some assumptions on φ, the expected delivery time is $\Omega((\log^2 n)/\ell^2 \log\log n)$. Also, Flammini *et al.* [19] have shown that in the special case where each node has exactly one long-range contact and certain assumptions on its distribution apply, the maximum expected number of steps over all source–destination pairs is $\Omega(\log^2 n)$.

In a balanced tree of degree ℓ spanning n nodes, the average distance of a node from the root is $\Theta(\log n/\log \ell)$. Thus, this bound represents the optimal expected delivery time in an n-node network of degree ℓ. Note that this bound is (asymptotically) better than the lower bound for greedy routing in augmented rings. In particular, the lower bound of Aspnes *et al.* implies that, for $\ell = \mathrm{o}(\log n)$, the combination of augmented rings and greedy routing cannot achieve an optimal tradeoff between node degree and routing paths length. Our improved lower bound establishes that this is also true when $\ell = \Theta(\log n)$, a case of practical interest since many P2P designs have degree $\Theta(\log n)$.

In view of the lower bound showing that the combination of augmented rings and greedy routing cannot achieve an optimal degree–routing paths length tradeoff, designs that achieve such an optimal tradeoff must abandon at least one of the assumptions underlying that bound: Either they must be based on constructions that are not augmented rings, or they must use non-greedy routing — or both. As we argued earlier, these assumptions have considerable advantages, and so the gain of more efficient routing has to be weighed against the loss of these advantages.

We now give some examples of proposed designs that achieve more efficient decentralized routing than is possible with greedy routing in augmented rings. Papillon [3] is an example of a network that achieves optimal routing by abandoning only the first assumption. It uses greedy routing but the underlying graph is a *non-uniformly* augmented ring (a ring-embedded butterfly-like network). A similar construction is also described in [19]. The so-called “neighbor-of-neighbor” approach of [15, 53] is an example where better routing performance is achieved by abandoning only the second assumption. Using a non-greedy algorithm (where the routing decision at each node is based not only on the node's neighbors, but also on *their* neighbors) it improves routing performance in Kleinberg's small-world model. Finally, several deterministic and randomized designs have been proposed that achieve optimal routing by abandoning both assumptions: They use non-greedy routing in constructions that are not augmented rings, such as the de Bruijn graph, randomized versions of it, or randomized versions of the butterfly; e.g., [48, 34, 49, 62].

We conclude this section with a brief survey of results in two related research areas:

routing in small worlds, and networks for DHTs.

7.2.1 Decentralized routing in small worlds

The *small-world phenomenon* — the premise that almost all pairs of people in a society are connected by short chains of acquaintances, was first verified experimentally by Milgram [57]. Milgram's experiments also revealed that individuals are able *to find* such short paths efficiently using only local information. Kleinberg [39] proposed a simple framework to model this routing aspect of the small-world phenomenon. He modeled the graph of acquaintances as a d-dimensional n-node grid augmented by adding links from each node to a small number ℓ of long-range contacts selected independently at random; each long-range contact of a node u is chosen to be node v with probability $\propto 1/\mathit{dist}(u,v)^{\alpha}$, for a constant α, where $\mathit{dist}(u,v)$ is the grid distance between u and v. Kleinberg showed that for $\alpha = d$ greedy routing, with respect to *dist*, achieves expected delivery time $\mathrm{O}((\log^2 n)/\ell)$, while for $\alpha \neq d$ the expected delivery time for any decentralized routing algorithm is polynomial in n.

Kleinberg's work inspired a large body of subsequent research. Variations and extensions of his model were proposed where different base structures than the grid were used: trees [40, 80], sets of groups [40], or the grid with non-uniformly populated lattice points [46]. Also variations of the greedy routing strategy were studied [24, 54, 42, 53]; in all these variations improved routing performance is achieved by allowing each node to "consult" a small number of nearly nodes for free. For the diameter of Kleinberg's grid-based family of networks see [54, 55]. Finally, lower bounds for the complexity of greedy routing were derived for variations of Kleinberg's model where more general distributions for choosing the long-range contacts of each node are used [6, 19, 28]; these results suggest that the distribution Kleinberg proposed is possibly (asymptotically) optimal. Our work can be viewed as part of this last volume of work.

A number of recent papers study the problem, proposed by Fraigniaud in [20], of whether it is possible to construct random graphs that support efficient greedy routing (i.e., that requires at most a poly-logarithmic number of steps) by augmenting an *arbitrary* base graph. In particular, each node of the base graph is augmented by a long-range contact selected independently from some distribution over the remaining nodes (possibly, a different distribution for each node). In the greedy routing scheme considered, a node forwards a message to its neighbor that has the shortest path to the destination in the base graph (not in the augmented one). This question was answered in the affirmative for certain classes of graphs [20, 18, 75, 23], and in the negative for the general case [23].

For a more detailed survey of work in decentralized routing in small worlds see [41] and [21].

7.2.2 Routing networks for DHTs

Recall from Chapter 1 that a main component of a DHT is an overlay routing network, which facilitates efficient decentralized routing from any node to the node responsible for any given key in the key-space. The connections in this network are determined by the position of nodes (i.e., of their blocks) in the key-space. The standard approach used in designing networks for DHTs is to, first, find a *static family of graphs* that support efficient decentralized routing, and then show how to construct in a distributed manner a network with a topology that "approximates" the topology of this static family of graphs.

Routing networks for DHTs can be broadly classified into two categories: *deterministic* and *randomized*. The topology of a deterministic routing network is a function of the current partition of the key-space into blocks, while the topology of a randomized routing network depends also on additional random choices — other than those made to determine the partition of the key-space.

Deterministic routing networks for DHTs are typically based upon classical parallel interconnection networks, such as the hypercube, the butterfly, and the de Bruijn graph [43]. Examples of early such designs include CAN [68], whose routing network is an adaptation of the d-dimensional torus, and Chord [77], Pastry [72], Tapestry [31], and Kademlia [56], all of which are variations of the hypercube. (Pastry, Tapestry, and Kademlia were inspired by a prefix-based routing scheme proposed in [64].) In all these DHTs the routing schemes used are greedy with respect to some distance function in the key-space. CAN achieves routing paths of length $\mathrm{O}(n^{1/d})$[3] with $\Theta(d)$ links per node, in an n-node system; and the other four DHTs achieve routing paths of length $\mathrm{O}(\log n)$ using $\Theta(\log n)$ links per node. More recently, a number of designs that are based on high-degree de Bruijn graphs were proposed [62, 2, 34, 22]. The routing schemes they employ are non-greedy, and they achieve an optimal tradeoff between node degree and path length: for $\Theta(k)$ links per node the routes are of length $\mathrm{O}(\log n/\log k)$. Two deterministic variations of the butterfly, proposed in [19, 3], also achieve optimal routing performance, using greedy routing.

A variety of randomized routing networks have been designed for DHTs. These in-

[3] For all source–destination pairs.

clude Viceroy [48] and Mariposa [49, 51] (two randomized butterfly networks), Randomized-Chord [29, 84], Randomized-Hypercube [29], Skip-Graphs [7, 30] (an adaptation of skip-lists [65]), and Symphony [52, 6] (an adaptation of Kleinberg's small-world construction [39]). Viceroy achieves routes of length $\mathrm{O}(\log n)$[4] with only $\mathrm{O}(1)$ links per node, which is optimal. Mariposa also achieves optimal degree–routing paths length tradeoff using k links per node, where k is a parameter of the model. Randomized-Chord, Randomized-Hypercube and Skip-Graphs have a node degree of $\Theta(\log n)$ and achieve routing paths of length $\Theta(\log n)$, when greedy routing is used. By employing the neighbor-of-neighbor approach the length of the routing paths becomes optimal, i.e., $\mathrm{O}(\log n/\log\log n)$ [53]. Symphony achieves greedy routing paths of length $\mathrm{O}(\log^2 n/k)$ using k links per node; using the neighbor-of-neighbor approach routing paths of length $\mathrm{O}(\log^2 n/(k\log k))$ are achieved, which is optimal when $k = \log n$.

For a more detailed survey of work on routing networks for DHTs see [47].

7.3 Rigorous statement of our result

Let G be an instance of an n-node augmented ring. For every $u, v \in [0..n-1]$, the *delivery time* from node u to node v in G, denoted $L(G, u, v)$, is the length of the greedy routing path in G from source u to destination v. Recall that the greedy routing path in G from u to v is the path $\langle u_0 = u, u_1, \ldots, u_k = v\rangle$, such that, for each $i < k$, $u_i \neq v$ and u_{i+1} is the neighbor of u_i that is of minimum ring distance $(v - u_{i+1} + n) \bmod n$ to the destination v.

For every $n \geq 2$ and $\ell \in [0, n-2]$, we denote by $\mathbf{\Phi}_{n,\ell}$ the set of all probability distributions φ on the powerset of $[2..n-1]$ such that the expected number of long-range contacts per node in $\mathcal{G}(\varphi)$ is ℓ; i.e.,

$$\sum_{\Delta\subseteq[2..n-1]} \big(|\Delta| \cdot \varphi(\Delta)\big) = \ell$$

The *expected delivery time* in $\mathcal{G}(\varphi)$, denoted $T(\varphi)$, is the expected value of the average delivery time in $\mathcal{G}(\varphi)$, where the average is taken over all source–destination pairs; i.e.,

$$T(\varphi) = \mathbb{E}\left[\frac{1}{n^2}\sum_{0\leq u,v<n} L(G, u, v)\right]$$

where G is randomly generated in $\mathcal{G}(\varphi)$. Finally, by $T(n, \ell)$ we denote the optimal expected delivery time in $\mathcal{G}(\varphi)$, over all φ for n nodes such that the expected number of long-range

[4] For all source–destination pairs, with high probability.

contacts per node is ℓ; i.e.,

$$T(n, \ell) = \inf\{T(\varphi) \,:\, \varphi \in \mathbf{\Phi}_{n,\ell}\}$$

We can now state our main result. Below we assume that ℓ is a non-decreasing function of n, perhaps a constant.[5] The asymptotic notation we use is for $n \to \infty$.

Theorem 7.1. *If $\ell = \Omega(1)$ then $T(n, \ell) = \Omega((\log^2 n)/\ell a^{\log^* n})$, where a is a constant > 1.*

The corresponding upper bound, which follows mostly from previously known results, is

Theorem 7.2. *If $\ell = \mathrm{O}(\log n)$ then $T(n, \ell) = \mathrm{O}((\log^2 n)/\ell)$.*

We describe the proofs of these results in Chapter 8.

[5] Technically, the following weaker condition on ℓ suffices: $\ell = \Theta(g)$, where g is a non-decreasing function of n.

Chapter 8

Proof of the lower bound

In this chapter we describe the proofs of Theorems 7.1 and 7.2, stated in Section 7.3. In Section 8.1, we state four auxiliary result and use them to derive Theorems 7.1 and 7.2. We prove these results in Sections 8.3, 8.5, and 8.7. In Section 8.2 we introduce some terminology. In Sections 8.4 and 8.6 we define the *routing tree* of an augmented ring — a structure we use in our analysis, and discuss some of its properties.

8.1 Statement of auxiliary results and derivation of Theorems 7.1 and 7.2

We begin with three lemmata that allow us to bound $T(n, \ell)$ in terms of $T(n', \ell')$, for $n' \neq n$ or $\ell' \neq \ell$. The first lemma states the intuitive result that $T(n, \ell)$ is a non-increasing function of ℓ.

Lemma 8.1. *If $\ell < \ell'$ then $T(n, \ell) \geq T(n, \ell')$.*

The next lemma says what happens to $T(n, \ell)$ for fixed ℓ, as n increases. One might expect $T(n, \ell)$ to be a non-decreasing function of n. This is not necessarily the case, since some "convenient" values of n (say, powers of 2) may result in smaller T than smaller values of n. We show the following weaker result, which, however, suffices for our analysis.

Lemma 8.2. *If $n > n'$ then $T(n, \ell) \geq (n'/n) \cdot T(n', \ell)$.*

The third lemma is more interesting than the previous two. Lemma 8.1 shows that, for fixed n, as ℓ increases $T(n, \ell)$ decreases; Lemma 8.3 says it does not decrease too much.

Lemma 8.3. *If $\ell > \ell'$ then $T(n, \ell) \geq (\ell'/\ell) \cdot T(n, \ell')$.*

The main part of our analysis is the proof of the following result, which gives a lower bound for T when $\ell = 1$. We use $T(n)$ as a shorthand for $T(n, 1)$.

Theorem 8.4. *$T(n) = \Omega((\log^2 n)/a^{\log^* n})$, for some constant $a > 1$.*

Before we proceed to prove the above results, we show how we can use them to derive Theorems 7.1 and 7.2.

Proof of Theorem 7.1. Since ℓ is a non-decreasing function of n, we have that, for all sufficiently large values of n, it is $\ell < 1$, or $\ell > 1$, or $\ell = 1$. If $\ell < 1$ then, by Lemma 8.1 (applied for $\ell' = 1$) and Theorem 8.4, we have

$$T(n, \ell) = \Omega((\log^2 n)/a^{\log^* n})$$

Combining this with the fact that ℓ is larger than some positive constant (since $\ell = \Omega(1)$), yields the desired bound for $T(n, \ell)$, when $\ell < 1$. If $\ell > 1$ the desired bound follows from Lemma 8.3 (applied for $\ell' = 1$) and Theorem 8.4. Finally, the case where $\ell = 1$ is handled in Theorem 8.4. ■

Proof of Theorem 7.2. Kleinberg [39] showed that in the n-node augmented ring where each node u has a single long-range contact, and this long-range contact is selected to be node v with probability inversely proportional to the ring distance from u to v, the expected delivery time is $O(\log^2 n)$. Using a similar technique, Aspnes *et al.* [5] showed that if each node chooses k long-range contacts, each selected independently with replacement from the same distribution as in Kleinberg's model, then the expected delivery time is $O((\log^2 n)/k)$, for all integers k such that $1 \leq k \leq \log n$.[1] Note that in the family of augmented rings that Aspnes *et al.* analyzed the expected number k' of long-range contacts per node is $k' \leq k$, since for each node the same long-range contact may be selected more than once. Therefore,

$$T(n, k) \leq T(n, k') = O((\log^2 n)/k) \tag{8.1}$$

where the first relation holds because of Lemma 8.1. Also, by Lemma 8.1, for all real $\ell \geq 1$,

$$T(n, \ell) \leq T(n, \lfloor \min\{\ell, \log n\} \rfloor)$$

[1] These results were shown for a slightly different model than ours. Specifically, the links to successors are bidirectional; the absolute ring distance is considered when choosing the long-range contacts of a node; and bidirectional greedy routing is used. Also the long-range contacts of a node form a *multi-set*, and a long-range contact may be at ring distance 1 from the node. Despite these differences, essentially the same proofs can be used for our model.

