Randomized Diffusion for Indivisible Loads

Petra Berenbrink* Colin Cooper† Tom Friedetzky ‡ Tobias Friedrich§
Thomas Sauerwald¶

Abstract

We present a new randomized diffusion-based algorithm for balancing indivisible tasks (tokens) on a network. Our aim is to minimize the discrepancy between the maximum and minimum load. The algorithm works as follows. Every vertex distributes its tokens as evenly as possible among its neighbors and itself. If this is not possible without splitting some tokens, the vertex redistributes its excess tokens among all its neighbors randomly (without replacement).

In this paper we prove several upper bounds on the load discrepancy for general networks. These bounds depend on some expansion properties of the network, that is, the second largest eigenvalue, and a novel measure which we refer to as refined local divergence. We then apply these general bounds to obtain results for some specific networks. For constant-degree expanders and torus graphs, these yield exponential improvements on the discrepancy bounds compared to the algorithm of Rabani, Sinclair, and Wanka [14]. For hypercubes we obtain a polynomial improvement.

In contrast to previous papers, our algorithm is vertex-based and not edge-based. This means excess tokens are assigned to vertices instead to edges, and the vertex reallocates all of its excess tokens by itself. This approach avoids nodes having "negative loads" (like in [8, 10]), but causes additional dependencies for the analysis.

*School of Computing Science, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada, petra@cs.sfu.ca. Supported by an NSERC Discovery grant.

†Department of Computer Science, King's College London, London, WC2R 2LS, U.K., colin.cooper@kcl.ac.uk.

‡School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, U.K., tom.friedetzky@dur.ac.uk.Supported by EPSRC grant EP/E029124/1 and EU MCITN grant 238808.

§Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany, tobias.friedrich@mpi.inf.mpg.de.

¶Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany, Thomas.sauerwald@mpi.inf.mpg.de. Supported by a PIMS fellowship, part of this work was done while being at SFU, Canada.

1 Introduction

During the last years, large parallel networks became widely available for industrial and academic users. An important prerequisite for their efficient usage is to balance the workload efficiently. Load balancing also has applications in scheduling, routing, numerical computation, and finite element computations.

In this paper we analyze a very simple neighborhood-based load balancing algorithm. We assume that the processors are connected by an arbitrary d-regular network. In the beginning, every vertex has a certain number of tokens (load). The goal is to distribute the tokens as evenly as possible. More precisely, we aim at minimizing the difference between the minimum load and the maximum load, which we call *discrepancy*.

In general, neighborhood-based load balancing algorithms operate in parallel steps (sometimes also called rounds). In each step, every processor is allowed to probe the load of all of its neighbors (*diffusion load balancing*), or to probe the load of one neighbor (*dimension exchange*). Then each processor decides how much load it will forward to its neighbors. In this paper, we consider a very natural diffusion-based approach. In the *continuous* diffusion model, where tokens can be split arbitrarily, the method works as follows. Along each edge a load of *load-difference*$/(d+1)$ is sent from the vertex with the higher load to the vertex with less tokens. Note that this method balances the load perfectly if the number of steps is sufficiently large. Here we consider the (arguably more realistic [15]) case of *discrete* diffusion where tokens are indivisible. Quantifying by how much the integrality assumption decreases the efficiency of load balancing is an interesting question and has been posed by many authors (e.g., [8, 11–15]).

Most results known so far ([9, 10, 14]) employ an *edge-oriented* view where each edge decides between forwarding either $\lceil$*load-difference*$/(d+1)\rceil$ or $\lfloor$*load-difference*$/(d+1)\rfloor$ tokens (referred to as rounding up or rounding down). Rounding up results in a load balancing algorithm that keeps sending tokens back and forth between processors with a small load difference. Another disadvantage is that the approach can generate "negative loads" for vertices with only a few tokens. On the other hand, always rounding down results in a discrepancy of up to $d \cdot \mathrm{diam}(G)$, where $\mathrm{diam}(G)$ denotes the diameter of the underlying graph G. To overcome these problems we adopt a *vertex-oriented* view in this paper where the vertices (not edges) decide randomly how much they are sending.

1.1 Related Work. Due to the vast amount of literature on load balancing, we consider only previous work dealing with diffusion load balancing or randomized algorithms for neighborhood-based load balancing. We do not consider literature on the dimension exchange model in general, or literature for the token distribution model.

Continuous Diffusion. The diffusion model was first studied by Cybenko [3] and, independently, Boillat [1]. Cybenko [3] (see also [13, 15]) shows a tight connection between the convergence rate of the diffusion algorithm and the absolute value of the second largest eigenvalue $\lambda_{\max}$ of the diffusion matrix $\mathbf{P}$ with $\mathbf{P}_{ij} = 1/(d+1)$ if $\{i,j\} \in E$. Subramanian and Scherson [15] observe similar relations between convergence time and certain properties of the underlying network like electrical and fluid conductance.

Muthukrishnan et al. [13] refer to the above diffusion model as the *first order scheme* and generalize it to the so called *second order scheme*. Here the load transferred over an edge (i,j) in step t does not only depend on the load difference of i and j, but also on the amount of load transferred over the edge in step $t-1$. Diekmann, Frommer, and Monien [4] extend the idea of [13] and propose a general framework to analyze the convergence behavior of a wide range of diffusion type methods.

Discrete Diffusion. Rabani et al. [14] consider the diffusion algorithm that always rounds down (called RSW algorithm in the following). They approximate the idealized and continuous process by this process with indivisible load. To quantify the deviation of the discrete load process from the idealized process, they propose a natural measure, the *local divergence* Ψ_1. The local divergence measures the sum of load differences across all edges in the network, aggregated over the time. They give a general bound on Ψ_1 in terms of $\lambda_{\max}$ (which is the second largest eigenvalue in absolute value of the diffusion matrix). By a more careful analysis, they also get improved bounds on Ψ_1 for tori graphs resulting in tight bounds on the discrepancy achieved by their algorithm.

Discrete Load Balancing via Random Walks. Elsässer et al. [7, 8] propose an algorithm based on random walks. They show that after $\mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ steps, the maximum load is at most the average load plus a constant [7]. In comparison to our algorithm, their algorithm is more complicated and not a simple diffusion type algorithm. For example, vertices require an estimate of n and they have to compute the average load during the balancing procedure. Moreover, the final stage uses concurrent random walks (representing tokens) to reduce the maximum load. In this stage, the load transfer along an edge may be much smaller (or higher) than *load-difference*$/(d+1)$.

Discrete Neighborhood Load Balancing with Randomization. In [9], the authors consider a dimension-exchange algorithm using randomly or deterministically generated matchings. Their algorithm randomly decides to round up or down. For detailed results see Table 1. Note that an algorithm in the dimension-exchange model is typically much easier to analyze than diffusion algorithms since every node exchanges load with at most one neighbor. In [10], the authors analyze a deterministic modification of the standard diffusion algorithm for hypercubes and constant-dimensional tori. The idea is that each edge keeps track of its own rounding errors. In each step an edge's decision to round up or down is done in a way that the sum of its rounding errors is minimized. Again, the detailed results can be found in Table 1. The authors of [10] also consider a randomized version of the diffusion algorithm. Their approach is edge-based, more precisely, edges decide independently at random whether to round up or down. They present a general upper bound for their approach in terms of $\lambda_{\max}$. Note that both algorithms in [10] may generate negative load due to the edge-based rounding.