Combining the above two results, yields the desired bound for $T(n,\ell)$, for all ℓ that take real values such that $1 \leq \ell = \mathrm{O}(\log n)$. For $\ell < 1$, the bound follows by combining (8.1) (for $k = 1$), and the fact that $T(n,\ell) \leq T(n)/\ell$, which follows from Lemma 8.3. ■

8.2 Definitions

We now describe some definitions we will use in the rest of this chapter. For $n \geq 2$, we denote by $\mathbf{G}_n$ the set of all directed graphs on the set of nodes $[0..n-1]$ that contain as a subgraph the directed ring $0 \to 1 \to 2 \to \cdots \to (n-1) \to 0$. Thus, $\mathbf{G}_n$ consists of all the graphs that are instances of some n-node augmented ring. For every $G \in \mathbf{G}_n$ and node u of G, we call the set of ring distances from u to its long-range contacts the *delta-set* of u in G. E.g., for the graph in Figure 8.1(a) on page 141, the delta-sets of nodes 1 and 2 are $\emptyset$ and $\{3,4\}$, respectively.

Recall from Section 7.3 that the expected delivery time in $\mathcal{G}(\varphi)$ is the expected value of the average delivery time, where the average is taken over all the n^2 source–destination pairs of nodes (n is the number of nodes in $\mathcal{G}(\varphi)$). Because of the way the long-range contacts of nodes are selected, it is equivalent to consider, instead, the average of the delivery times from a fixed source, say node 0, to all (the n) possible destinations. Furthermore, instead of the average over the n possible destinations, we could take the expectation for a destination selected uniformly at random. Formally, for any $G \in \mathbf{G}_n$, let

$$L(G) = \frac{1}{n} \sum_{0 \leq v < n} L(G, 0, v)$$

Then,

$$T(\varphi) = \mathbb{E}[L(G)] \tag{8.2}$$

where G is randomly generated in $\mathcal{G}(\varphi)$. Also, if X is a uniform random variable over $[0..n-1]$ that is independent of the construction of G then

$$T(\varphi) = \mathbb{E}[L(G, 0, X)] \tag{8.3}$$

where the expectation is over the random construction of G and the random selection of X.

8.3 Proofs of Lemmata 8.1 and 8.2

We now describe the proofs of the first two of the auxiliary lemmata in Section 8.1. Both proofs employ the same technique, which is based on the coupling method [79]: For an arbitrary $\varphi \in \mathbf{\Phi}_{n,\ell}$, we construct a coupling $\langle G, G' \rangle$ of $\mathcal{G}(\varphi)$ and $\mathcal{G}(\varphi')$, for some $\varphi' \in \mathbf{\Phi}_{n',\ell'}$ (for suitable n', ℓ', depending on the lemma) — i.e., we describe a joint construction of random graphs G and G' such that their marginal distributions are the same as the distributions of $\mathcal{G}(\varphi)$ and $\mathcal{G}(\varphi')$, respectively. For this pair of graphs, we show that

$$L(G) \geq \alpha L(G') \tag{8.4}$$

(for a suitable α, depending on the lemma). So, by (8.2),

$$T(\varphi) \geq \alpha T(\varphi')$$

and since we assumed an arbitrary $\varphi \in \mathbf{\Phi}_{n,\ell}$,

$$T(n, \ell) \geq \alpha T(n', \ell') \tag{8.5}$$

Specifically, in the coupling construction we generate G first, according to the $\mathcal{G}(\varphi)$ model, and then we construct G' based on the G constructed and, possibly, on some additional random choices. Below we denote by Δ_u, for $0 \leq u < n$, the delta-set of node u in G, and by Δ'_v, for $0 \leq v < n'$, the delta-set of node v in G'.

Proof of Lemma 8.1. For this lemma $n' = n$ and $\alpha = 1$. We let φ' be a distribution obtained from φ by reducing the probability associated with (some of) the *proper* subsets of $[2..n-1]$, and correspondingly increasing the probability of the entire set $[2..n-1]$, such that the resulting distribution is in $\mathbf{\Phi}_{n,\ell'}$. For instance, one such φ' is defined by

$$\varphi'(\Delta) = \begin{cases} (1-q) \cdot \varphi(\Delta), & \text{if } \Delta \subset [2..n-1] \\ \varphi([2..n-1]) + q \sum_{\Delta' \subset [2..n-1]} \varphi(\Delta'), & \text{if } \Delta = [2..n-1] \end{cases}$$

where

$$q = \frac{\ell' - \ell}{(n-2) - \ell}$$

It is straightforward to show that the above is a valid distribution, and that

$$\sum_{\Delta} \left(|\Delta| \cdot \varphi'(\Delta) \right) = \ell'$$

G' is then generated as follows. For every $u \in [0..n-1]$ such that $\Delta_u = [2..n-1]$ (i.e., node u has outgoing edges to all other nodes in G), we let $\Delta'_u = \Delta_u$; for each of the remaining u, we let $\Delta'_u = \Delta_u$ or $\Delta'_u = [2..n-1]$ with probabilities $\varphi'(\Delta_u)/\varphi(\Delta_u)$ and $1 - \varphi'(\Delta_u)/\varphi(\Delta_u)$, respectively — the random choice for each u is made independently. It is straightforward to verify that G' has the same distribution as $\mathcal{G}(\varphi')$. Also, for any $u \neq 0$, if $\langle v_0 = 0, v_1, \ldots, v_k = u\rangle$ is the greedy routing path in G' from 0 to u then $\langle v_0, \ldots, v_{k-1}\rangle$ is a proper prefix of the greedy routing path from 0 to u in G. Thus, for all u,

$$L(G, 0, u) \geq L(G', 0, u)$$

which yields (8.4), for $\alpha = 1$. Hence, by (8.5), $T(n, \ell) \geq T(n, \ell')$. ■

Proof of Lemma 8.2. We define G' as a function of G. For each $u \in [0..n' - 1]$, we let

$$\Delta'_u = \Delta_u \cap [0..n' - 1]$$

Note that the greedy routing path in G' from 0 to each u is identical to the corresponding routing path in G, and, thus,

$$L(G', 0, u) = L(G, 0, u)$$

So,

$$L(G') = (1/n') \sum_{u<n'} L(G, 0, u) \leq (1/n') \sum_{u<n} L(G, 0, u) = (n/n') \cdot L(G)$$

— i.e., (8.4) holds for $\alpha = n'/n$. Note also that G' has the same distribution as $\mathcal{G}(\varphi')$ for some $\varphi' \in \mathbf{\Phi}_{n',\ell'}$ where

$$\ell' = \sum_{\Delta \subseteq [0..n-1]} \big(|\Delta \cap [0..n' - 1]| \cdot \varphi(\Delta)\big) \leq \ell$$

Therefore, by (8.5),

$$T(n, \ell) \geq (n'/n) \cdot T(n', \ell') \geq (n'/n) \cdot T(n', \ell)$$

where the second inequality holds because of Lemma 8.1 (since $\ell' \leq \ell$). ■

8.4 Routing trees

In this section, we describe a structure we will use in the proofs of the remaining two auxiliary results, Lemma 8.3 and Theorem 8.4. For every $G \in \mathbf{G}_n$, the *routing tree* of G is the subgraph of G that consists of the greedy routing paths from node 0 to all the other nodes. An example is illustrated in Figures 8.1(a) and (b).[2] The next lemma states an invariant of routing trees. (The proof follows from the properties of greedy routing and is omitted.) By $\langle\!\langle i, k\rangle\!\rangle$, for $i, k \in \mathbb{Z}$, we denote the set $[i..i + k - 1]$.

[2] Note that for each $u \in [1..n-1]$, we can similarly define the routing tree where the source of the routing paths is node u (instead of 0). However, in all the routing trees we consider in our analysis we assume that the source is node 0.

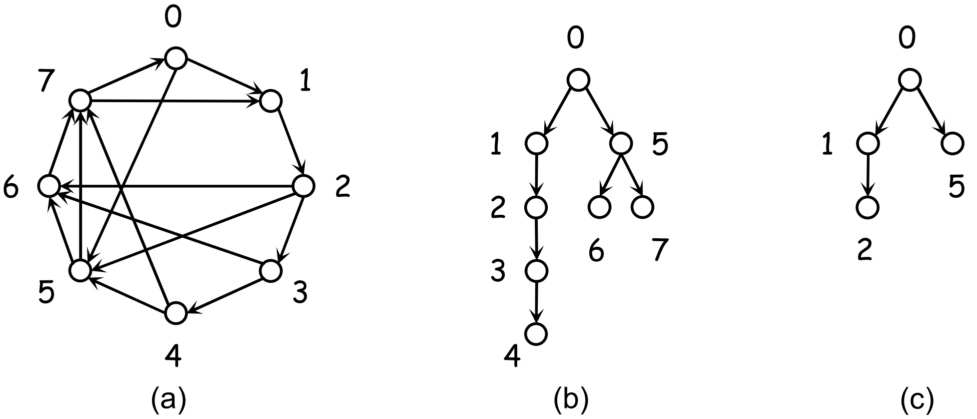

Figure 8.1: (a) Example of a $G \in \mathbf{G}_8$; (b) the routing tree of G; (c) the 2-prefix of G's routing tree.

Lemma 8.5. *Let R be the routing tree of a graph in $\mathbf{G}_n$. Then,*

(a) R is a tree, and

(b) for every node u, the subtree of R rooted at u consists of the nodes in $\langle\!\langle u, s\rangle\!\rangle$, where s is the size of the subtree.

Note that the depth of each node u in R is the delivery time $L(G, 0, u)$. So, the average node depth in R is equal to $L(G)$, and if G is randomly generated in $\mathcal{G}(\varphi)$ then the expected average depth is equal to $T(\varphi)$. Note also that part (b) of the lemma can be equivalently stated as: a pre-order walk of R visits the nodes in increasing order. (We assume that the children of each internal node of R are sorted from left to right in increasing order.)

Suppose now we have only partial knowledge of G; specifically, suppose we just know its size n, and the delta-sets of the first $t \in [0..n]$ nodes, $0, \ldots, t-1$. (For $t = 0$, we do not know the delta-set of any node.) What can we infer about the routing tree R of G? Consider the subgraph of R induced by the nodes $0, \ldots, t-1$ and their children in R; we call this subgraph the *t-prefix* of R. (For $t = 0$, the t-prefix of R consists only of node 0.) An example is shown in Figure 8.1(c). Based on Lemma 8.5 we can show the following result. (The proof is by induction on t and is omitted).

Lemma 8.6. *Let $G \in \mathbf{G}_n$, R be the routing tree of G, and R^t be the t-prefix of R, for some $t \in [0..n]$. Then, R^t is a tree, and is completely determined given n and the delta-sets of nodes $0, \ldots, t-1$. Each node of G that is not in R^t is in a subtree of R rooted at some leaf of R^t. For each leaf u of R^t, the subtree of R rooted at u has size $u - u'$, where u' is the smallest node of R^t that is larger than u, or $u' = n$ if no such node exists.*

8.5 Proof of Lemma 8.3

As in the proof of Lemmata 8.1 and 8.2, we employ a coupling argument. For an arbitrary $\varphi \in \mathbf{\Phi}_{n,\ell}$, we describe a coupling $\langle G, G' \rangle$ of $\mathcal{G}(\varphi)$ and $\mathcal{G}(\varphi')$, for some $\varphi' \in \mathbf{\Phi}_{n,\ell'}$, such that

$$\mathbb{E}[L(G)] \geq \alpha\, \mathbb{E}[L(G')], \qquad \text{for } \alpha = \ell'/\ell \tag{8.6}$$

So, by (8.2), we have $T(\varphi) \geq \alpha T(\varphi')$, which yields the desired result, $T(n,\ell) \geq \alpha T(n,\ell')$ — since φ is an arbitrary distribution in $\mathbf{\Phi}_{n,\ell}$.

Below we denote by R and R' the routing trees of G and G', respectively. Also, for each $u \in [0..n-1]$, we let Δ_u and Δ'_u be the delta-sets of u in G and G', respectively.

We begin with an informal description of the coupling. We generate G first, according to the $\mathcal{G}(\varphi)$ model. Then, based on the G constructed we construct G' inductively by considering each node $u = 0, 1, \ldots, n-1$ in turn. With each node u of G' we associate a node C_u of G (this association is not necessarily one-to-one). We initialize the inductive construction by setting $C_0 = 0$ — i.e., associating with the root of R' the root of R. For the node u of G' under consideration, we define Δ'_u and, simultaneously, define the association of u's children in R' to nodes in R. Specifically, we choose $\Delta'_u = \emptyset$ with probability $1-\alpha$, and $\Delta'_u = \Delta_{C_u}$ with probability α. So, in the first case u has no long-range contacts in G' and (if u is not a leaf of R') its only child in R' is $u+1$. In this case, we associate with $u+1$ the node in G to which u is already associated, i.e., $C_{u+1} = C_u$. In the second case, u has long-range contacts in G' at the same ring distances as the corresponding node C_u does in G and this defines the set of u's children in R'. In this case, we associate with each child $u+\delta$ of u in R' the corresponding child of C_u in R, i.e., $C_{u+\delta} = C_u + \delta$.

We now describe the coupling construction more formally. We choose $\Delta_0, \ldots, \Delta_{n-1}$ independently at random, each according to distribution φ — as in $\mathcal{G}(\varphi)$. Also, we independently choose n random bits $B_0, \ldots, B_{n-1}$, such that, for each $0 \leq u < n$, $\mathbb{Pr}[B_u = 1] = \alpha$. Then, for each $u = 0, \ldots, n-1$, Δ'_u is defined inductively by

$$\Delta'_u = \begin{cases} \emptyset, & \text{if } B_u = 0 \\ \Delta_{C_u}, & \text{if } B_u = 1 \end{cases}$$

where

$$C_u = \begin{cases} 0, & \text{if } u = 0 \\ C_{F_u}, & \text{if } u > 0 \text{ and } B_{F_u} = 0 \\ C_{F_u} + (u - F_u), & \text{if } u > 0 \text{ and } B_{F_u} = 1 \end{cases}$$

and F_u is u's parent in the u-prefix of R' — which is also u's parent in R.

That G' is well defined is immediate from the fact that the u-prefix of R' is completely determined given $\Delta'_0, \ldots, \Delta'_{u-1}$ (by Lemma 8.6), and from the following claim. (The proof of the claim is by an easy induction and is omitted).

Claim 8.7. *For all $u \in [0..n-1]$, $0 \le C_u \le u$.*

The next result gives more insight into the construction of G'. (The proof, by induction, is omitted). S_u and S'_u denote the sizes of the subtrees of R and R', respectively, rooted at u. Recall (from Lemma 8.5) that the sets of nodes of these subtrees are $\langle\!\langle u, S_u \rangle\!\rangle$ and $\langle\!\langle u, S'_u \rangle\!\rangle$, respectively.

Claim 8.8. *For all $u \in [0..n-1]$,*

(a) $S'_u \le S_{C_u}$.

(b) *For all $u' \in [u+1, n-1]$,*

- *If $u' < u + S'_u$ then $C_{u'} \in \langle\!\langle C_u, S_{C_u} \rangle\!\rangle$; in particular, if $B_u = 1$ then $C_{u'} \neq C_u$.*
- *If $u' \ge u + S'_u$ then $C_{u'} \ge C_u + S_{C_u}$*

Part (b) says that if $u' > u$ then $C_{u'} \ge C_u$ (where the inequality is strict if $B_u = 1$), and $C_{u'}$ is in the subtree of R rooted at C_u iff u' is in the subtree of R' rooted at u.

We now show that the marginal distribution of G' is the same as the distribution of $\mathcal{G}(\varphi')$, for some $\varphi' \in \mathbf{\Phi}_{n,\ell'}$. For that, it is convenient to think of the construction of G' as a random process consisting of n steps, 0 up to $n-1$, where in step t we decide the value of Δ'_t, and the value of each B_u and Δ_u is generated right before it is about to be used — not earlier. Clearly, B_t is generated in step t. Let U_t be the set of nodes u for which Δ_u is generated in some of the steps $0, \ldots, t-1$. Note that $U_t = \{C_v : v < t,\ B_v = 1\}$, so, by Claim 8.8(b), $C_t \notin U_t$; i.e., Δ_{C_t} is not generated before step t. Therefore, in each step t: we first choose B_t (independently of past choices); if $B_t = 0$ we set $\Delta'_t = \emptyset$; otherwise, we choose Δ_{C_t} (again independently of past decisions) and let $\Delta'_t = \Delta_{C_t}$. Consequently, $\Delta'_0, \ldots, \Delta'_{n-1}$ are generated independently at random, each according to the distribution φ' defined by

$$\varphi'(\Delta) = \begin{cases} \alpha\varphi(\Delta), & \text{if } \Delta \neq \emptyset \\ \alpha\varphi(\emptyset) + (1-\alpha), & \text{if } \Delta = \emptyset \end{cases}$$

Note that $\varphi' \in \mathbf{\Phi}_{n,\ell'}$, since

$$\sum_{\Delta} \big(|\Delta| \cdot \varphi'(\Delta)\big) = \sum_{\Delta \neq \emptyset} \big(|\Delta| \cdot \varphi'(\Delta)\big) = \alpha \sum_{\Delta \neq \emptyset} \big(|\Delta| \cdot \varphi(\Delta)\big) = \alpha\ell = \ell'$$

It remains to prove (8.6), i.e., that $\mathbb{E}[L(G)] \geq \alpha\,\mathbb{E}[L(G')]$. Roughly speaking, we describe a sequence $L_0, \ldots, L_n$ of progressively more accurate estimates of $L(G')$, where estimate L_t is based on G and the binary string

$$A_t = \langle B_0, \ldots, B_{t-1} \rangle$$

The first of these estimates is $L_0 = \alpha^{-1} L(G)$, the last one is $L_n = L(G')$, and for all $t < n$, we have $\mathbb{E}[L_{t+1}] \leq \mathbb{E}[L_t]$. Combining these three facts yields the desired result.

For every $t \in [0..n]$, we let R^t be the t-prefix of R' and V^t be the set of nodes of R^t, and we define

$$L_t = \frac{1}{n}\left(\sum_{u\in[0..t-1]} L(G', 0, u) + \sum_{u \in V^t - [0..t-1]} \left(S'_u \cdot L(G', 0, u) + \alpha^{-1} \sum_{v \in \langle\!\langle C_u, S'_u \rangle\!\rangle} L(G, C_u, v)\right)\right)$$

Before we explain the above formula, we establish that $L_t = L_t(G, A_t)$. Recall that $[0..t-1] \subseteq V^t$, and that, for all $u \in V^t$, $L(G', 0, u)$ and S'_u are a function of R^t (by Lemma 8.6). Also, since R^t is a function of $\Delta'_0, \ldots, \Delta'_{t-1}$ (by Lemma 8.6), and these delta-sets are a function of G, A_t (by the coupling construction), we have that, for all $u \in V^t$, $L(G', 0, u)$ and S'_u are a function of G, A_t. Also, for all $u \in V^t$, $C_u = C_u(G, A_t)$ (by the coupling construction). Therefore, $L_t = L_t(G, A_t)$.

In the definition of L_t, the first sum accounts for the lengths of the greedy routing paths from 0 to the first t nodes of G'; the second sum is an estimate of the lengths of the routing paths to the remaining nodes of G' — by Lemma 8.6, these nodes are in the subtrees of R' rooted at all $u \geq t$ that are leaves of R^t. Specifically, inside this second sum, the first term accounts for the lengths of the routing paths up to u for all (the S'_u) nodes in R''s subtree rooted at u; the second term is an estimate of the lengths of the routing paths from u to these nodes; this estimate is proportional to the sum of the lengths of the routing paths in G from C_u to the nodes $C_u, C_u + 1, \ldots, C_u + S'_u - 1$, which, by Claim 8.8(a), are all in the subtree of R rooted at C_u.

Note that

$$L_0 = \alpha^{-1} L(G) \qquad \text{and} \qquad L_n = L(G') \tag{8.7}$$

Let $Z_{t+1} = n(L_{t+1} - L_t)$. It is straightforward to show that

$$Z_{t+1} = \begin{cases} S'_t - 1 - \alpha^{-1} L(G, C_t, C_t + S'_t - 1), & \text{if } B_t = 0 \\ -(\alpha^{-1} - 1) \cdot (S'_t - 1), & \text{if } B_t = 1 \end{cases}$$

We then have

$$\begin{aligned}\mathbb{E}[Z_{t+1} \mid G, A_t] &= \mathbb{E}[Z_{t+1} \mid G, A_t, B_t = 0] \cdot \mathbb{Pr}[B_t = 0 \mid G, A_t] \\ &\qquad + \mathbb{E}[Z_{t+1} \mid G, A_t, B_t = 1] \cdot \mathbb{Pr}[B_t = 1 \mid G, A_t] \\ &= \big(S'_t - 1 - \alpha^{-1} L(G, C_t, C_t + S'_t - 1)\big) \cdot (1-\alpha) \\ &\qquad - (\alpha^{-1} - 1) \cdot (S'_t - 1) \cdot \alpha \\ &= -(\alpha^{-1} - 1) \cdot L(G, C_t, C_t + S'_t - 1) \\ &\le 0\end{aligned}$$

Therefore, $\mathbb{E}[L_{t+1} - L_t \mid G, A_t] \le 0$. Taking the expectation of both sides yields $\mathbb{E}[L_{t+1} - L_t] \le 0$, or, equivalently, $\mathbb{E}[L_{t+1}] \le \mathbb{E}[L_t]$, which implies that

$$\mathbb{E}[L_n] \le \mathbb{E}[L_0]$$

Substituting the values of L_n and L_0 from (8.7), we obtain (8.6).

8.6 More on routing trees

We now describe some additional concepts and terminology about routing trees. Let $G \in \mathbf{G}_n$ and R be the routing tree of G. Also, for every $u \in [0..n-1]$, let R_u be the subtree of R rooted at node u, and s_u be the size of R_u. Recall from Lemma 8.5(b) that the set of nodes of R_u is $\langle\!\langle u, s_u \rangle\!\rangle = [u..u + s_u - 1]$. A node v of G is called an *r-descendant* of u, for some $r \ge 1$, if v is a node of R_u and $v - u < r$; or, equivalently, if $v \in \langle\!\langle u, \min\{r, s_u\} \rangle\!\rangle$. If the parent of v in R is an r-descendant of u, but v itself is not then v is called an *r-successor* of u. Examples of these definitions are illustrated in Figure 8.2(a). If $s_u \le r$, all the nodes of R_u are r-descendants of u; i.e., u has no r-successors. If $s_u > r$, the following picture emerges: Let $p = \langle u_0, \ldots, u_k \rangle$ be the path in R_u from u to the largest r-descendant of u, i.e., node $u + r - 1$; we call p the *r-path* of u. For example, the 7-path of node 1 in Figure 8.2(a) is $\langle 1, 5, 7 \rangle$. The r-descendants of u are the nodes along p, plus the nodes of the subtrees rooted at the children of nodes $u_0, \ldots, u_{k-1}$ that lie to the *left* of p on the plane; the r-successors of u are the children of nodes $u_0, \ldots, u_{k-1}$ that lie to the *right* of p, plus all the children of u_k. Note that the r-successors of u form a "frontier" between the r-descendants of u and the other nodes of R_u: The path from u to each of its r-descendants consists only of r-descendants of u; and the path to any other node of R_u consists of one or more r-descendants of u, followed

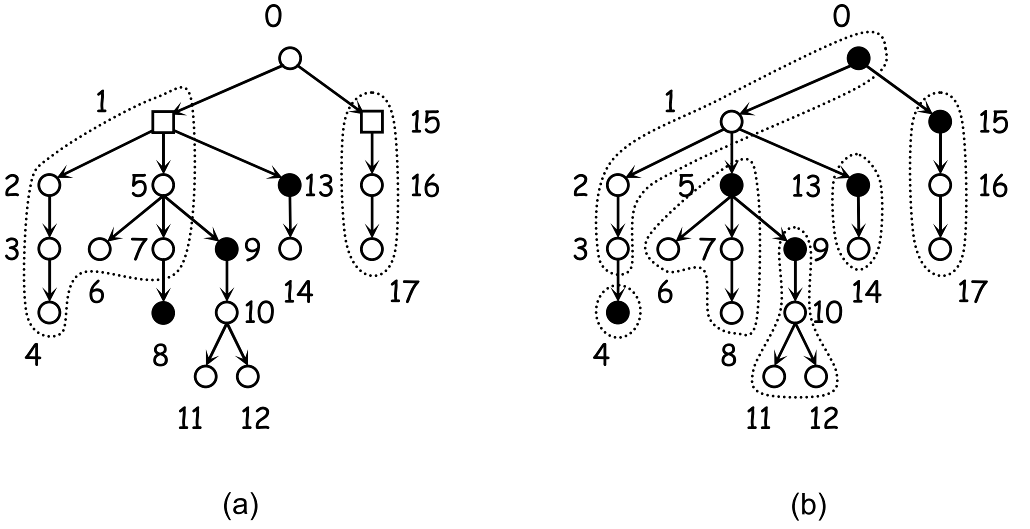

Figure 8.2: (a) The 7-descendants of nodes 1 and 15 are the nodes $1, \ldots, 7$, and $15, 16, 17$, respectively; the 7-successors of node 1 are the nodes $8, 9, 13$; (b) the 4-significant nodes are marked with filled circles; the $(12, 4)$-partition is $\{[0..3], \{4\}, [5..8], [9..11]\}$.

by *one* r-successor of u, and then zero or more nodes that are neither r-descendants nor r-successors of u.

Consider now the set of nodes that consists of node 0, 0's r-successors, the r-successors of them, and so on. We call the nodes in this set the r*-significant* nodes of G. Also, if v is an r-descendant of some r-significant node u, we say that u is the r*-ancestor* of v. Let $z_0 = 0 < z_1 < \cdots < z_{\kappa-1}$ be the r-significant nodes of G, and $z_\kappa = n$. Then, for each $k \in [0..\kappa - 1]$, the r-descendants of z_k (or, equivalently, the nodes whose r-ancestor is z_k) are the nodes in $[z_k..z_{k+1} - 1]$. For every $m \in [1..n]$, the partition $[0..m - 1]$ into the sets

$$[z_0..z_1 - 1],\ [z_1..z_2 - 1], \ldots,\ [z_{\kappa'-1}..m - 1]$$

where $\kappa' = \min\{k : z_k \geq m\}$, is called the (m, r)*-partition with respect to* G. An example is described in Figure 8.2(b).

From the above, it follows that the routing path in G from node 0 to any node consists of one or more r-significant nodes, each followed by zero or more r-descendants of it. For example, in the path $\langle \mathbf{0}, 1, \mathbf{5}, \mathbf{9}, 10, 12\rangle$ of the routing tree in Figure 8.2(b), nodes $0, 5, 9$ are 4-significant nodes, node 1 is a 4-descendant of 0, and nodes $10, 12$ are 4-descendants of 9.

For every node v of G, it is possible to identify the r-significant nodes in the routing path from 0 to v in an "on-line" fashion as we perform greedy routing from 0 to v — we assume that initially (when at node 0) we have no knowledge of G other than its size n, and we learn the delta-set of each node in the path when we visit that node. We can achieve that using the following simple algorithm. Below by Δ_u we denote the delta-set of node u.

- Let S denote the set of r-significant nodes identified so far; initially, $S = \emptyset$.
- We add 0 to S — we do not know Δ_0, yet.
- Each time we add a node u to S:
  - ◦ If $v - u < r$ the algorithm ends.
  - ◦ Otherwise:
    - – we learn Δ_u (and, thus, the next node in the path to v)
    - – while the next node in the path is $< u + r$ we move to that node, and learn its delta-set
    - – let w be the first node in the path such that $w - u \geq r$ — we do not know Δ_w, yet
    - – we add w to S.

It is straightforward to verify that when the algorithm ends the nodes in S are the r-significant nodes in the routing path from 0 to v. For each $u \in S$, we can also compute the number of its r-descendants, based only on the delta-sets of the nodes that precede u in this path; it is equal to $\min\{r, u' - u\}$ where u' is the smallest among the neighbors that are greater than u of the nodes that precede u in the path; or $u' = n$, if no such node exists.