Source of Inspiration. We wish to point out that our work was inspired by recent combinatorial results regarding so-called *rotor-router walks* [2, 5]. Unlike in a random walk, in a rotor-router walk each vertex serves its neighbors in a fixed order. The resulting (completely deterministic) walk nevertheless closely resembles a random walk in several respects. Similarly, one can say that in each round of our load-balancing algorithm a vertex chooses a random order of its neighbors (and itself) and sends around all its tokens in this order in a round-robin fashion.

1.2 Our Contribution. Algorithm. We consider a vertex-based randomized diffusion algorithm for the discrete model with indivisible tokens. Let d be the degree of the (regular) network and let X_i be the load of vertex i. Our algorithm works as follows. First, vertex i sends $\lfloor X_i/(d+1) \rfloor$ tokens to each neighbor and keeps the same amount of tokens for itself. Then the remaining $X_i - (d+1)\lfloor X_i/(d+1)\rfloor$ tokens (called *excess tokens*) are randomly distributed (without replacement) among vertex i and its d neighbors.

Results. To state our results formally, we let $\tau(G,K) = \mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ be the number of steps after which the continuous process achieves a constant discrepancy for any initial load distribution with discrepancy K (cf. Fact 2.2, [14]). All our bounds on the discrepancy are independent of the initial load vector, and hold with high probability (w.h.p.), i.e., with probability at least $1 - n^{-\Omega(1)}$.

THEOREM 1.1. *Let G be an arbitrary d-regular graph and let K be the initial discrepancy. Then the discrepancy after $\tau(G,K) = \mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ rounds is w.h.p. at most*

(1) $\mathcal{O}(\Upsilon_2(G)\sqrt{d\log n})$,
(2) $\mathcal{O}\big(d + \sqrt{d\log(n)\ ((\Upsilon_2(G))^2 - d)}\big)$, *and*
(3) $\mathcal{O}\big(d\,\frac{\log\log n}{1-\lambda_{\max}}\big)$.

The role of $\Upsilon_2(G)$ is similar to the local divergence $\Psi_1(G)$ used in [14] (cf. Definitions 2.1 and 2.2). $\Upsilon_2(G)$ is much smaller than $\Psi_1(G)$, i.e., $\Upsilon_2(G) \leqslant \sqrt{\Psi_1(G)}$ for any graph G. The improvement is due to the more balanced reallocation of the excess tokens due to our randomized approach and

The next theorem provides more specific bounds on the discrepancy. It is derived by first bounding $\Upsilon_2(G)$ and then applying Theorem 1.1.

THEOREM 1.2. *The following upper bounds on the discrepancy after $\tau(G,K) = \mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ rounds hold w.h.p.*

(1) $\mathcal{O}\Big(d\sqrt{\log n} + \sqrt{\frac{d\log n\log d}{1-\lambda_{\max}}}\Big)$,
(2) d-regular Expander: $\mathcal{O}(d\log\log n)$,
(3) r-dim. Torus, $r = \mathcal{O}(1)$: $\mathcal{O}(\sqrt{\log n})$, *and*
(4) Hypercube: $\mathcal{O}(\log n)$.

Let us compare our results to the RSW algorithm (see [14]) since that algorithm is also a very natural diffusion algorithm that avoids negative loads. For d-regular expanders, [14] proves a discrepancy bound of $\mathcal{O}(d\log n)$ after $\tau(G,K)$ rounds. This is almost tight, as $d{\cdot}\mathrm{diam}(G)$ is a simple lower bound for the RSW algorithm. Hence for small d, we obtain an exponential improvement in terms of the discrepancy.

For the r-dimensional Torus graph, [14, Theorem 8] proves a bound of $\mathcal{O}(n^{1/r})$ on the discrepancy after $\tau(G,K)$ rounds. This is tight due to the lower bound of $\mathrm{diam}(G)$. Again, our new algorithm achieves an exponential improvement. For the hypercube with n vertices, [14, Theorem 4] implies a discrepancy bound of $\mathcal{O}(\log^3 n)$ after $\tau(G,K)$ rounds.

The techniques used to analyze our new algorithm can be used to prove a tight bound of $\Theta(\log^2 n)$ on the discrepancy of the RSW algorithm. For our new algorithm we obtain a smaller discrepancy bound of $\mathcal{O}(\log n)$.

Techniques. The key ingredient of the analysis in [9, 10, 14] is "*an appropriate edge-oriented view of the rounding errors in each balancing step, which allows them to be handled independently*" (as stated by Rabani et al. [14]). The problem with vertex-oriented algorithms are the dependencies between the rounding results for edges incident to the same vertex. To deal with these dependencies we use a different analysis compared to [9, 10]. Our analysis is based on martingale tail estimates. The other main technical contribution is the use of the new parameter $\Upsilon_2(G)$ (Definition 2.2) as opposed to the local divergence $\Psi_1(G)$ as used in [14].

2 Algorithms and Notation

We use standard graph-theoretical notation. Let $G = (V,E)$ be a connected, undirected, d-regular and simple graph with n vertices $[n] := \{1,2,\ldots n\}$. The neighborhood of a vertex i is denoted by $N(i)$. For a pair of vertices $i,j \in V(G)$, let $\mathrm{dist}(i,j)$ be the length of a shortest path between i and j, and $\mathrm{diam}(G)$ be the diameter of G. $[i:j]$ refers to an edge $\{i,j\} \in E$ with $i < j$. Every vertex in the graph has a certain amount of load items (tokens). We assume that the load is indivisible and each token is of unit-size.

We denote by $\mathbf{P}$ the *transition matrix*, i.e., $\mathbf{P}_{i,j} = \frac{1}{d+1}$ if $\{i,j\} \in E$ or $i = j$, and $\mathbf{P}_{i,j} = 0$ otherwise. We will often use $\mathbf{P}^t$ which means that we raise the matrix $\mathbf{P}$ to the power of t. Note that $\mathbf{P}^t_{i,j}$ can be also seen as the probability for a random walk being located at vertex j at step t, when having started from vertex i.

For the estimation of the convergence of our processes, the absolute value of the second largest eigenvalues of $\mathbf{P}$ plays a crucial role. Let us denote the eigenvalues of $\mathbf{P}$ by $1 = \lambda_1 \geqslant \lambda_2 \geqslant \lambda_3 \geqslant \ldots \geqslant \lambda_n > -1$ and define

$$\lambda_{\max} := \max\{\lambda_2, |\lambda_n|\}.$$

We show the following general bound, the proof can be found in the full version.

LEMMA 2.1. *For any graph G, $1/(1-\lambda_{\max}(G)) \leqslant 2n^4$.*

For bounding the deviation between the discrete and continuous process, we adapt a definition from [9] that generalizes the original definition of local divergence from [14] for $p = 1$.