The next lemma is immediate from the above discussion. It says that the r-ancestor z of v and the number m of the r-descendants of z are completely determined by the delta-sets of the nodes that precede z in the routing path from 0 to z. Also, knowing these delta-sets (and, thus, z and m) does not reveal any information about the delta-sets of the remaining nodes in the graph.

Lemma 8.9. *Let $G \in \mathbf{G}_n$, $v \in [0..n-1]$, z be the r-ancestor of v in G, for some $r \in [1..n]$, and m be the number of r-descendants of z in G. Let also $\langle u_0, \ldots, u_k \rangle$ be the prefix of the greedy routing path in G from 0 to v, up to (but not including) z. Then, for every $G' \in \mathbf{G}_n$ such that nodes $u_0, \ldots, u_k$ of G' have the same delta-sets as the corresponding nodes of G, the r-ancestor of v in G' is z, and the number of r-descendants of z in G' is m.*

8.7 Proof of Theorem 8.4

Let φ be an arbitrary distribution in $\mathbf{\Phi}_{n,1}$. For every $m \in [1..n]$, we denote by Λ_m the expected value of the average of the delivery times in $\mathcal{G}(\varphi)$ from node 0 to the first m nodes;

i.e.,

$$\Lambda_m = \mathbb{E}\left[\frac{1}{m}\sum_{0\leq u<m} L(G,0,u)\right]$$

where G is randomly generated in $\mathcal{G}(\varphi)$. Note that $\Lambda_n = \mathbb{E}[L(G)] = T(\varphi)$.

Informally, the proof proceeds as follows. First, we observe that for any triplet $m, r, \eta \in \mathbb{N}$ such that $\eta \leq r \leq m \leq n$, Λ_m is bounded from below by the sum of the following two terms:

(1) $\mathbb{E}[L(G,0,Z)]$, where Z is the r-ancestor of a uniformly-random node in $[0..m-1]$;

(2) $q \cdot \Lambda_{m^*}$, where q is the probability that Z has at least η r-descendants in $[0, m-1]$, and $m^* \in [\eta..r]$ is such that Λ_{m^*} is minimal.

Roughly speaking, term (1) accounts for the number of steps until we get within distance r of the average destination, and term (2) bounds from below the remaining number of steps to the destination. By recursively apply this result, we then obtain a lower bound for $T(\varphi) = \Lambda_n$. Specifically, in each recursive step we take $r \approx m/\log^{\beta} n$ and $\eta \approx m/\log^{\beta+\gamma} n$, for some constants $\beta, \gamma > 0$. For these values of r and η we show that each term of type (1) is bounded from below by $T(n')/\lambda$, where n' is polylogarithmic in n, and λ is proportional to the expected number of long-range contacts a node has at ring distances between r and m. We also show that the probabilities q in the terms of type (2) are very close to 1. So, the lower bound we obtain for $T(\varphi)$ looks roughly like: $T(n')\cdot\sum_i(1/\lambda_i)$, where i ranges from 1 to the total number of recursive steps, which is $\Theta(\log n/\log\log n)$, and λ_i is the value of λ that corresponds to step i. Then, by observing that $\sum_i \lambda_i$ is bounded from above by a constant, we obtain that $T(\varphi) \geq \Theta(\log^2 n/\log^2\log n)\cdot T(n')$; thus, $T(n) \geq \Theta(\log^2 n/\log^2\log n)\cdot T(n')$. Finally, since n' is polylogarithmic in n, recursive application of the last inequality yields the desired bound for $T(n)$.

Our proof suggests that to achieve (asymptotically) optimal routing performance, φ should be such that a node in $\mathcal{G}(\varphi)$ has *roughly* the same expected number of long-range contacts in all intervals of ring distances that are of the form $[u..u\log^c n]$, for a constant c. Note that Kleinberg's construction has the property that a node has *exactly* the same expected number of long-range contacts in all intervals of ring distances of the form $[u..cu]$.

We now present the detailed proof. Let G be a randomly generated graph in $\mathcal{G}(\varphi)$, and fix $r, m \in \mathbb{N}^*$ such that $r \leq m \leq n$. Let Z be the r-ancestor in G of a uniformly-random node in $[0..m-1]$, selected independently of the choice of G. Let also M be the number of r-descendants of Z in G that are in $[0..m-1]$. We can express Λ_m in terms of Z and M

as follows. Let H be an arbitrary possible instance of $\mathcal{G}(\varphi)$, and Π_H be the (m,r)-partition with respect to H. Then,

$$\frac{1}{m}\sum_{0\le u<m} L(H,0,u) = \frac{1}{m}\sum_{b\in\Pi_H}\sum_{u\in b} L(H,0,u) \tag{8.8}$$

Given $G = H$, the probability that $\langle\langle Z, M\rangle\rangle$ is equal to any fixed block of Π_H is proportional to the size of that block; formally, for every $b \in \Pi_H$,

$$\mathbb{Pr}[\langle\langle Z,M\rangle\rangle = b \mid G = H] = \frac{|b|}{m}$$

Applying the above to the right-hand side of (8.8) yields

$$\begin{aligned}\frac{1}{m}\sum_{0\le u<m} L(H,0,u) &= \sum_{b\in\Pi_H}\left(\mathbb{Pr}[\langle\langle Z,M\rangle\rangle = b \mid G = H]\cdot\frac{1}{|b|}\sum_{u\in b} L(H,0,u)\right)\\ &= \mathbb{E}\left[\frac{1}{M}\sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,0,u)\,\middle|\, G = H\right]\end{aligned}$$

Therefore,

$$\Lambda_m = \mathbb{E}\left[\frac{1}{M}\sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,0,u)\right]$$

Combining this and the fact that

$$\sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,0,u) = M\cdot L(G,0,Z) + \sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,Z,u)$$

we obtain

$$\Lambda_m = \mathbb{E}[L(G,0,Z)] + \mathbb{E}\left[\frac{1}{M}\sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,Z,u)\right] \tag{8.9}$$

We now focus on the term $\mathbb{E}\left[(1/M)\sum_{u\in\langle\langle Z,M\rangle\rangle} L(G,Z,u)\right]$ on the right-hand side of (8.9). By Lemma 8.9, knowledge of the values of Z and M does not reveal any information about the delta-sets of the nodes in $\langle\langle Z,M\rangle\rangle$. More formally, let Δ_u, for each $u \in [0..n-1]$, denote the delta-set of node u. Let also $E_{z,m'}$ denote the event: "$\langle\langle Z,M\rangle\rangle = \langle\langle z,m'\rangle\rangle$." For any z, m' such that $\mathbb{Pr}[E_{z,m'}] > 0$, we have that, conditioned on $E_{z,m'}$, $\Delta_z, \ldots, \Delta_{z+m'-1}$ are independent, and each has distribution φ. From this, it is immediate that for every $u \in \langle\langle z,m'\rangle\rangle$,

$$\mathbb{E}[L(G,Z,u) \mid E_{z,m'}] = \mathbb{E}[L(G,0,u-z)]$$

So, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{M}\sum_{u\in\langle\!\langle Z,M\rangle\!\rangle} L(G,Z,u)\right] &= \sum_{z,m'}\left(\mathbb{E}\left[\frac{1}{m'}\sum_{u\in\langle\!\langle z,m'\rangle\!\rangle} L(G,z,u)\;\middle|\; E_{z,m'}\right]\cdot \mathbb{Pr}[E_{z,m'}]\right)\\
&= \sum_{z,m'}\left(\frac{1}{m'}\sum_{u\in\langle\!\langle 0,m'\rangle\!\rangle}\mathbb{E}[L(G,0,u)]\cdot\mathbb{Pr}[E_{z,m'}]\right)\\
&= \sum_{z,m'}\big(\Lambda_{m'}\cdot\mathbb{Pr}[E_{z,m'}]\big)\\
&= \sum_{1\le m'\le r}\big(\Lambda_{m'}\cdot\mathbb{Pr}[M=m']\big)
\end{aligned}$$

In the last sum, if we restrict the range of m' to $[\eta..r]$, for some $\eta\in[1..r]$, and replace each $\Lambda_{m'}$ with Λ_{m^*}, for some $m^*\in[\eta..r]$ such that $\Lambda_{m^*}\le\Lambda_{m'}$ for all $m'\in[\eta..r]$, we obtain

$$\mathbb{E}\left[\frac{1}{M}\sum_{u\in\langle\!\langle Z,M\rangle\!\rangle} L(G,Z,u)\right]\ge\Lambda_{m^*}\cdot\mathbb{Pr}[M\ge\eta]$$

By applying the above to (8.9), yields

$$\Lambda_m\ge\mathbb{E}[L(G,0,Z)]+\Lambda_{m^*}\cdot\mathbb{Pr}[M\ge\eta] \tag{8.10}$$

Then next lemma provides lower bounds for $\mathbb{E}[L(G,0,Z)]$ and $\mathbb{Pr}[M\ge\eta]$. (Its proof is described at the end of this section.)

Lemma 8.10.

(a) Let $\theta\in[2..r-1]$, and λ be the expected number of long-range contacts of a node at ring distances between θ and $m-1$, i.e.,

$$\lambda=\mathbb{E}\left[|\Delta_0\cap[\theta..m-1]|\right]$$

There are constants $c>0$ and $b>1$ such that if $m/r\ge b$ then

$$\mathbb{E}[L(G,0,Z)]\ge c\cdot\min\left\{\frac{r}{\theta-1},\ \frac{1}{\lambda}\cdot T\Big(\Big\lfloor\frac{m}{r+\theta}\Big\rfloor\Big)\right\}$$

(b)

$$\mathbb{Pr}[M\ge\eta]\ge 1-\frac{\eta-1}{r}\cdot\big(\mathbb{E}[L(G,0,r-1)]+\mathrm{O}(1)\big)$$

Based on (8.10) we can bound $T(\varphi)$ from below as follows. First we define quantities n_i, r_i, η_i, θ_i, λ_i, Z_i, and M_i. The definition is recursive:

$$n_0=n$$

and for each $i = 0, \ldots, \tau - 1$, where

$$\tau = \min\{j \,:\, n_j \leq \log^{\beta+\gamma} n\}$$

we let:

- r_i be the $d \in \left[\lceil n_i / \log^{\beta} n\rceil .. \lceil 2n_i / \log^{\beta} n\rceil\right]$ such that $\mathbb{E}[L(G, 0, d-1)] \leq \mathbb{E}[L(G, 0, d'-1)]$, for all $d' \in \left[\lceil n_i / \log^{\beta} n\rceil .. \lceil 2n_i / \log^{\beta} n\rceil\right]$
- η_i $= \lceil n_i / \log^{\beta+\gamma} n\rceil$
- θ_i $= \lceil n_i / \log^{\beta+\delta} n\rceil$
- n_{i+1} be the $d \in [\eta_i .. r_i]$ such that $\Lambda_d \leq \Lambda_{d'}$, for all $d' \in [\eta_i .. r_i]$
- λ_i $= \mathbb{E}\left[|\Delta_0 \cap [\theta_i .. n_i - 1]|\right]$
- Z_i be the r_i-ancestor of a node chosen independently and uniformly at random from $[0..n_i - 1]$
- M_i be the number of the r_i-descendants of Z_i that are in $[0..n_i - 1]$.

β, γ, and δ are constants such that

$$2 \leq \delta \leq \beta \leq \gamma - 3$$

(Note that all quantities defined above, except for the Z_i and M_i, can be computed directly from φ — they are not random variables.) Now, by recursively applying (8.10) for k times, where $k \in [0..\tau]$, we obtain the following bound for $T(\varphi) = \Lambda_n$:

$$T(\varphi) \geq \sum_{i<k} \left(\mathbb{E}[L(G, 0, Z_i)] \cdot \prod_{j<i} \mathbb{P}\mathrm{r}[M_j \geq \eta_j] \right) + \Lambda_{n_k} \cdot \prod_{j<k} \mathbb{P}\mathrm{r}[M_j \geq \eta_j] \tag{8.11}$$

Next, we use Lemma 8.10 to derive lower bounds for the $\mathbb{E}[L(G, 0, Z_i)]$ and $\mathbb{P}\mathrm{r}[M_i \geq \eta_i]$. By Lemma 8.10(a),

$$\mathbb{E}[L(G, 0, Z_i)] \geq c \cdot \min\left\{ \frac{r_i}{\theta_i - 1},\ \frac{1}{\lambda_i} \cdot T\left(\left\lfloor \frac{n_i}{r_i + \theta_i} \right\rfloor \right) \right\}$$

(We can use Lemma 8.10(a), because for $i < \tau$, $n_i / r_i = \Theta(\log^{\beta} n)$, and, thus, $n_i / r_i > b$ for all large enough n.) Note that since $\frac{1}{2}\log^{\beta} n - 1 \leq \frac{n_i}{r_i + \theta_i} \leq \log^{\beta} n$, we have, by Lemma 8.2, that

$$T\left(\left\lfloor \frac{n_i}{r_i + \theta_i} \right\rfloor \right) \geq \left(\frac{1}{2} - \mathrm{o}(1) \right) \cdot T\left(\left\lfloor \frac{1}{2} \log^{\beta} n - 1 \right\rfloor \right)$$

Also,

$$\frac{r_i}{\theta_i - 1} \geq \log^{\delta} n \geq \log^{2} n$$

since $\delta \geq 2$. Therefore,

$$\mathbb{E}[L(G,0,Z_i)] \geq c' \cdot \min\left\{ \log^2 n,\ \frac{1}{\lambda_i} \cdot T\Big(\Big\lfloor \frac{1}{2}\log^\beta n - 1\Big\rfloor\Big)\right\} \tag{8.12}$$

for some constant $c' > 0$. For $\mathbb{Pr}[M_i \geq \eta_i]$, by Lemma 8.10(b), we have that

$$\mathbb{Pr}[M_i \geq \eta_i] \geq 1 - \frac{\eta_i - 1}{r_i} \cdot \big(\,\mathbb{E}[L(G,0,r_i-1)] + \mathrm{O}(1)\big)$$

Note that, by the definition of r_i,

$$\Big(\Big\lceil \frac{2n_i}{\log^\beta n}\Big\rceil - \Big\lceil \frac{n_i}{\log^\beta n}\Big\rceil + 1\Big) \cdot \mathbb{E}[L(G,0,r_i-1)] \leq n_i \Lambda_{n_i}$$

which implies that

$$\mathbb{E}[L(G,0,r_i-1)] \leq \log^\beta n \cdot \Lambda_{n_i}$$

Also,

$$\frac{\eta_i - 1}{r_i} \leq \frac{1}{\log^\gamma n} \leq \frac{1}{\log^{\beta+3} n}$$

since $\gamma \geq \beta + 3$. Therefore,

$$\mathbb{Pr}[M_i \geq \eta_i] \geq 1 - \hat{c}\frac{\Lambda_{n_i}}{\log^3 n} \tag{8.13}$$

for a constant $\hat{c} > 0$.

We now combine results (8.11)–(8.13) to derive a lower bound for $T(\varphi)$ in terms of $T(n')$, for some n' that is polylogarithmic in n. We assume that $n \geq \tilde{n}$, where $\tilde{n}$ is a constant such that $\tau > 0$ for all $n \geq \tilde{n}$. Note that

$$\frac{1}{\beta+\gamma} \cdot \frac{\log n}{\log\log n} - \mathrm{O}(1) \leq \tau \leq \Big(\frac{2}{\beta} + \mathrm{o}(1)\Big)\frac{\log n}{\log\log n} \tag{8.14}$$

We distinguish two cases.

Case 1: $\max\big\{\Lambda_{n_i},\ \frac{1}{c'}\,\mathbb{E}[L(G,0,Z_i)]\big\} \geq \log^2 n$, for some $i < \tau$.
Let k be the smallest such i. Then, by (8.13), for all $i < k$,

$$\mathbb{Pr}[M_i \geq \eta_i] \geq 1 - \frac{\hat{c}}{\log n}$$

If $\Lambda_{n_k} \geq \log^2 n$ then, by (8.11),

$$T(\varphi) \geq \Lambda_{n_k} \cdot \prod_{j<k} \mathbb{Pr}[M_j \geq \eta_j] \geq \log^2 n \cdot \Big(1 - \frac{\hat{c}}{\log n}\Big)^k$$

Likewise, if $\mathbb{E}[L(G,0,Z_k)] \geq c' \log^2 n$,

$$T(\varphi) \geq \mathbb{E}[L(G,0,Z_k)] \cdot \prod_{j<k} \mathbb{Pr}[M_j \geq \eta_j] \geq c' \log^2 n \cdot \Big(1 - \frac{\hat{c}}{\log n}\Big)^k$$

Since $k < \tau = \mathrm{O}(\log n / \log\log n)$ (by (8.14)), we have that in both cases

$$T(\varphi) \geq c_1 \log^2 n \tag{8.15}$$

for some constant $c_1 > 0$.

Case 2: $\max\left\{\Lambda_{n_i},\ \frac{1}{c'}\,\mathbb{E}[L(G,0,Z_i)]\right\} < \log^2 n$, for all $i < \tau$.
By (8.13), for all $i < \tau$,

$$\mathbb{Pr}[M_i \geq \eta_i] \geq 1 - \frac{\hat{c}}{\log n}$$

So, by (8.11), applied for $k = \tau$, we obtain

$$\begin{aligned} T(\varphi) &\geq \sum_{i<\tau} \Big(\mathbb{E}[L(G,0,Z_i)] \cdot \prod_{j<i} \mathbb{Pr}[M_j \geq \eta_j] \Big) \\ &\geq \Big(1 - \frac{\hat{c}}{\log n}\Big)^{\tau} \sum_{i<\tau} \mathbb{E}[L(G,0,Z_i)] \\ &\geq c'' \sum_{i<\tau} \mathbb{E}[L(G,0,Z_i)] \end{aligned}$$

for a constant $c'' > 0$. Note that, by (8.12) and the case hypothesis that $\mathbb{E}[L(G,0,Z_i)] < c' \log^2 n$ we have that, for all $i < \tau$,

$$\mathbb{E}[L(G,0,Z_i)] \geq \frac{c'}{\lambda_i} \cdot T\Big(\Big\lfloor \frac{1}{2} \log^{\beta} n - 1 \Big\rfloor\Big)$$

Therefore,

$$T(\varphi) \geq c'c'' T\Big(\Big\lfloor \frac{1}{2} \log^{\beta} n - 1 \Big\rfloor\Big) \sum_{i<\tau} \frac{1}{\lambda_i}$$

Recall now that $\lambda_i = \mathbb{E}\left[|\Delta_0 \cap [\theta_i..n_i - 1]|\right]$, and note that, since $\beta \geq \delta$, each $d \in [0..n-1]$ belongs to at most two of the sets $[\theta_i..n_i - 1]$, for $i = 0, \ldots, \tau - 1$. Therefore, $\sum_{i<\tau} \lambda_i \leq 2\,\mathbb{E}[|\Delta_0|] = 2$. Because of the convexity of $1/x$, the last result yields

$$\sum_{i<\tau} \frac{1}{\lambda_i} \geq \frac{1}{2}\tau^2 \geq \frac{1}{2}\Big(\frac{\log n}{(\beta+\gamma)\log\log n} - \mathrm{O}(1)\Big)^2$$

by (8.14). Thus,

$$T(\varphi) \geq c_2 \Big(\frac{\log n}{\log\log n}\Big)^2 \cdot T\Big(\Big\lfloor \frac{1}{2} \log^{\beta} n - 1 \Big\rfloor\Big) \tag{8.16}$$

for some constant $c_2 > 0$. {end of Case 2}

Since (8.15) and (8.16) hold for all $\varphi \in \mathbf{\Phi}_{n,1}$, we have that, for all large enough n,

$$T(n) \geq \min\left\{c_1 \log^2 n,\ g(n) \cdot T\big(\alpha(n)\big)\right\} \tag{8.17}$$

where

$$g(n) = c_2 \frac{\log^2 n}{\log^2 \log n} \qquad \text{and} \qquad \alpha(n) = \left\lfloor \frac{1}{2} \log^\beta n - 1 \right\rfloor$$

We can now obtain the desired bound for $T(n)$ as follows. Let $\hat{n}$ be a large enough constant such that (8.17) holds for all $n \geq \hat{n}$. Let also κ be such that

$$\alpha^{(\kappa+1)}(n) < \hat{n} \leq \alpha^{(\kappa)}(n)$$

(By $f^{(k)}(x)$ we denote the function $f(x)$ iteratively applied $k \geq 0$ times to an initial value of x.) We recursively apply (8.17) until some of the following two conditions is met:

(i) in some recursive step the first argument of $\min\{\cdot, \cdot\}$ is smaller or equal to the second;

(ii) κ' steps have been performed, for some $\kappa' \leq \kappa$ that we specify later.

So, there are $\kappa' + 1$ possible cases: one for each of the κ steps in which condition (i) may be satisfied, and the case where condition (i) is never satisfied. The corresponding bounds obtained for $T(n)$ are $T_0, \ldots, T_{\kappa'}$, where

$$T_k = \begin{cases} \left(\prod_{i<k} g(\alpha^{(i)}(n))\right) \cdot c_1 \log^2(\alpha^{(k)}(n)), & \text{if } 0 \leq k < \kappa' \\ \left(\prod_{i<\kappa} g(\alpha^{(i)}(n))\right) \cdot \frac{2}{3}, & \text{if } k = \kappa' \end{cases}$$

(To obtain $T_{\kappa'}$ we used the trivial fact that $T(m) \geq \frac{2}{3}$, for $m \geq 3$.) Thus,

$$T(n) \geq \min\{T_0, \ldots, T_{\kappa'}\} \tag{8.18}$$

Note that $\alpha(n) \geq \log n$, for all $n \geq 8$; so, if we stipulate that $\hat{n} \geq 8$ we have that, for all $k \leq \kappa$,

$$\alpha^{(k)}(n) \geq \log^{(k)} n$$

Thus, for every $k \in [0..\kappa' - 1]$,

$$T_k \geq \left(\prod_{i<k} g(\log^{(i)}(n))\right) \cdot c_1 \log^2(\log^{(k)}(n)) = c_1 c_2^k \log^2 n$$

and, similarly,

$$T_{\kappa'} \geq \frac{2}{3} c_2^{\kappa'} \frac{\log^2 n}{(\log^{(\kappa'+1)} n)^2}$$

By letting

$$\kappa' = \min\{\kappa, \log^*(n) - 1\}$$

we obtain that, for all $k \in [0..\kappa']$,

$$T_k \geq c_3 \frac{\log^2 n}{a^{\log^* n}} \tag{8.19}$$

for some constants $c_3 > 0$ and $a > 1$. (Note that

$$\log^{(\kappa'+1)} n \le \max\{\log^{(\kappa+1)} n,\, \log^{(\log^*(n))} n\} \le \max\{\alpha^{(\kappa+1)}(n), 1\} \le \hat{n}.)$$

Combining (8.18) and (8.19), yields $T(n) = \Omega(\log^2 n / a^{\log^* n})$.

Proof of Lemma 8.10(a)

Recall that G is a randomly generated graph in $\mathcal{G}(\varphi)$, where $\varphi \in \mathbf{\Phi}_{n,1}$, and Z is the r-ancestor of a uniformly-random node in $[0..m-1]$.

Let $G^\star$ be the m-node augmented ring such that, for each $k \in [0..m-1]$, the delta-set of node k of $G^\star$ is

$$\Delta^\star_k = \Delta_k \cap [0..m-1]$$

(Δ_k is the delta-set of node k in G.) Note that the routing tree of $G^\star$ is identical to the subgraph of the routing tree of G induced by the nodes in $[0..m-1]$. So, Z can be equivalently viewed as the r-ancestor of a uniformly-random node in $G^\star$, and $\mathbb{E}[L(G, 0, Z)] = \mathbb{E}[L(G^\star, 0, Z)]$.

The proof proceeds roughly as follows. Based on G, we describe the construction of an m'-node augmented ring G', where $m' \approx m/(r+\theta)$. Roughly speaking, G' is a "scaled-down" version $G^\star$. We then compute a lower bound for $\mathbb{E}[L(G^\star, 0, Z)]$ in terms of $\mathbb{E}[L(G')]$. To make the lengths of the routing paths in G' comparable to those in $G^\star$ we assign to each node u of G' a positive *weight* W_u, and consider, instead, the "weighted" length of each routing path in G' — i.e., the sum of the weights of the nodes along this path. We show that $\mathbb{E}[L(G^\star, 0, Z)]$ is bounded from below by the expected weighted length of the routing paths in G', and then we bound the latter in terms of $\mathbb{E}[L(G')]$. We complete the proof by deriving a lower bound for $\mathbb{E}[L(G')]$ in terms of $T(m')$.

Before we describe how graph G' and the node weights $W_0, \dots, W_{m'-1}$ are generated in terms of $G^\star$, we describe the distribution they will have. We also compute some quantities related to this distribution. The size of G' is

$$m' = \left\lfloor \frac{m}{r'} \right\rfloor, \qquad \text{where } r' = r + \theta$$

We require that $m \ge 3r'$, which is true when, say,

$$\frac{m}{r} \ge b = 6$$

For each $u \in [0..m'-1]$, let Δ'_u denote the delta-set of node u of G'. The pairs $\langle \Delta'_0, W_0\rangle, \ldots, \langle \Delta'_{m'-1}, W_{m'-1}\rangle$ are mutually independent and they have the same distribution, which is roughly as follows. Recall from Section 8.6 that the r-path of node 0 in $G^\star$ is the routing path from 0 to its largest r-descendant, $r-1$. Consider the prefix of this path until we reach a node Y such that either (i) some of the long-range contacts of Y is at ring distance $\geq \theta$ from Y, or (ii) some of the neighbors of Y (successor or long-range contact) is $\geq r$. Let W be the length of this prefix, and D be a "scaled-down" version of $\Delta^\star_Y$ (which we describe later). Then, for each u, $\langle \Delta'_u, W_u\rangle$ has the same distribution as $\langle D, W\rangle$.

More precisely, let $\langle X_1, X_2, \ldots\rangle$ denote r-path of 0 in $G^\star$. We will assume that the sequence of X_t is infinite, and that $X_t = r-1$ for all t after the destination $r-1$ is reached. Let τ_1 be the earliest step when a node with some long-range contact at a ring distance $\geq \theta$ is encountered; i.e.,

$$\tau_1 = \min\left\{t \,:\, \Delta^*_{X_t} \cap [\theta..m-1] \neq \emptyset\right\}$$

Let also τ_2 be the earliest step when a node with a neighbor that is $\geq r$ is reached; i.e.,

$$\tau_2 = \min\left\{t \,:\, (\Delta^\star_{X_t} \cup \{1\}) \cap [r - X_t..m-1] \neq \emptyset\right\}$$

We define

$$W = \min\{\tau_1, \tau_2\} \quad \text{and} \quad D = \Psi(\Delta^\star_{X_W})$$

where

$$\Psi(\Delta) = [2..m'-1] \cap \bigcup_{d \in \Delta \cap [\theta..m-1]} \left\{\left\lfloor \frac{d}{r'} \right\rfloor + \delta \,:\, \delta = -1, 0, 1, 2\right\}$$

Note that $\Psi(\Delta) = \emptyset$ iff $\Delta \cap [\theta..m-1] = \emptyset$; and $\Delta^\star_{X_W} \cap [\theta..m-1] = \emptyset$ iff $\tau_1 > \tau_2$. So,

$$D = \begin{cases} \Psi(\Delta^\star_{X_{\tau_1}}) \neq \emptyset, & \text{if } \tau_1 \leq \tau_2 \\ \emptyset, & \text{if } \tau_1 > \tau_2 \end{cases}$$

We denote the joint distribution of $\langle D, W\rangle$ by $\hat{\varphi}$, and the marginal distribution of D by φ'. Note that $\varphi' \in \mathbf{\Phi}_{m',\ell'}$, where

$$\ell' = \mathbb{E}[|D|]$$

For each $u \in [0..m-1]$, $\hat{\varphi}$ will be the joint distribution of $\langle \Delta'_u, W_u\rangle$ (and φ' the marginal of Δ'_u).

Claim 8.11, below, provides an upper bound for ℓ', and a lower bound for the conditional expectation of W given D. Recall that λ is the expected number of long-range contacts a node in $\mathcal{G}(\varphi)$ (or, equivalently, in $G^\star$) has at ring distances between θ and $m-1$. Let π be the probability a node has at least one such long-range contact; i.e.,

$$\pi = \mathbb{Pr}\left[\Delta_0^* \cap [\theta..m-1] \neq \emptyset\right]$$

(Note that $\lambda \geq \pi$.)