Graph class	FS [9]	RSW [14]	FGS [10] det.	FGS [10] rand.	our algorithm
d-reg. graph	$\mathcal{O}(\Psi_2(G)\sqrt{\log n})$	$\mathcal{O}(\Psi_1(G))$	–	–	$\mathcal{O}(\Upsilon_2(G)\sqrt{d\log n})$
	$\mathcal{O}\left(\frac{d\log\log n}{1-\lambda_{\max}}\right)$	–	–	$\mathcal{O}\left(\frac{d\log\log n}{1-\lambda_{\max}}\right)$	$\mathcal{O}\left(\frac{d\log\log n}{1-\lambda_{\max}}\right)$
	$\mathcal{O}\left(\sqrt{\frac{d\log n}{1-\lambda_{\max}}}\right)$	$\mathcal{O}\left(\frac{d\log n}{1-\lambda_{\max}}\right)$	–	–	$\mathcal{O}\left(d\sqrt{\log n}+\sqrt{\frac{d\log n\log d}{1-\lambda_{\max}}}\right)$
d-reg. expander	$\mathcal{O}(d\log\log n)$	$\mathcal{O}(d\log n)$	–	$\mathcal{O}(d\log\log n)$	$\mathcal{O}(d\log\log n)$
hypercube	$\mathcal{O}(\log^2 n)$	$\Theta(\log^2 n)$	$\Theta(\log n)$	$\mathcal{O}(\log^2 n\,\log\log n)$	$\mathcal{O}(\log n)$
r-dim. torus	$\mathcal{O}(n^{1/(2r)}\sqrt{\log n})$	$\Theta(n^{1/r})$	$\mathcal{O}(1)$	$\mathcal{O}(n^{1/r}\,\log\log n)$	$\mathcal{O}(\sqrt{\log n})$
Properties	**FS [9]**	**RSW [14]**	**FGS [10] det.**	**FGS [10] rand.**	**our algorithm**
diffusion	✗	✔	✔	✔	✔
no neg. load	✔	✔	✗	✗	✔

Table 1: Discrepancy of neighborhood load balancing after $\tau(G,K)=\Theta(\log(Kn)/1-\lambda_{\max})$ rounds.

Definition 2.1. ([9, 14]) *For any $p\in\mathbb{N}_{>0}$, the local p-divergence of a graph $G=(V,E)$ is*

$$\Psi_p(G):=\max_{k\in V}\left(\sum_{t=0}^{\infty}\sum_{[i:j]\in E}\left|\mathbf{P}^t_{i,k}-\mathbf{P}^t_{j,k}\right|^p\right)^{1/p}.$$

Note that $\Psi_2(G)^2\leqslant\Psi_1(G)$ since $\left|\mathbf{P}^t_{i,k}-\mathbf{P}^t_{j,k}\right|\leqslant 1$ for all t,i,k. As pointed out in [14], "*$\Psi_1(G)$ is a natural quantity that measures the sum of load differences across all edges in the network, aggregated over time (and suitably normalized) which may be of independent interest*". Here, we will mainly consider a natural extension of $\Psi_1(G)$ to the ℓ_2-norm, $\Psi_2(G)$, and $\Upsilon_2(G)$ which is defined below.

Definition 2.2. *For any $p\in\mathbb{N}_{>0}$, the refined local p-divergence of a graph $G=(V,E)$ is defined as*

$$\Upsilon_p(G):=\max_{k\in V}\left(\frac{1}{2}\sum_{t=0}^{\infty}\sum_{i=1}^{n}\max_{j\in N(i)}\left|\mathbf{P}^t_{i,k}-\mathbf{P}^t_{j,k}\right|^p\right)^{1/p}.$$

Note that $\Upsilon_p(G)\leqslant\Psi_p(G)$, since for each $\{i,j\}\in E(G)$ the term $\left|\mathbf{P}^t_{i,k}-\mathbf{P}^t_{j,k}\right|^p$ appears once in $\Psi_p(G)$ and at most twice in $\Upsilon_p(G)$.

For our probabilistic analysis, we use the following concentration result for martingales, which is commonly known as the "method of average bounded differences".

Theorem 2.1. ([6, p. 83]) *Let $Y_1,\ldots,Y_n$ be an arbitrary set of random variables and let f be a function of these random variables satisfying the property that for each $\ell\in[n]$, there is a non-negative c_ℓ such that*

$$\left|\mathbf{E}\left[f\mid Y_\ell,Y_{\ell-1},\ldots,Y_1\right]-\mathbf{E}\left[f\mid Y_{\ell-1},\ldots,Y_1\right]\right|\leqslant c_\ell.$$

Then for any $\delta>0$,

$$\mathbf{Pr}\left[|f-\mathbf{E}\left[f\right]|>\delta\right]\leqslant 2\,\exp\left(-\frac{\delta^2}{2c}\right),$$

where $c:=\sum_{\ell=1}^n c_\ell^2$.

2.1 Our Discrete Process.

Our balancing procedure proceeds in rounds $1,2,\ldots$. Fix a vertex i at some round and let X_i be the current load of this vertex. Then i sends $\lfloor X_i/(d+1)\rfloor$ tokens to each of its neighbors and keeps $\lfloor X_i/(d+1)\rfloor$ for itself. The remaining $X_i-(d+1)\lfloor X_i/(d+1)\rfloor\in[0,d]$ *excess-tokens* are distributed randomly (without replacement) among i and its d neighbors.

To describe our processes more formally, we first present our notation that is based on [14]. For any round t, let $X^{(t)}$ be the n-dimensional load-vector at (the end of) step t (load vectors are always regarded as column-vectors here). The *discrepancy* of the load vector $X^{(t)}$ at step t is defined as $\max_{i,j\in[n]}\left|X^{(t)}_i-X^{(t)}_j\right|$. For each edge $\{i,j\}\in E$ we define a random variable $Z^{(t)}_{i,j}$ with $Z^{(t)}_{i,j}=1$ if i sends an excess token to j at step t, and $Z^{(t)}_{i,j}=0$ otherwise. Similarly, let $Z^{(t)}_{i,i}$ be one if i keeps an excess token for itself, and zero otherwise. Note that each $Z^{(t)}_{i,j}$ with $j\in N(i)\cup\{i\}$ is a Bernoulli random variable with

$$\mathbf{Pr}\left[Z^t_{i,j}=1\right]=\frac{X^{(t-1)}_i}{d+1}-\left\lfloor\frac{X^{(t-1)}_i}{d+1}\right\rfloor.$$

Additionally, the number of excess tokens sent out by i satisfies

$$Z^{(t)}_{i,i}+\sum_{j:\,\{i,j\}\in E}Z^{(t)}_{i,j}$$

$$=X^{(t-1)}_i-(d+1)\left\lfloor\frac{X^{(t-1)}_i}{d+1}\right\rfloor. \tag{2.1}$$

Note that $Z_{i,j}$ and $Z_{j,i}$ are independent for $i\neq j$. Now we can describe the discrete process as follows,

$$X^{(t)}_i=\left\lfloor\frac{X^{(t-1)}_i}{d+1}\right\rfloor+Z^{(t)}_{i,i}$$

$$(2.2) \qquad + \sum_{j\colon \{i,j\}\in E} \left(\left\lfloor \frac{X_j^{(t-1)}}{d+1} \right\rfloor + Z_{j,i}^{(t)} \right).$$

2.2 The Continuous Process. Here we consider the continuous process where the load is arbitrarily divisible. The load vector in round t of this process is denoted by $\xi^{(t)}$. To analyze $X^{(t)}$ we bound its deviation from $\xi^{(t)}$ Here we use the fact that the evolution of $\xi^{(t)}$ is well-understood in the area of Markov chain theory since $\xi^{(t)} = \xi^{(t-1)}\,\mathbf{P}$, which results in $\xi^{(t)} = \xi^{(0)}\,\mathbf{P}^t$. Alternatively, we can write

$$(2.3) \qquad \xi_i^{(t)} = \xi_i^{(t-1)} + \sum_{j\colon \{i,j\}\in E} \frac{\xi_j^{(t-1)} - \xi_i^{(t-1)}}{d+1}.$$

We define the average load as $\overline{\xi} := \sum_{i=1}^n \xi_i^{(0)}/n$. The following result bounds the load difference of the vertices and the average load in step t of the continuous process.