Claim 8.11.

(a) If $\pi \neq 0$ then $\ell' \leq \dfrac{4\lambda}{\pi}$.

(b) $\mathbb{E}[W \mid D] \geq c_1 \min\left\{\dfrac{r}{\theta-1}, \dfrac{1}{\pi}\right\}$, for some constant $c_1 > 0$.

Proof. For part (a) we have

$$\begin{aligned}
\ell' &= \mathbb{E}\left[|\Psi(\Delta^\star_{X_{\tau_1}})| \;\middle|\; \tau_1 \leq \tau_2\right] \cdot \mathbb{Pr}[\tau_1 \leq \tau_2] \\
&\leq \mathbb{E}\left[|\Psi(\Delta^\star_{X_{\tau_1}})| \;\middle|\; \tau_1 \leq \tau_2\right] \\
&= \mathbb{E}\left[|\Psi(\Delta^\star_0)| \;\middle|\; \Delta^\star_0 \cap [\theta..m-1] \neq \emptyset\right] \\
&= \frac{\mathbb{E}[|\Psi(\Delta^\star_0)|]}{\mathbb{Pr}[\Delta^\star_0 \cap [\theta..m-1] \neq \emptyset]} \\
&= \frac{1}{\pi} \cdot \mathbb{E}[|\Psi(\Delta^\star_0)|] \qquad (8.20)
\end{aligned}$$

where the second-to-last equality holds because $\Psi(\Delta^\star_0) = \emptyset$ if $\Delta^\star_0 \cap [\theta..m-1] = \emptyset$. By the definition of Ψ,

$$|\Psi(\Delta^\star_0)| \leq 4|\Delta^\star_0 \cap [\theta..m-1]|$$

so,

$$\mathbb{E}[|\Psi(\Delta^\star_0)|] \leq 4\,\mathbb{E}[|\Delta^\star_0 \cap [\theta..m-1]|] = 4\lambda$$

Applying this to (8.20) yields $\ell' \leq 4\lambda/\pi$.

We now proceed to part (b). Let

$$\rho = \left\lceil \frac{r}{\theta-1} \right\rceil$$

For the case $D = \emptyset$ we have

$$\mathbb{E}[W \mid D = \emptyset] = \mathbb{E}[\tau_2 \mid \tau_1 > \tau_2] \geq \rho \qquad (8.21)$$

where the first relation holds because $D = \emptyset$ iff $\tau_1 > \tau_2$; and the second relation holds because if $\tau_1 > \tau_2$ then $X_{t+1} - X_t \leq \theta - 1$ for all $t \in [1..\tau_2]$, and $r - X_{\tau_2} \leq \theta - 1$, and, thus, $\tau_2 \geq r/(\theta-1)$.

Next, we consider the case $D = \Delta$, for some $\Delta \neq \emptyset$. We have

$$\begin{aligned}\mathbb{E}[W \mid D = \Delta] = \mathbb{E}[W \mid D \neq \emptyset] &= \mathbb{E}[\tau_1 \mid \tau_1 \leq \tau_2] \\ &= \mathbb{E}[\tau_1 \mid \tau_1 \leq \min\{\rho, \tau_2\}] \cdot \mathbb{Pr}[\tau_1 \leq \min\{\rho, \tau_2\}] \\ &\quad + \mathbb{E}[\tau_1 \mid \rho < \tau_1 \leq \tau_2] \cdot \mathbb{Pr}[\rho < \tau_1 \leq \tau_2]\end{aligned}$$

Since $\mathbb{E}[\tau_1 \mid \tau_1 \leq \min\{\rho, \tau_2\}] \leq \rho < \mathbb{E}[\tau_1 \mid \rho < \tau_1 \leq \tau_2]$,

$$\mathbb{E}[W \mid D = \Delta] \geq \mathbb{E}[\tau_1 \mid \tau_1 \leq \min\{\rho, \tau_2\}] = \mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] \tag{8.22}$$

where the second relation holds because if $\tau_1 \leq \rho$ then $\tau_1 \leq \tau_2$ (since $\tau_1 > \tau_2$ yields $\tau_1 > \tau_2 \geq \rho$, as we argued earlier). We have

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] = \sum_{j=1}^{\rho} j\, \mathbb{Pr}[\tau_1 = j \mid \tau_1 \leq \rho] = \sum_{j=1}^{\rho} j \frac{\mathbb{Pr}[\tau_1 = j]}{\mathbb{Pr}[\tau_1 \leq \rho]} = \sum_{j=1}^{\rho} j \frac{\pi q^{j-1}}{1 - q^{\rho}}$$

where $q = 1 - \pi$. After some computations we obtain

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] = \frac{1 - q^{\rho}(1 + \pi\rho)}{\pi(1 - q^{\rho})} \tag{8.23}$$

Since $(1 - q^{\rho}) \leq 1$ and $q^{\rho} \leq e^{-\pi\rho}$,

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] \geq \frac{1 - e^{-\pi\rho}(1 + \pi\rho)}{\pi} \geq \frac{0.18}{\pi}, \qquad \text{if } \pi\rho \geq 0.78 \tag{8.24}$$

For smaller values of $\pi\rho$ we obtain a lower bound for $\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho]$ as follows. By dividing both the numerator and denominator on the right-hand side of (8.23) by $(1 - q^{\rho})$ we get

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] = \frac{1 + \pi\rho - \pi\rho(1 - q^{\rho})^{-1}}{\pi}$$

and since, for $\pi\rho < 1$, $q^{\rho} \leq 1 - \pi\rho + \frac{1}{2}(\pi\rho)^2$,

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] \geq \frac{1 + \pi\rho - \pi\rho(\pi\rho - \frac{1}{2}(\pi\rho)^2)^{-1}}{\pi} = \rho\frac{1 - \pi\rho}{2 - \pi\rho} \geq 0.18\rho, \quad \text{if } \pi\rho \leq 0.78 \tag{8.25}$$

By (8.24) and (8.25), for all values of $\pi\rho$,

$$\mathbb{E}[\tau_1 \mid \tau_1 \leq \rho] \geq 0.18 \min\left\{\rho, \frac{1}{\pi}\right\}$$

So, by (8.22),

$$\mathbb{E}[W \mid D = \Delta] \geq 0.18 \min\left\{\rho, \frac{1}{\pi}\right\}$$

Combining this and (8.21) yields part (b) of the claim. ■

We now describe how we generate G' and $W_0, W_1, \ldots, W_{m'-1}$ from $G^\star$. The construction has similarities to that we used in the proof of Lemma 8.3. We denote by R and R' the routing trees of $G^\star$ and G', respectively.

We begin with an informal exposition. We generate the pairs $\langle \Delta'_u, W_u\rangle$ inductively by considering each node $u = 0, \ldots, m'-1$ in turn. If u is a leaf of R', we choose $\langle \Delta'_u, W_u\rangle$ independently at random from $\hat{\varphi}$. With each u that is not a leaf we associate a node C_u of R. As in the construction in the proof of Lemma 8.3, $C_0 = 0$, and for $u \neq 0$, C_u is determined when the delta-set of the parent of u in R' is determined. Δ'_u and W_u are defined similarly to D and W, using the r-path of C_u in $G^\star$ instead of the r-path of 0. Let Y_u be the node in the r-path of C_u whose delta-set is used to compute Δ'_u. With each non-leaf child $u+\delta$ of u in R' we associate a distinct child $C_{u+\delta}$ of Y_u in R, such that $C_{u+\delta} \approx Y_u + \delta r'$.

We now give a formal description of the construction. For each $k \in [0..m-1]$, let A_k be the r-ancestor of node k in $G^\star$, and

$$B_k = \begin{cases} A_k, & \text{if } A_k = k \\ A_k + r', & \text{otherwise} \end{cases}$$

So, if k is an r-significant node of $G^\star$ then $B_k = k$; otherwise, $B_k = A_k + r' > A_k + r > k > A_k$. For each $u = 0, \ldots, m'-1$, $\langle \Delta'_u, W_u\rangle$ is defined as follows. Let S'_u be the size of the subtree of R' rooted at u.

∘ If $S'_u > 1$ then

$$\Delta'_u = \Psi(\Delta^\star_{Y_u}) \quad \text{and} \quad W_u = L(G^\star, C_u, Y_u) + 1$$

where:

- Y_u is the first node in the r-path of C_u in $G^\star$ such that some of the following conditions holds: (i) $\Delta^\star_{Y_u} \cap [\theta..m-1] \neq \emptyset$, or (ii) a child of Y_u in R is $\geq C_u + r$.
- $C_u = 0$, if $u = 0$; and if $u > 0$, C_u is the largest child of Y_{F_u} in R that is $\leq B_{C_{F_u}} + (u - F_u) \cdot r'$, where F_u is u's parent in R'.

∘ If $S'_u = 1$ we choose $\langle \Delta'_u, W_u\rangle$ independently at random according to $\hat{\varphi}$.

Note that, given G, the above construction is "almost" deterministic: $R' = R'(R)$ and for every node u that is not a leaf of R', $\Delta'_u = \Delta'_u(G^\star)$ and $W_u = W_u(G^\star)$. The only randomness introduced (other than the choice of $G^\star$) is in the selection of the delta-sets and weights of the leaves of R'.

The next result is the analogue of Claims 8.7 and 8.8. (The proof is omitted.) For $k < m$, we denote by S_k the size of the subtree of R rooted at k.

Claim 8.12. *For all $u \in [0..m'-1]$, if $S'_u > 1$ then*

(a) $0 \le C_u \le Y_u < C_u + r$ *and* $B_{C_u} \le ur'$.

(b) $B_{C_u} + S'_u r' \le C_u + S_{C_u} = Y_u + S_{Y_u}$.

(c) For all $u' \in [u+1..m-1]$,

- *If* $u' < u + S'_u$ *and* $S'_{u'} > 1$ *then* $C_{u'} \in \langle\langle Y_u, S_{Y_u} \rangle\rangle - \{Y_u\}$.
- *If* $u' \ge u + S'_u$ *and* $S'_{u'} > 1$ *then* $C_{u'} \ge Y_u + S_{Y_u}$.

We now show that G' and $W_0, \ldots, W_{m'-1}$ have the distribution we described at the beginning of this proof; i.e., $\langle \Delta'_0, W_0 \rangle, \ldots, \langle \Delta'_{m'-1}, W_{m'-1} \rangle$ are mutually independent, and each pair has distribution $\hat{\varphi}$. As in the proof of Lemma 8.3, we think of the construction as a random m'-step process, where in step t we determine the value of $\langle \Delta'_t, W_t \rangle$, and each $\Delta^\star_k$ is generated right before it is about to be used for the first time — not earlier. Let U_t be the set of nodes k for which $\Delta^\star_k$ is generated in some of the steps $0, \ldots, t-1$. Note that U_t consists of the nodes in the routing paths of $G^\star$ from C_u to Y_u, for all $u < t$ such that $S'_u > 1$. So, by Claim 8.12(c), if $S'_t > 1$ then $k \notin U_t$, for all $k \ge C_t$, and, thus, the delta-sets of the nodes in the path from C_t to Y_t are generated in step t. It is now easy to see that $\langle \Delta'_t, W_t \rangle$ is constructed independently of choices made in previous steps, and has distribution $\hat{\varphi}$.

Next we bound the expected "weighted" delivery time in G' in terms of the corresponding "unweighted" quantity. For each $u, v \in [0..m'-1]$, let

$$L^w(u,v) = \sum_{u' \in p_{u,v}} W_{u'}$$

where $p_{u,v}$ is the set of all the nodes in the routing path from u to v in G', *excluding the last node* v. Let also

$$L^w = \frac{1}{m'} \sum_{u=0}^{m'-1} L^w(0,u)$$

So, $L^w(u,v)$ and L^w are the weighted versions of $L(G',u,v)$ and $L(G')$, respectively. Let H be any (fixed) possible instance of $\mathcal{G}(\varphi')$, and let $p^H_{0,u}, {\Delta'_v}^H$ be the corresponding values of $p_{0,u}, \Delta'_v$. Then,

$$\mathbb{E}\left[L^w(0,u) \mid G' = H\right] = \mathbb{E}\Big[\sum_{v \in p_{0,u}} W_v \;\Big|\; G' = H\Big] = \sum_{v \in p^H_{0,u}} \mathbb{E}[W_v \mid G' = H]$$

$$= \sum_{v \in p^H_{0,u}} \mathbb{E}\left[W \mid D = {\Delta'_v}^H\right]$$

where the last relation holds because of the independence of the $\langle \Delta'_i, W_i\rangle$ pairs, and the fact that they are distributed like $\langle D, W\rangle$. By Claim 8.11(b) then,

$$\mathbb{E}[L^w(0,u) \mid G' = H] \geq c_1 \min\left\{\frac{r}{\theta-1}, \frac{1}{\pi}\right\} \cdot |p^H_{0,u}| = c_1 \min\left\{\frac{r}{\theta-1}, \frac{1}{\pi}\right\} \cdot L(H,0,u)$$

Taking the average over all $u < m'$, and the expectation over all H, yields

$$\mathbb{E}[L^w] \geq c_1 \min\left\{\frac{r}{\theta-1}, \frac{1}{\pi}\right\} \cdot \mathbb{E}[L(G')] \tag{8.26}$$

Now, since G' is a random graph in $\mathcal{G}(\varphi')$, and $\varphi' \in \mathbf{\Phi}_{m',\ell'}$, $\mathbb{E}[L(G')] \geq T(m', \ell')$. So, if $\pi > 0$ then, by Claim 8.11(a) and Lemma 8.1,

$$\mathbb{E}[L(G')] \geq T\Big(m', \min\left\{\frac{4\lambda}{\pi}, m'-2\right\}\Big)$$

and by applying Lemma 8.3 to the right-hand side, we obtain

$$\mathbb{E}[L(G')] \geq \frac{\pi}{4\lambda} \cdot T(m')$$

Using the above and the trivial fact that $T(i) \geq 2/3$, for all $i \geq 3$, (8.26) yields

$$\mathbb{E}[L^w] \geq c_2 \min\left\{\frac{r}{\theta-1}, \frac{1}{\lambda} \cdot T(m')\right\} \tag{8.27}$$

for some constant $c_2 > 0$.