LEMMA 2.2. ([13, LEM. 1]) *Let $G = (V,E)$ be an arbitrary connected graph. Then for any initial vector $\xi^{(0)}$ and time step $t \geqslant 0$,*

$$\sum_{i=1}^n \left(\xi_i^{(t)} - \overline{\xi}\right)^2 \leqslant \lambda_{\max}^{2t} \sum_{i=1}^n \left(\xi_i^{(0)} - \overline{\xi}\right)^2.$$

We will use the following immediate consequence of this lemma.

COROLLARY 2.1. *Let $G = (V,E)$ be an arbitrary connected graph. Then for any time step $t \geqslant 0$ and any vertex $k \in V$,*

$$\sum_{i=1}^n \left(\mathbf{P}_{i,k}^t - \frac{1}{n}\right)^2 \leqslant \lambda_{\max}^{2t}.$$

The following well-known result bounds the discrepancy of ξ.

THEOREM 2.2. ([14, THM. 1]) *Let G be a regular graph with n vertices. For the continuous process, the discrepancy is reduced to $\varepsilon > 0$ after*

$$\frac{2}{1-\lambda_{\max}} \cdot \ln\left(\frac{K\,n^2}{\varepsilon}\right)$$

steps, where K is the discrepancy of the initial load vector.

By $\tau(G,K)$ we denote the number of steps required for the continuous process to achieve a discrepancy of 1 for any initial load vector with discrepancy K. Fact 2.2 implies that $\tau(G,K) = \mathcal{O}((\log(Kn)/(1-\lambda_{\max}))$.

2.3 Difference between Continuous Process and Discrete Process. To obtain results for the discrete process, we upper bound the deviation between the discrete and continuous process at step t, assuming that both processes are initialized with the same load vector. t is chosen large enough so that after t steps the continuous process has achieved a discrepancy of at most 1 for every load vector with initial discrepancy K (cf. Fact 2.2). Hence, the discrepancy of the discrete process is upper bounded by the deviation between the discrete and continuous process (plus 1).

Similar to [9, 10, 14], we first express the discrepancy between the discrete and idealized process by a sum of weighted rounding errors (equation (2.7)). In this sum, the rounding errors are weighted by powers of the transition probabilities. In contrast to [9, 10, 14], the rounding errors (of the same time step) are not independent for all edges. This is due to our vertex-based approach and complicates the analysis.

To obtain a recursion for the discrete process, which similar to equation (2.3) for the continuous process, we plug equation (2.1) into equation (2.2) and obtain

$$\begin{aligned} X_i^{(t)} =& \left\lfloor \frac{X_i^{(t-1)}}{d+1} \right\rfloor - \left(\sum_{j\colon \{i,j\}\in E} Z_{i,j}^{(t)} \right) + X_i^{(t-1)} \\ & - (d+1) \left\lfloor \frac{X_i^{(t-1)}}{d+1} \right\rfloor \\ & + \sum_{j\colon \{i,j\}\in E} \left(\left\lfloor \frac{X_j^{(t-1)}}{d+1} \right\rfloor + Z_{j,i}^{(t)} \right) \\ =& X_i^{(t-1)} + \sum_{j\colon \{i,j\}\in E} \left(\left\lfloor \frac{X_j^{(t-1)}}{d+1} \right\rfloor - \left\lfloor \frac{X_i^{(t-1)}}{d+1} \right\rfloor \right. \\ (2.4) \qquad & \left. + Z_{j,i}^{(t)} - Z_{i,j}^{(t)} \right). \end{aligned}$$

Comparing equation (2.4) to equation (2.3) motivates the definition of the random variable $\Delta_{i,j}^{(t)}$, which counts the rounding error made by vertex i on the edge $\{i,j\}$ in step t.

$$\begin{aligned} \Delta_{i,j}^{(t)} :=& -\frac{X_j^{(t-1)}}{d+1} + \frac{X_i^{(t-1)}}{d+1} + \left\lfloor \frac{X_j^{(t-1)}}{d+1} \right\rfloor - \left\lfloor \frac{X_i^{(t-1)}}{d+1} \right\rfloor \\ (2.5) \qquad & + Z_{j,i}^{(t)} - Z_{i,j}^{(t)}. \end{aligned}$$

This allows us to write

$$(2.6) \qquad X_i^{(t)} = X_i^{(t-1)} + \sum_{j\colon \{i,j\}\in E} \frac{X_j^{(t-1)} - X_i^{(t-1)}}{d+1} + \Delta_{i,j}^{(t)}.$$

Now we state some basic properties of the rounding errors. The proofs can be found in the full version of the paper.

LEMMA 2.3. *Let $G = (V, E)$ be an arbitrary connected graph.*

(1) For every $\{i, j\} \in E$ and time step t, $\Delta_{i,j}^{(t)} = -\Delta_{j,i}^{(t)}$ and $\mathbf{E}\left[\Delta_{i,j}^{(t)}\right] = 0$.
(2) Consider two vertex-disjoint edges $(\{i, j\}, \{k, \ell\}) \in E$ and assume that $X^{(t-1)}$ is fixed. Then $\Delta_{i,j}^{(t)}$ and $\Delta_{k,\ell}^{(t)}$ are independent.

We now continue by returning to equation (2.6). For any vertex $i \in V$ and step t, let us define an error vector $\Delta^{(t-1)}$ with $\Delta_i^{(t)} := \sum_{j:\ \{i,j\}\in E} \Delta_{i,j}^{(t)}$. With this notation we have

$$X^{(t)} = X^{(t-1)}\mathbf{P} + \Delta^{(t)}.$$

Solving this recursion (see [14]) and setting $\xi^{(0)} = X^{(0)}$ results in

$$\begin{aligned} X^{(t)} &= X^{(0)}\mathbf{P}^t + \sum_{s=0}^{t-1} \Delta^{(t-s)}\mathbf{P}^s \\ &= \xi^{(t)} + \sum_{s=0}^{t-1} \Delta^{(t-s)}\mathbf{P}^s, \end{aligned}$$

where $\mathbf{P}^0$ is the $n \times n$-identity matrix. Hence, for any vertex $k \in V$

$$\begin{aligned} X_k^{(t)} - \xi_k^{(t)} &= \sum_{s=0}^{t-1}\sum_{i=1}^{n} \Delta_i^{(t-s)}\mathbf{P}_{i,k}^s \\ &= \sum_{s=0}^{t-1}\sum_{i=1}^{n}\sum_{j:\ \{i,j\}\in E} \Delta_{i,j}^{(t-s)}\mathbf{P}_{i,k}^s \\ &= \sum_{s=0}^{t-1}\sum_{[i:j]\in E} \Delta_{i,j}^{(t-s)} \left(\mathbf{P}_{i,k}^s - \mathbf{P}_{j,k}^s\right), \end{aligned} \tag{2.7}$$

where the last equality uses $\Delta_{i,j}^{(t-s)} = -\Delta_{j,i}^{(t-s)}$ (see Lemma 2.3 (1)).

3 Proof of Theorem 1.1

We now bound the discrepancy of our discrete process in terms of the local divergence $\Upsilon_2(G)$. We do this by upper bounding the deviation between the discrete and the continuous process. A similar approach was used in Rabani et al. [14], who bounded this deviation in terms of $\Psi_1(G)$. They showed that reducing the initial discrepancy from K to $\mathcal{O}(\Psi_1(G))$ can be achieved within $\mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ steps for any initial load vector. However, it turns out that our randomized process can be bounded in terms of $\Upsilon_2(G)$. Note that $\Upsilon_2(G)$ is in general much smaller than $\Upsilon_1(G)$ (or $\Psi_1(G)$) (cf. the remarks after Definition 2.1).

Proof. [Proof of Theorem 1.1] We start with the proof of the first statement. Let us now fix a vertex $k \in V$ and a time step t. Recall from equation (2.7) that

$$\begin{aligned} X_k^{(t)} - \xi_k^{(t)} &= \sum_{s=0}^{t-1}\sum_{[i:j]\in E} \Delta_{i,j}^{(t-s)} \left(\mathbf{P}_{i,k}^s - \mathbf{P}_{j,k}^s\right) \\ &= \sum_{s=1}^{t}\sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right). \end{aligned} \tag{3.8}$$

Consider the random variable $X_k^{(t)} - \xi_k^{(t)}$. By Lemma 2.3, $\mathbf{E}\left[X_k^{(t)} - \xi_k^{(t)}\right] = 0$. Our goal is to apply the martingale tail estimate from Theorem 2.1 to $f_k := X_k^{(t)} - \xi_k^{(t)}$. We first rewrite f_k,

$$\begin{aligned} f_k &= \sum_{s=1}^{t}\sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right) \\ &= \sum_{s=1}^{t}\sum_{[i:j]\in E} \Bigg(-\frac{X_j^{(t-1)}}{d+1} + \frac{X_i^{(t-1)}}{d+1} + \left\lfloor \frac{X_j^{(t-1)}}{d+1} \right\rfloor \\ &\qquad - \left\lfloor \frac{X_i^{(t-1)}}{d+1} \right\rfloor + Z_{j,i}^{(t)} - Z_{i,j}^{(t)} \Bigg) \cdot \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right), \end{aligned}$$

where the last equality follows by the definition of $\Delta_{i,j}^{(s)}$.

We observe that for a fixed load vector $X^{(0)}$ the function f_k depends only on the randomly chosen destinations of the excess tokens. There are t steps, n nodes, and at most d excess tokens per node per step. We describe these random choices by a sequence of $t \cdot n \cdot d$ random variables, $Y_1, Y_2, \ldots, Y_{tnd}$. For any ℓ with $1 \leqslant \ell \leqslant tnd$, let $(s, i, r) \in [t] \times [n] \times [d]$ be such that $\ell = (s-1)\,nd + (i-1)\,d + r$ (note that (s, i, r) is the ℓ-th largest element in an increasing lexicographic ordering of $[t] \times [n] \times [d]$). Then Y_ℓ refers to the destination of the r-th excess token of vertex i at step s (if there is one). More precisely,

$$Y_\ell := \begin{cases} j & \text{if } r \leqslant X_i^{(s-1)} - (d+1)\left\lfloor \frac{X_i^{(s-1)}}{d+1} \right\rfloor \text{ and the } r\text{-th excess token of vertex } i \text{ at step } s \text{ is sent to } j, \\ 0 & \text{otherwise.} \end{cases}$$

In order to apply Theorem 2.1, we have to upper bound

$$\left| \mathbf{E}\left[f_k \mid Y_\ell, \ldots, Y_1\right] - \mathbf{E}\left[f_k \mid Y_{\ell-1}, \ldots, Y_1\right] \right|. \tag{3.9}$$

Consider now a fixed ℓ that corresponds to (s_1, i_1, r_1) in the lexicographic ordering. To bound equation (3.9), we use equation (3.8) to get

$$\begin{aligned}&\left|\mathbf{E}\left[f_k \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[f_k \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \\ &\leqslant \sum_{s=1}^{t} \sum_{[i:j]\in E} \left|\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right]\right. \\ &\quad \left. - \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\end{aligned}$$

In the remainder of the proof we split the sum over s into the three parts $1 \leqslant s < s_1$, $s = s_1$, and $s_1 < s \leqslant t$. We prove that the parts $s < s_1$ and $s > s_1$ both equal zero while the part $s = s_1$ is upper bounded by $2 \cdot \max_{j\in N(i_1)} \left|\mathbf{P}_{i_1,k}^{t-s_1} - \mathbf{P}_{j,k}^{t-s_1}\right|$.

$\underline{\mathbf{s < s_1}:}$ For every $\{i, j\} \in E$, $\Delta_{i,j}^{(s)}$ is already determined by $Y_{\ell-1}, \ldots, Y_1$. Hence,

$$\sum_{s=1}^{s_1-1} \sum_{[i:j]\in E} \left|\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| = 0. \tag{3.10}$$

$\underline{\mathbf{s = s_1}:}$ This is the most involved case due to the dependencies among $\{\Delta_{i,j}^{(s)} : \{i, j\} \in E\}$.

$$\begin{aligned}&\sum_{[i:j]\in E} \left|\mathbf{E}\left[\Delta_{i,j}^{(s)} \,\middle|\, Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \\ &\leqslant \sum_{[i:j]\in E} \left|\mathbf{E}\left[-\frac{X_j^{(s-1)}}{d+1} + \frac{X_i^{(s-1)}}{d+1} + \left\lfloor\frac{X_j^{(s-1)}}{d+1}\right\rfloor - \left\lfloor\frac{X_i^{(s-1)}}{d+1}\right\rfloor + Z_{j,i}^{(s)} - Z_{i,j}^{(s)} \,\middle|\, Y_\ell, Y_{\ell-1}, \ldots, Y_1\right]\right. \\ &\quad - \mathbf{E}\left[-\frac{X_j^{(s-1)}}{d+1} + \frac{X_i^{(t-1)}}{d+1} + \left\lfloor\frac{X_j^{(s-1)}}{d+1}\right\rfloor - \left\lfloor\frac{X_i^{(s-1)}}{d+1}\right\rfloor \left. + Z_{j,i}^{(s)} - Z_{i,j}^{(s)} \,\middle|\, Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \\ &= \sum_{[i:j]\in E} \left|\mathbf{E}\left[Z_{j,i}^{(s)} - Z_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[Z_{j,i}^{(s)} - Z_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \qquad (3.11) \\ &\leqslant \sum_{[i:j]\in E} \left(\left|\mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| + \right. \\ &\quad \left. \left|\mathbf{E}\left[Z_{j,i}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[Z_{j,i}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\right) \\ &= \sum_{i\in V} \sum_{j\in N(i)} \left|\Lambda_{i,j}^{(s)}\right| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \\ &\leqslant \sum_{i\in V} \left(\max_{j\in N(i)} \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\right) \sum_{j\in N(i)} \left|\Lambda_{i,j}^{(s)}\right|, \qquad (3.12)\end{aligned}$$

where we used

$$\Lambda_{i,j}^{(s)} := \mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]$$

to simplify the notation. Eqn. 3.11 follows as $Y_{\ell-1}, \ldots, Y_1$ determine the load vector $X^{(s-1)}$. To bound equation (3.12) we consider $\sum_{j\in N(i)} \left|\Lambda_{i,j}^{(s)}\right|$ for $i = i_1$ and $i \neq i_1$ separately.