The last piece of the proof is to bound $\mathbb{E}[L(G^\star, 0, Z)]$ in terms of $\mathbb{E}[L^w]$. We use the following result.

Claim 8.13. *Let H be a (fixed) possible instance of $G^\star$ and $u \in [0..m'-1]$ be such that if $G^\star = H$ then u is not a leaf of R'. Then,*

$$r' \cdot \mathbb{E}\Big[\sum_{v \in \langle\!\langle u, S'_u \rangle\!\rangle} L^w(u,v) \;\Big|\; G^\star = H\Big] \leq \sum_{k \in \langle\!\langle B_{C_u}, r'S'_u \rangle\!\rangle} L(H, C_u, A_k)$$

(The proof is by induction on S'_u; the details of the proof are omitted.)

By Claim 8.13, applied for $u = 0$,

$$\sum_{k < r'm'} L(H, 0, A_k) \geq r' \, \mathbb{E}\Big[\sum_{v < m'} L^w(0,v) \;\Big|\; G^\star = H\Big]$$

and taking the expectation over all H, yields

$$\mathbb{E}\Big[\sum_{k<r'm'} L(G^\star, 0, A_k)\Big] \geq r'm' \, \mathbb{E}[L^w]$$

So,

$$\begin{aligned}\mathbb{E}[L(G^\star, 0, Z)] &= \frac{1}{m}\,\mathbb{E}\Big[\sum_{k<m} L(G^\star, 0, A_k)\Big] \geq \frac{1}{m}\,\mathbb{E}\Big[\sum_{k<r'm'} L(G^\star, 0, A_k)\Big] \\ &\geq \frac{r'm'}{m}\,\mathbb{E}[L^w] \geq \frac{2}{3}\,\mathbb{E}[L^w]\end{aligned}$$

where the last relation holds when $m/r' \geq 3r'$. Substituting $\mathbb{E}[L^w]$ above with the right-hand side of (8.27) yields the desired bound for $\mathbb{E}[L(G, 0, Z)] = \mathbb{E}[L(G^\star, 0, Z)]$.

Proof of Lemma 8.10(b)

Recall that $1 \leq \eta \leq r \leq m \leq n$, G is a randomly generated graph in $\mathcal{G}(\varphi)$, where $\varphi \in \mathbf{\Phi}_{n,1}$, and M is the number of nodes in $[0..m-1]$ that are r-descendants of Z in G, where Z is the r-ancestor of a uniformly-random node in $[0..m-1]$.

For each node u of G, let D_u be the number of nodes in $[0..m-1]$ that are r-descendants of the r-ancestor of u. (So, if u is an r-significant node then D_u is the number of *its* r-descendants that are in $[0..m-1]$.) Let S be the set of the r-significant nodes that are in $[0..m-1]$, and

$$S^* = \{u \in S \,:\, D_u = r\}$$

Then,

$$\begin{aligned}\mathbb{P}\mathrm{r}[M < \eta] &= \mathbb{E}\big[\mathbb{P}\mathrm{r}[M < \eta \mid G]\big] = \mathbb{E}\Big[\frac{1}{m}\cdot|\{u \,:\, D_u < \eta\}|\Big] \\ &\leq \mathbb{E}\Big[\frac{\eta-1}{m}\cdot|\{u \in S \,:\, D_u < \eta\}|\Big] \\ &\leq \frac{\eta-1}{m}\cdot\mathbb{E}\big[|S| - |S^*|\big] \qquad (8.28)\end{aligned}$$

We now describe an upper bound for $|S|$. For each node u, we define N_u as follows: if $u \in S^*$, N_u is the total number of long-range contacts of the nodes in the r-path of u (i.e., the path from u to its largest r-descendant $u + r - 1$ — see Section 8.6); if $u \notin S^*$, $N_u = 0$. Note that every node in $S - \{0\}$ is an r-successor of some node in S^*. Note also that if $u \in S^*$ then the number of its r-successors is at most equal to N_u, plus 1 (for the ring successor of $u + r - 1$). From these two observations it follows that

$$|S| - 1 \leq \sum_{u\in S^*}(N_u + 1) = |S^*| + \sum_u N_u$$

Combining this with (8.28) yields

$$\mathbb{P}\mathrm{r}[M < \eta] \leq \frac{\eta-1}{m}\Big(1 + \mathbb{E}\Big[\sum_u N_u\Big]\Big) \qquad (8.29)$$

In the rest of the proof we establish an upper bound for $\mathbb{E}[\sum_u N_u]$; we show it is at most $\frac{m}{r}(1+\mathbb{E}[L(G,0,r-1)])$. Roughly speaking, this is true because, for every $u \in S^*$, the expected value of N_u is $\mathbb{E}[L(G,0,r-1)]+1$, and $|S^*|$ is at most m/r. Combining this bound for $\mathbb{E}[\sum_u N_u]$ and (8.29) yields the desired bound for $\mathbb{Pr}[M < \eta]$. The next claim computes the expected value of N_u. For each node u, let Δ_u denote the delta-set of u. Note that, by Lemma 8.6, $\Delta_0, \ldots, \Delta_{u-1}$ completely determine which of the nodes in $[0..u]$ are in S^*.

Claim 8.14. *If $u \in S^*$ then $\mathbb{E}[N_u \mid \Delta_0, \ldots, \Delta_{u-1}] = \mathbb{E}[L(G,0,r-1)]+1$.*

Proof. Conditioned on the event "$u \in S^*$," N_u is independent of $\Delta_0, \ldots, \Delta_{u-1}$ (since $\Delta_u, \ldots, \Delta_{u+r-1}$ are independent of $\Delta_0, \ldots, \Delta_{u-1}$). Also, the conditional distribution of N_u given $u \in S^*$ does not depend on the value of u (since $\Delta_u, \ldots, \Delta_{u+r-1}$ are independent, each with distribution φ). Thus,

$$\mathbb{E}[N_u \mid \Delta_0, \ldots, \Delta_{u-1}] = \mathbb{E}[N_0]$$

Let $\langle v_0, v_1, \ldots, v_\sigma \rangle$ be the r-path of node 0. For each $j \geq 0$, we define K_j as follows: if $j \leq \sigma$, K_j is the number of long-range contacts of v_j; if $j > \sigma$, K_j is chosen independently at random according the distribution of $|\Delta_0|$. Let A be the set of all possible values of σ. Note that

$$\mathbb{E}[K_j \mid \sigma = i] = \mathbb{E}[|\Delta_0|], \qquad \text{for } j > i \in A \tag{8.30}$$

Also,

$$\mathbb{E}[K_j \mid \sigma \geq j] = \mathbb{E}[|\Delta_0|], \qquad \text{for } 0 \leq j \leq \max A$$

Combining the above two results yields that, for all $j \geq 0$,

$$\mathbb{E}[K_j] = \mathbb{E}[|\Delta_0|] \tag{8.31}$$

Now, we have

$$\begin{aligned}
\mathbb{E}[N_0] &= \sum_{i \in A} \Big(\mathbb{E}[N_0 \mid \sigma = i] \cdot \mathbb{Pr}[\sigma = i] \Big) \\
&= \sum_{i \in A} \sum_{j=0}^{i} \Big(\mathbb{E}[K_j \mid \sigma = i] \cdot \mathbb{Pr}[\sigma = i] \Big) \\
&= \sum_{j \geq 0} \Big(\mathbb{E}[K_j] - \sum_{i \in A \,:\, i<j} \big(\mathbb{E}[K_j \mid \sigma = i] \cdot \mathbb{Pr}[\sigma = i] \big) \Big) \\
&= \sum_{j \geq 0} \Big(\mathbb{E}[|\Delta_0|] - \mathbb{E}[|\Delta_0|] \cdot \mathbb{Pr}[\sigma < j] \Big) \\
&= \sum_{j \geq 0} (1 - \mathbb{Pr}[\sigma < j])
\end{aligned}$$

where the second-to-last line is obtained using (8.31) and (8.30), and the last line holds because $\mathbb{E}[|\Delta_0|] = 1$. Therefore, by letting $j' = j - 1$, we obtain

$$\mathbb{E}[N_0] = 1 + \sum_{j' \geq 0} (1 - \mathbb{Pr}[\sigma \leq j']) = 1 + \mathbb{E}[\sigma] = 1 + \mathbb{E}[L(G, 0, r-1)] \qquad \blacksquare$$

We now derive the upper bound for $\mathbb{E}[\sum_u N_u]$. We do that by describing a sequence $X_0, \ldots, X_m$ of progressively more accurate estimates of $\sum_u N_u$, such that $X_m \geq \sum_u N_u$, $X_0 = \frac{m}{r}(\mathbb{E}[L(G, 0, r-1)] + 1)$, and, for all $t < m$, $\mathbb{E}[X_{t+1}] \leq \mathbb{E}[X_t]$. We define the sequence if X_t as follows. For each $t \in [0..m]$,

$$X_t = \sum_{0 \leq u < t} N_u + \frac{m - Z_t}{r}\big(1 + \mathbb{E}[L(G, 0, r-1)]\big)$$

where

$$Z_t = \begin{cases} 0, & \text{if } t = 0 \\ \max\big\{u \in S^* \;:\; u < t\big\} + r, & \text{if } t > 0 \end{cases}$$

For all $t \in [0..m-1]$, we then have

$$X_{t+1} - X_t = \begin{cases} 0, & \text{if } t \notin S^* \\ N_t - \frac{Z_t - Z_{t+1}}{r}(1 + \mathbb{E}[L(G, 0, r-1)]) & \text{if } t \in S^* \end{cases}$$

Note that if $t \in S^*$ then $Z_t - Z_{t+1} \geq r$, and, by Claim 8.14, $\mathbb{E}[N_t \mid \Delta_0, \ldots, \Delta_{t-1}] = 1 + \mathbb{E}[L(G, 0, r-1)]$. So, if $t \in S^*$,

$$\mathbb{E}[X_{t+1} - X_t \mid \Delta_0, \ldots, \Delta_{t-1}] \leq 0$$

The above also holds (as equality) if $t \notin S^*$. Therefore, for all $t \in [0..m-1]$, $\mathbb{E}[X_{t+1}] \leq \mathbb{E}[X_1]$, which implies that $\mathbb{E}[X_m] \leq \mathbb{E}[X_0]$. Substituting the values of X_m and X_0, we obtain

$$\mathbb{E}\Big[\sum_u N_u + \frac{m - Z_m}{r}(1 + \mathbb{E}[L(G, 0, r-1)])\Big] \leq \frac{m}{r}(1 + \mathbb{E}[L(G, 0, r-1)])$$

and, since $Z_m \leq m$, we have

$$\mathbb{E}\Big[\sum_u N_u\Big] \leq \frac{m}{r}(1 + \mathbb{E}[L(G, 0, r-1)])$$

This, together with (8.29), yields

$$\mathbb{Pr}[M < \eta] \leq \frac{\eta - 1}{m}\Big(1 + \frac{m}{r}(1 + \mathbb{E}[L(G, 0, r-1)])\Big) = \frac{\eta - 1}{r}\Big(\frac{r}{m} + 1 + \mathbb{E}[L(G, 0, r-1)])\Big)$$

which yields the desired bound for $\mathbb{Pr}[M \geq \eta]$, since $r \leq m$.

Chapter 9

Concluding remarks and future work

We conclude with a brief summary of our results and an outline of some open research problems that are closely related to them.

9.1 Adversarial load balancing in DHTs

We proposed the first key-space partitioning scheme for DHTs that provably maintains bounded ratio of largest to smallest block sizes, in the face of *adversarial* node arrivals and departures. All other key-space partitioning schemes that have been proposed so far rely on the assumption that either there are no departures, or that nodes leave the system randomly.

Our scheme requires $\Theta(R \log n)$ messages per arrival and departure of a node, in an n-node system where routing requires R messages. It would be interesting to investigate whether it is possible to achieve load balancing against an adversary using a more efficient key-space partitioning scheme, or if our scheme is (asymptotically) optimal in terms of message complexity. Note that the most efficient key-space partitioning scheme for a non-adversarial setting that has been proposed so far ([50]) requires only $\Theta(R + \log n)$ messages per arrival and $\Theta(\log n)$ messages per departure.

9.2 Complexity of greedy routing in augmented grids

We have shown that the expected number of steps for greedy routing in augmented rings of n nodes with on average ℓ long-range contacts per node is $\Omega((\log^2 n)/\ell a^{\log^* n})$. This improves a lower bound by Aspnes *et al.*, and shows that the combination of augmented rings and greedy routing cannot achieve an optimal tradeoff between degree and routing paths length, even when $\ell = \Theta(\log n)$, a case of practical interest.

Our lower bound is very close to the upper bound of $O((\log^2 n)/\ell)$ that greedy routing achieves in Kleinberg's (one-dimensional) small-world networks, a particular instance of augmented rings. Our analysis suggests that an (asymptotically) optimal distribution φ for choosing long-range contacts has structural properties similar to those of the distribution used in Kleinberg's construction. We conjecture that our lower bound can be improved to $\Omega((\log^2 n)/\ell)$, i.e., that Kleinberg's construction is in fact (asymptotically) optimal for greedy routing in augmented rings.

In our work we have focused on *unidirectional* greedy routing, where the distance from node u to node v is the number of edges along the ring in, say, clockwise direction from u to v. In *bidirectional* greedy routing, the distance between two nodes is the minimum number of ring edges between them in either clockwise or counterclockwise direction. In most actual designs, both versions of greedy routing give (asymptotically) the same results. We conjecture that the same asymptotic bounds apply to both versions.

The augmented ring model naturally generalizes to more than one dimensions, by using the d-dimensional torus (or grid) as a base graph instead of the ring. Kleinberg's construction also generalizes to higher dimensions resulting in the same $O((\log^2 n)/\ell)$ upper bound for greedy routing. It is interesting to investigate whether the use of additional dimensions improves the performance of greedy routing or if a lower bound similar to that for the one-dimensional case applies.

Bibliography

[1] Karl Aberer, Philippe Cudré-Mauroux, Anwitaman Datta, Zoran Despotovic, Manfred Hauswirth, Magdalena Punceva, and Roman Schmidt. P-Grid: A self-organizing structured P2P system. *SIGMOD Record*, 32(3):29–33, 2003.