Case 1: Let $i = i_1$. Assume first $Y_\ell = 0$. This means that node i_1 has less than r_1 extra tokens at step t_1. Hence $\left|\Lambda_{i,j}^{s}\right| = 0$.

Now we assume that $Y_\ell \neq 0$. This means that node i_1 has at least r_1 extra tokens at step t_1. Let $b \geqslant r_1$ be the number of extra tokens of i_1 at step s_1. Clearly, b and the destinations of the extra tokens considered in the previous rounds, $Y_{\ell-r_1+1}, \ldots, Y_{\ell-1}$, are already determined by $Y_{\ell-1}, \ldots, Y_1$ (note that if $r_1 = 1$ then this set is empty). The remaining $Y_{\ell+1}, \ldots, Y_{\ell+b-r_1}$ are chosen uniformly at random among $(N(i_1) \cup \{i_1\}) \setminus \{Y_{\ell-r_1+1}, \ldots, Y_\ell\} =: \widetilde{N}(i_1)$ without replacement. Let $w \in \widetilde{N}(i_1)$ be the destination of the r_1-th excess token of i_1 at step s_1, that is, $Y_\ell = w$ and consequently, $Z_{i_1,w}^{(s_1)} = 1$. Clearly, $0 < \Lambda_{i_1,w}^{(s_1)} \leqslant 1$, and for all $j \in \widetilde{N}(i_1) \setminus \{w\}$, $\Lambda_{i_1,j}^{(s_1)} < 0$. For the vertices $j \in \{Y_{\ell-r_1+1}, \ldots, Y_{\ell-1}\}$, $\Lambda_{i_1,j}^{(s_1)} = 0$, as $Y_{\ell-1}, \ldots, Y_1$ already determined that $Z_{i_1,j}^{(s_1)} = 1$. Linearity of expectations yields

$$\begin{aligned}\sum_{j\in N(i_1)\cup\{i_1\}} \Lambda_{i_1,j}^{(s_1)} &= \mathbf{E}\left[\sum_{j\in N(i_1)\cup\{i_1\}} Z_{i_1,j}^{(s_1)} \,\middle|\, Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ &\quad - \mathbf{E}\left[\sum_{j\in N(i_1)\cup\{i_1\}} Z_{i_1,j}^{(s_1)} \,\middle|\, Y_{\ell-1}, \ldots, Y_1\right] = 0.\end{aligned}$$

The last equality holds since $\sum_{j\in N(i_1)\cup\{i_1\}} Z_{i_1,j}^{(s_1)} = b$

and b is determined by $Y_{\ell-1}, \ldots, Y_1$. Hence,

$$\begin{aligned} &\sum_{j \in N(i_1) \cup \{i_1\}} \left|\Lambda_{i_1,j}^{(s_1)}\right| \\ &= \sum_{\substack{j \in N(i_1) \cup \{i_1\}: \\ \Lambda_{i_1,j}^{(s_1)} > 0}} \Lambda_{i_1,j}^{(s_1)} - \sum_{\substack{j \in N(i_1) \cup \{i_1\}: \\ \Lambda_{i_1,j}^{(s_1)} \leqslant 0}} \Lambda_{i_1,j}^{(s_1)} \\ &= 2 \cdot \sum_{\substack{j \in N(i_1) \cup \{i_1\}: \\ \Lambda_{i_1,j}^{(s_1)} > 0}} \Lambda_{i_1,j}^{(s_1)} = 2\left|\Lambda_{i_1,w}^{(s_1)}\right| \leqslant 2. \end{aligned} \tag{3.13}$$

Case 2: $i \neq i_1$. As ℓ corresponds to (s_1, i_1, r_1), the random variable $Z_{i,j}^{(s_1)}$ is independent of Y_ℓ, which is the choice of the r_1-th excess token of vertex i_1 at step s_1. Hence

$$\begin{aligned} \sum_{j \in N(i)} \left|\Lambda_{i,j}^{(s_1)}\right| = \sum_{j \in N(i)} \Big| \mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ - \mathbf{E}\left[Z_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\Big| = 0. \end{aligned}$$

Combining Case 1 and Case 2 we obtain

$$\begin{aligned} (3.12) &= \left(\max_{j \in N(i_1)} \left|\mathbf{P}_{i_1,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\right) \sum_{j \in N(i_1)} \left|\Lambda_{i_1,j}^{(s)}\right| \\ &\quad + \sum_{i \in V, i \neq i_1} \left(\max_{j \in N(i)} \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\right) \sum_{j \in N(i)} \left|\Lambda_{i,j}^{(s)}\right| \\ &\leqslant \max_{j \in N(i_1)} \left|\mathbf{P}_{i_1,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \cdot 2 + 0. \end{aligned} \tag{3.14}$$

<u>$\mathbf{s > s_1}$:</u> Let $\widetilde{\ell}$ be the largest integer that corresponds to time-step $s-1$. Since $s > s_1$, we have $s-1 \geqslant s_1$ and therefore $\widetilde{\ell} \geqslant \ell$. By the choice of $\widetilde{\ell}$, $Y_{\widetilde{\ell}}, \ldots, Y_1$ determine the load vector at the end of step s_1, $X^{(s_1)}$. By Lemma 2.3, we obtain $\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\widetilde{\ell}}, \ldots, Y_1\right] = 0$, and by the chain rule of expectations,

$$\begin{aligned} &\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ &= \mathbf{E}\Big[\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\widetilde{\ell}}, \ldots, Y_1\right] \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\Big] \\ &= \mathbf{E}\left[0 \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ &= 0. \end{aligned}$$

With the same arguments, $\mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right] = 0$, and therefore

$$\begin{aligned} &\sum_{s=s_1+1}^{t} \sum_{[i:j] \in E} \Big| \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ &\quad - \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\Big| \cdot \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \\ &= 0. \end{aligned} \tag{3.15}$$

This finishes the case distinction. Combining equations (3.10), (3.14), and (3.15) for the three cases $s < s_1$, $s = s_1$, and $s > s_1$, we obtain that for every fixed $1 \leqslant \ell \leqslant tnd$,

$$\begin{aligned} &\left| \mathbf{E}\left[f_k \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] - \mathbf{E}\left[f_k \mid Y_{\ell-1}, \ldots, Y_1\right]\right| \\ &\leqslant \sum_{s=1}^{t} \sum_{[i:j] \in E} \Big| \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_\ell, Y_{\ell-1}, \ldots, Y_1\right] \\ &\quad - \mathbf{E}\left[\Delta_{i,j}^{(s)} \mid Y_{\ell-1}, \ldots, Y_1\right]\Big| \cdot \left|\mathbf{P}_{i,k}^{t-s_1} - \mathbf{P}_{j,k}^{t-s_1}\right| \\ &= 0 + \max_{j \in N(i_1)} \left|\mathbf{P}_{i_1,k}^{t-s_1} - \mathbf{P}_{j,k}^{t-s_1}\right| \cdot 2 + 0 \\ &= 2 \cdot \max_{j \in N(i_1)} \left|\mathbf{P}_{i_1,k}^{t-s_1} - \mathbf{P}_{j,k}^{t-s_1}\right| =: c_\ell. \end{aligned}$$

To apply Theorem 2.1, we first estimate $\sum_{\ell=1}^{tnd} (c_\ell)^2$.

$$\begin{aligned} \sum_{\ell=1}^{tnd} (c_\ell)^2 &= \sum_{s=1}^{t} \sum_{i=1}^{n} \sum_{r=1}^{d} \left(2 \max_{j \in N(i)} \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right|\right)^2 \\ &= 4d \sum_{s=0}^{t-1} \sum_{i=1}^{n} \max_{j \in N(i)} \left(\mathbf{P}_{i,k}^{s} - \mathbf{P}_{j,k}^{s}\right)^2 \\ &\leqslant 4d \max_{k \in V} \left(\sum_{s=0}^{\infty} \sum_{i=1}^{n} \max_{j \in N(i)} \left(\mathbf{P}_{i,k}^{s} - \mathbf{P}_{j,k}^{s}\right)^2 \right) \\ &= 8d \left(\Upsilon_2(G)\right)^2. \end{aligned} \tag{3.16}$$

So we have for any $\delta \geqslant 0$,

$$\mathbf{Pr}\left[|f_k| > \delta\right] \leqslant 2 \exp\Big(-\delta^2 / \big(2 \sum_{\ell=1}^{tnd} (c_\ell)^2\big)\Big).$$

Hence by choosing $\delta := \Upsilon_2(G) \sqrt{32d \ln n}$, the probability above gets smaller than $2\,n^{-2}$. Applying the union bound we obtain

$$\mathbf{Pr}\left[\forall k \in V \colon |f_k| > \delta\right] \leqslant n\, 2n^{-2} = 2n^{-1}.$$

By equation (3.8), $\max_{k \in [n]} X_k^{(t)} \leqslant |\xi_k^{(t)}| + |f_k|$. For $t := \tau(G, K)$, we obtain $|\xi_k^{(t)} - \overline{\xi}| \leqslant 1$ for every vertex k. Hence

$$\max_{k \in [n]} X_k^{(t)} - \min_{k \in [n]} X_k^{(t)} \leqslant 2|f_k| + 2.$$

This implies

$$\mathbf{Pr}\left[\max_{i,j \in [n]} \left|X_i^{(t)} - X_j^{(t)}\right| \leqslant 2\delta + 2\right] \geqslant 1 - 2n^{-1},$$

as needed.

Now we prove of the second statement. Fix a vertex $k \in V$. Recall from equation (3.8) that

$$X_k^{(t)} - \xi_k^{(t)} = \sum_{s=1}^{t} \sum_{[i:j] \in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right). \tag{3.17}$$

We split the right hand side of equation (3.17) at step $t-1$ to obtain

$$\overbrace{\sum_{s=1}^{t-1} \sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)}^{=:g} + \underbrace{\sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{0} - \mathbf{P}_{j,k}^{0}\right)}_{=:h}.$$

We can bound h using the triangle inequality as follows

$$|h| \leqslant \sum_{[i:j]\in E} \left|\Delta_{i,j}^{(t)}\right| \cdot \left|\mathbf{P}_{i,k}^{0} - \mathbf{P}_{j,k}^{0}\right| \leqslant 1 \cdot \sum_{[i:j]\in E} \left| \mathbf{P}_{i,k}^{0} - \mathbf{P}_{j,k}^{0}\right| = d,$$

since

$$|\Delta_{i,j}^{(t)}| \leqslant 1 \text{ and } \sum_{i=1}^{n} \mathbf{P}_{i,k}^{0} = 1.$$

To bound h, we use the same approach as in the proof of the first statement in Theorem 1.1. Again, we define a sequence of random variables Y_ℓ with $1 \leqslant \ell \leqslant (t-1)nd$. In order to apply Theorem 2.1, we have to estimate the differences c_ℓ, $1 \leqslant \ell \leqslant (t-1)nd$. As in equation (3.16) we obtain

$$\sum_{\ell=1}^{(t-1)nd} (c_\ell)^2 \leqslant 2d \sum_{s=1}^{t-1}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2.$$

Since $\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{0} - \mathbf{P}_{j,k}^{0}\right)^2 = 2d$, we obtain that

$$2d \sum_{s=1}^{t-1}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2 \leqslant 4d \cdot \left((\Upsilon_2(G))^2 - d\right)$$

By Theorem 2.1, we obtain that

$$\mathbf{Pr}\left[|h| > \delta\right] \leqslant 2 \exp\left(-\delta^2 \Big/ \left(8d \cdot \left((\Upsilon_2(G))^2 - d\right)\right)\right).$$

Hence by choosing $\delta := \sqrt{16 \log(n)\, d\, ((\Upsilon_2(G))^2 - d)}$ we get $\mathbf{Pr}\left[|h| > \delta\right] \leqslant 2n^{-2}$. Hence,

$$\begin{aligned}\mathbf{Pr}&\left[\left|X_k^{(t)} - \xi_k^{(t)}\right| \geqslant d + \delta\right] \\ &\leqslant \mathbf{Pr}\left[|g| \geqslant d\right] \\ &\quad + \mathbf{Pr}\left[|h| \geqslant \sqrt{16 \log(n)\, d\, ((\Upsilon_2(G))^2 - d)}\right] \\ &\leqslant 0 + 2n^{-2} = 2n^{-2}.\end{aligned}$$

Taking the union bound over all vertices k yields,

$$\begin{aligned}\mathbf{Pr}&\Big[\forall k \in V\colon \left|X_k^{(t)} - \xi_k^{(t)}\right| \\ &\qquad \leqslant d + \sqrt{16 \log(n)\, d\, ((\Upsilon_2(G))^2 - d)}\Big] \\ &\leqslant n\, 2n^{-2} = 2n^{-1}.\end{aligned} \tag{3.18}$$

The third statement is shown by a similar approach. Again, fix a vertex $k \in V$ and a time step t. Now we split the right hand side of equation (3.17) at step $t - \vartheta$, where $\vartheta := (4 \ln\ln n)/(1 - \lambda_{\max})$.

$$\begin{aligned}&\sum_{s=1}^{t} \sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right) \\ &= \underbrace{\sum_{s=1}^{t-\vartheta} \sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)}_{=:g} \\ &\quad + \underbrace{\sum_{s=t-\vartheta+1}^{t} \sum_{[i:j]\in E} \Delta_{i,j}^{(s)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)}_{=:h}.\end{aligned}$$

We first bound the last part directly by applying the triangle inequality as follows.

$$\begin{aligned}|g| &\leqslant \sum_{s=t-\vartheta+1}^{t} \sum_{[i:j]\in E} \left|\Delta_{i,j}^{(s)}\right| \left|\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right| \\ &\leqslant \vartheta \sum_{[i:j]\in E} \left(\mathbf{P}_{i,k}^{t-s} + \mathbf{P}_{j,k}^{t-s}\right) \leqslant \vartheta\, d,\end{aligned}$$

where the first inequality holds since $|\Delta_{i,j}^{(s)}| \leqslant 1$ and where the last inequality holds since $\sum_{i=1}^{n} \mathbf{P}_{i,k}^{t-s} = 1$ for every k.