[2] Ittai Abraham, Baruch Awerbuch, Yossi Azar, Yair Bartal, Dahlia Malkhi, and Elan Pavlov. A generic scheme for building overlay networks in adversarial scenarios. In *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, page 40.2, April 22–26 2003.

[3] Ittai Abraham, Dahlia Malkhi, and Gurmeet Singh Manku. Papillon: Greedy routing in rings. `http://arxiv.org/abs/cs/0507034`, 2005. See also *Proceedings of the 19th International Symposium on Distributed Computing (DISC 2005)*, pages 514–515, September 26–29 2005.

[4] Micah Adler, Eran Halperin, Richard Karp, and Vijay Vazirani. A stochastic process on the hypercube with applications to peer-to-peer networks. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC 2003)*, pages 575–584, June 9–11 2003.

[5] James Aspnes, Zoë Diamadi, and Gauri Shah. Greedy routing in peer-to-peer systems. `http://arxiv.org/abs/cs/0302022`, 2006.

[6] James Aspnes, Zoë Diamadi, and Gauri Shah. Fault-tolerant routing in peer-to-peer systems. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 223–232, July 21–24 2002.

[7] James Aspnes and Gauri Shah. Skip graphs. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pages 384–393, January 12–14 2003.

[8] Yossi Azar, Andrei Broder, Anna Karlin, and Eli Upfal. Balanced allocations. *SIAM Journal on Computing*, 29(1):180–200, 1999.

[9] Hari Balakrishnan, M. Frans Kaashoek, David R. Karger, Robert Morris, and Ion Stoica. Looking up data in P2P systems. *Communications of the ACM*, 46(2):43–48, 2003.

[10] Magdalena Balazinska, Hari Balakrishnan, and David Karger. INS/Twine: A scalable peer-to-peer architecture for intentional resource discovery. In *Proceedings of the 1st International Conference on Pervasive Computing*, pages 195–210, August 26–28 2002.

[11] Lali Barrière, Pierre Fraigniaud, Evangelos Kranakis, and Danny Krizanc. Efficient routing in networks with long range contacts. In *Proceedings of the 15th International Symposium on Distributed Computing (DISC 2001)*, pages 270–284, October 3–5 2001.

[12] Mayank Bawa, Gurmeet Singh Manku, and Prabhakar Raghavan. SETS: Search enhanced by topic segmentation. In *Proceedings of the 26th International ACM SIGIR Conference (SIGIR 2003)*, pages 306–313, July 28–August 1 2003.

[13] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Animesh Nandi, Antony Rowstron, and Atul Singh. SplitStream: High-bandwidth multicast in cooperative environments. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003)*, pages 298–313, October 19–22 2003.

[14] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*, pages 311–320, July 25–26 2000.

[15] Don Coppersmith, David Gamarnik, and Maxim Sviridenko. The diameter of a long range percolation graph. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, pages 329–337, January 6–8 2002.

[16] Frank Dabek, Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Wide-Area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, pages 202–215, October 21–24 2001.

[17] Frank Dabek, Jinyang Li, Emil Sit, James Robertson, M. Frans Kaashoek, and Robert Morris. Designing a DHT for low latency and high throughput. In *Proceedings of the 1st*

USENIX Symposium on Networked Systems Design and Implementation (NSDI 2004), pages 85–98, March 29–31 2004.

[18] Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Could any graph be turned into a small-world? *Theory of Computing Systems*, 355(1):96–103, 2006.

[19] Michele Flammini, Luca Moscardelli, Alfredo Navarra, and Stéphane Pérennes. Asymptotically optimal solutions for small world graphs. In *Proceedings of the 19th International Symposium on Distributed Computing (DISC 2005)*, pages 414–428, September 26-29 2005.

[20] Pierre Fraigniaud. Greedy routing in tree-decomposed graphs. In *Proceedings of the 13th European Symposium on Algorithms (ESA 2005)*, pages 791–802, October 3–6 2005.

[21] Pierre Fraigniaud. Small worlds as navigable augmented networks: Model, analysis, and validation. In *Proceedings of the 15th European Symposium on Algorithms (ESA 2007)*, pages 2–11, October 8–10 2007.

[22] Pierre Fraigniaud and Philippe Gauron. D2B: A de Bruijn based content-addressable network. *Theory of Computing Systems*, 355(1):65–79, 2006.

[23] Pierre Fraigniaud, Cyril Gavoille, Adrian Kosowski, Emmanuelle Lebhar, and Zvi Lotker. Universal augmentation schemes for network navigability: Overcoming the sqrt(n)-barrier. In *Proceedings of the 19th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2007)*, pages 1–7, June 9–11 2007.

[24] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC 2004)*, pages 169–178, July 25–28 2004.

[25] Michael Freedman, Eric Freudenthal, and David Mazières. Democratizing content publication with Coral. In *Proceedings of the 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004)*, pages 239–252, March 29–31 2004.

[26] Prasanna Ganesan and Gurmeet Singh Manku. Optimal routing in Chord. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, pages 176–185, January 11–14 2004.

[27] George Giakkoupis and Vassos Hadzilacos. A scheme for load balancing in heterogeneous distributed hash tables. In *Proceedings of the 24th ACM Symposium on Principles of Distributed Computing (PODC 2005)*, pages 302–311, July 17–20 2005.

[28] George Giakkoupis and Vassos Hadzilacos. On the complexity of greedy routing in ring-based peer-to-peer networks. In *Proceedings of the 26th ACM Symposium on Principles of Distributed Computing (PODC 2007)*, pages 99–108, August 12–15 2007.

[29] Krishna Gummadi, Ramakrishna Gummadi, Steven Gribble, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of the ACM SIGCOMM 2003 Conference*, pages 381–394, August 25–29 2003.

[30] Nicholas Harvey, Michael Jones, Stefan Saroiu, Marvin Theimer, and Alec Wolman. Skipnet: A scalable overlay network with practical locality properties. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)*, March 26–28 2003.

[31] Kirsten Hildrum and John Kubiatowicz amd Satosj Rap amd Ben Zhao. Distributed object location in a dynamic network. *Theory of Computing Systems*, 37(3):405–440, 2004.

[32] Ryan Huebsch, Joseph Hellerstein, Nick Lanham, Boon Thau Loo, Scott Shenker, and Ion Stoica. Querying the internet with PIER. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB 2003)*, pages 321–332, September 9–12 2003.

[33] Sitaram Iyer, Antony Rowstron, and Peter Druschel. Squirrel: A decentralized peer-to-peer web cache. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 213–222, July 21–24 2002.

[34] Frans Kaashoek and David Karger. Koorde: A simple degree-optimal hash table. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS 2003)*, pages 98–107, February 20–21 2003.

[35] David Karger, Eric Lehman, Tom Leighton, Matthew Levine, Daniel Lewin, and Rina Panigrahy. Consistent hashing and random trees: Distributed caching protocols for

relieving hot spots on the World Wide Web. In *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC 1997)*, pages 654–663, May 4–6 1997.

[36] David Karger and Matthias Ruhl. Simple efficient load balancing algorithms for peer-to-peer systems. In *Proceedings of the 16th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2004)*, pages 36–43, 2004.

[37] Krishnaram Kenthapadi and Gurmeet Singh Manku. Decentralized algorithms using both local and random probes for P2P load balancing. In *Proceedings of the 17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2005)*, pages 135–144, July 18–20 2005.

[38] Valerie King and Jared Saia. Choosing a random peer. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC 2004)*, pages 125–130, July 25–28 2004.

[39] Jon Kleinberg. The small-world phenomenon: An algorithm perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC 2000)*, pages 163–170, May 21–23 2000.

[40] Jon Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, pages 431–438, December 3–8 2001.

[41] Jon Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM)*, August 22–30 2006.

[42] Emmanuelle Lebhar and Nicolas Schabanel. Almost optimal decentralized routing in long-range contact networks. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, pages 894–905, July 12–16 2004.

[43] Frank Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, 1992.

[44] Xiaozhou Li, Jayadev Misra, and Greg Plaxton. Active and concurrent topology maintenance. In *Proceedings of the 18th International Symposium on Distributed Computing (DISC 2004)*, pages 320–334, October 4–7 2004.

[45] David Liben-Nowell, Hari Balakrishnan, and David Karger. Analysis of the evolution of peer-to-peer systems. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 233–242, July 21–24 2002.

[46] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the USA*, 102(33):11623–11628, August 2005.

[47] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7:72–93, 2005.

[48] Dahlia Malkhi, Moni Naor, and David Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, pages 183–192, July 21–24 2002.

[49] Gurmeet Singh Manku. Routing networks for DHTs. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 133–142, July 13–16 2003.

[50] Gurmeet Singh Manku. Balanced binary trees for ID management and load balance in distributed hash tables. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC 2004)*, pages 197–205, July 25–28 2004.

[51] Gurmeet Singh Manku. *Dipsea: A Modular Distributed Hash Table*. PhD thesis, Stanford University, Department of Computer Science, September 2004.

[52] Gurmeet Singh Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed hashing in a small world. In *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS 2003)*, pages 127–140, March 26–28 2003.

[53] Gurmeet Singh Manku, Moni Naor, and Udi Wieder. Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC 2004)*, pages 54–63, June 13–15 2004.

[54] Charles Martel and Van Nguyen. Analyzing Kleinberg's (and other) small-world models. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC 2004)*, pages 179–188, July 25–28 2004.

[55] Charles Martel and Van Nguyen. Analyzing and characterizing small-world graphs. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 311–320, January 23–25 2005.

[56] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pages 53–65, March 7–8 2002.

[57] Stanley Milgram. The small world problem. *Psychology Today*, 67(1):60–67, May 1967.

[58] Alan Mislove, Ansley Post, Charles Reis, Paul Willmann, Peter Druschel, Dan Wallach, Xavier Bonnaire, Pierre Sens, Jean-Michel Busca, and Luciana Bezerra Arantes. POST: A secure, resilient, cooperative messaging system. In *Proceedings of the 9th Workshop on Hot Topics in Operating Systems (HotOS 2003)*, pages 61–66, May 18–21 2003.

[59] Michael Mitzenmacher, Andrea Richa, and Sitaraman Sitaraman. The power of two random choices: A survey of techniques and results. *Handbook of Randomized Computing: vol. 1*, pages 255–312, June 2001.

[60] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[61] Athicha Muthitacharoen, Robert Morris, Thomer Gil, and Benjie Chen. Ivy: A read/write peer-to-peer file system. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, pages 31–44, December 9–11 2002.

[62] Moni Naor and Udi Wieder. Novel architectures for P2P applications: The continuous-discrete approach. In *Proceedings of the 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2003)*, pages 50–59, June 7–9 2003.

[63] Moni Naor and Udi Wieder. Novel architectures for P2P applications: The continuous-discrete approach. *ACM Transactions on Algorithms*, 3(3), 2007.

[64] Greg Plaxton, Rajmohan Rajaraman, and Andréa Richa. Accessing nearby copies of replicated objects in a distributed environment. *Theory of Computing Systems*, 32(3):241–280, 1999.

[65] William Pugh. Skip lists: A probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, June 1990.

[66] Venugopalan Ramasubramanian, Ryan Peterson, and Emin Gün Sirer. Corona: A high performance publish-subscribe system for the World Wide Web. In *Proceedings of the 3rd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2006)*, pages 15–28, May 8–10 2006.

[67] Venugopalan Ramasubramanian and Emin Gün Sirer. The design and implementation of a next generation name service for the internet. In *Proceedings of the ACM SIGCOMM 2004 Conference*, pages 331–342, August 30–September 3 2004.

[68] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable Content-Addressable Network. In *Proceedings of the ACM SIGCOMM 2001 Conference*, pages 161–172, August 27–31 2001.

[69] Sylvia Ratnasamy, Mark Handley, Richard Karp, and Scott Shenker. Application-level multicast using Content-Addressable Networks. In *Proceedings of the 3nd International Workshop on Networked Group Communication (NGC 2001)*, pages 14–29, November 7–9 2001.

[70] Sean Rhea, Patrick Eaton, Dennis Geels, Hakim Weatherspoon, Ben Zhao, and John Kubiatowicz. Pond: The OceanStore prototype. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST 2003)*, pages 1–14, March 31–2 2003.

[71] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a DHT. In *Proceedings of the USENIX Annual Technical Conference (USENIX 2004)*, June 27–July 31 2004.

[72] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (MIDDLEWARE 2001)*, pages 329–350, November 12–16 2001.

[73] Antony Rowstron and Peter Druschel. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the 18th ACM*

Symposium on Operating Systems Principles (SOSP 2001), pages 188–201, October 21–24 2001.

[74] Antony Rowstron, Anne-Marie Kermarrec, Miguel Castro, and Peter Druschel. Scribe: The design of a large-scale event notification infrastructure. In *Proceedings of the 3nd International Workshop on Networked Group Communication (NGC 2001)*, pages 30–43, November 7–9 2001.

[75] Aleksandrs Slivkins. Distance estimation and object location via rings of neighbors. In *Proceedings of the 24th ACM Symposium on Principles of Distributed Computing (PODC 2005)*, pages 41–50, July 17–20 2005.

[76] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet Indirection Infrastructure. In *Proceedings of the ACM SIGCOMM 2002 Conference*, pages 73–86, August 19–23 2002.

[77] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, February 2003.

[78] Jeremy Stribling, Jinyang Li, Isaac Councill, M. Frans Kaashoek, and Robert Morris. OverCite: A distributed, cooperative CiteSeer. In *Proceedings of the 3rd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2006)*, pages 143–153, May 8–10 2006.

[79] Hermann Thorisson. *Coupling, Stationarity, and Regeneration.* Springer, 2000.

[80] Duncan J. Watts, Peter Sheridan Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[81] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[82] Udi Wieder. *The Continuous-Discrete Approach for Designing P2P Networks and Algorithms.* PhD thesis, The Weizmann Institute of Science, Department of Computer Science and Applied Mathematics, August 10 2005.

[83] Jun Xu. On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks. In *Proceedings of the IEEE INFOCOM 2003 Conference*, March 30–April 3 2003.

[84] Hui Zhang, Ashish Goel, and Ramesh Govindan. Incrementally improving lookup latency in distributed hash table systems. In *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 114–125, June 11–14 2003.

[85] Shalley Zhuang, Ben Zhao, Anthony Joseph, Randy Katz, and John Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant widearea data dissemination. In *Proceedings of the 11nd International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001)*, pages 11–20, June 25–26 2001.