To bound h, we use the same approach as in the proof of the first statement in Theorem 1.1. Also here, we define a sequence of random variables Y_ℓ with $1 \leqslant \ell \leqslant (t-\vartheta)nd$. In order to apply Theorem 2.1, we have to estimate the differences c_ℓ, $1 \leqslant \ell \leqslant (t-\vartheta)nd$. As in equation (3.16) we obtain

$$\sum_{\ell=1}^{(t-\vartheta)nd} (c_\ell)^2 \leqslant 2d \sum_{s=1}^{t-\vartheta}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2.$$

By Theorem 2.1, we obtain that

$$\begin{aligned}&\mathbf{Pr}\left[|h| > \delta\right] \\ &\leqslant 2 \exp\left(-\delta^2 \Big/ \left(4d \sum_{s=1}^{t-\vartheta}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2\right)\right).\end{aligned}$$

Hence by choosing

$$\delta := \sqrt{8\log(n)\, d \sum_{s=1}^{t-\vartheta}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2}$$

we get $\mathbf{Pr}\left[|h| > \delta\right] \leqslant 2n^{-2}$ and

$$\begin{aligned}\mathbf{Pr}&\left[\left|X_k^{(t)} - \xi_k^{(t)}\right| \geqslant \vartheta\, d + \delta\right]\\ &\leqslant \mathbf{Pr}\left[|g| \geqslant \vartheta\, d\right] + \mathbf{Pr}\left[|h| \geqslant \delta\right]\\ &\leqslant 0 + 2n^{-2} = 2n^{-2}.\end{aligned}$$

Taking the union bound over all vertices k yields,

$$\begin{aligned}\mathbf{Pr}&\left[\forall k \in V\colon \left|X_k^{(t)} - \xi_k^{(t)}\right| \leqslant \vartheta\, d + \delta\right]\\ &\leqslant n\, 2n^{-2} = 2n^{-1}.\end{aligned} \tag{3.19}$$

In order to complete the proof, it remains to prove that $\delta = \mathcal{O}((d\,\ln\ln n)/(1-\lambda_{\max}))$.

$$\begin{aligned}&\sum_{s=1}^{t-\vartheta}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{t-s} - \mathbf{P}_{j,k}^{t-s}\right)^2\\ &= \sum_{s=\vartheta}^{t-1}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{i,k}^{s} - \mathbf{P}_{j,k}^{s}\right)^2\\ &\leqslant 2\sum_{s=\vartheta}^{t}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\left(\mathbf{P}_{i,k}^{s} - \frac{1}{n}\right)^2 + \left(\mathbf{P}_{j,k}^{s} - \frac{1}{n}\right)^2\right)\\ &\leqslant 2\sum_{s=\vartheta}^{t-1}\sum_{i=1}^{n} \left(\mathbf{P}_{i,k}^{s} - \frac{1}{n}\right)^2\\ &\quad + 2\sum_{s=\vartheta}^{t-1}\sum_{i=1}^{n} \max_{j\in N(i)} \left(\mathbf{P}_{j,k}^{s} - \frac{1}{n}\right)^2\\ &\leqslant 2\sum_{s=\vartheta}^{t-1}\sum_{i=1}^{n} \left(\mathbf{P}_{i,k}^{s} - \frac{1}{n}\right)^2\\ &\quad + 2\sum_{s=\vartheta}^{t-1}\sum_{j=1}^{n} d\left(\mathbf{P}_{j,k}^{s} - \frac{1}{n}\right)^2\\ &\leqslant (2d+2)\sum_{s=\vartheta}^{t-1} \lambda_{\max}^{2s},\end{aligned}$$

where the first inequality uses $(x-y)^2 \leqslant 2(x-z)^2 + 2(y-z)^2$ and the last inequality follows from Corollary 2.1. The last term can be now bounded as follows,

$$\begin{aligned}(2d+2)&\sum_{s=\vartheta}^{\infty} \lambda_{\max}^{2s}\\ &\leqslant (2d+2)\,\frac{\lambda_{\max}^{2\left(\frac{4\ln\ln n}{1-\lambda_{\max}}\right)}}{1-(\lambda_{\max})^2}\\ &\leqslant (2d+2)\,\frac{e^{-8\ln\ln n}}{1-\lambda_{\max}}\\ &= (2d+2)\,\frac{(\log n)^{-8}}{1-\lambda_{\max}},\end{aligned}$$

where the second last inequality uses the fact that $x^{1/(1-x)} \leqslant 1/e$ for $x \in [0,1)$. We can now use this bound to get a more explicit expression for the bound in equation (3.19),

$$\mathbf{Pr}\left[\forall k \in V\colon \left|X_k^{(t)} - \xi_k^{(t)}\right| \leqslant \frac{4d\ln\ln n}{1-\lambda_{\max}} + \sqrt{24\log(n)\, d^2\, \frac{(\log n)^{-8}}{1-\lambda_{\max}}}\right] \leqslant 2n^{-1}.$$

We choose $t = \tau(G,K)$ to get $|\xi_k^{(t)} - \overline{\xi}| \leqslant 1$ for every vertex k. As in the proof of Theorem 1.1 this yields

$$\mathbf{Pr}\left[\max_{i,j\in[n]} \left|X_i^{(t)} - X_j^{(t)}\right| \leqslant 2\delta + 2\right] \geqslant 1 - 4n^{-1}.$$

This completes the proof of the third statement.

4 Bounds on the Local Divergence and Proof of Theorem 1.2

Bounds on the Local 2-Divergence. We first present upper bounds on the (refined) local 2-divergence that are then used to prove Theorem 1.2.

THEOREM 4.1. *For any graph G,*

$$\Upsilon_2(G) = \mathcal{O}\left(\sqrt{d + \frac{\log d}{1-\lambda_{\max}}}\right)$$

Since for r-dimensional tori $1/(1-\lambda_{\max}) = \Theta(n^{2/r})$ and for hypercubes $1/(1-\lambda_{\max}) = \Theta(\log n)$, the following theorems represent improvements over the bound in Theorem 4.1 for these specific networks.

THEOREM 4.2. *For the r-dimensional torus graph with* $r = \mathcal{O}(1)$,

$$\Upsilon_2(G) \leqslant \Psi_2(G) = \mathcal{O}(1).$$

THEOREM 4.3. *For the hypercube G with n vertices* ($d = \log_2 n$),

$$\Upsilon_2(G) = \sqrt{d + \mathcal{O}(1)}.$$

Note that since $\Upsilon_2(G) \geqslant \sqrt{d}$ for any d-regular network, this bound is also tight. The proofs of Theorem 4.2 and Theorem 4.3 will be given in the full version of the paper. Now Theorem 1.2 follows by combining the three theorems above with Theorem 1.1.

Bounds on the Local 1-Divergence. In the following we give an *exact* bound on $\Psi_1(G)$.

THEOREM 4.4. *If G is the hypercube with n vertices,*

$$\Psi_1(G) = \frac{\log_2(n)+1}{n} \sum_{p=0}^{\log_2(n)-1} \sum_{\ell=p+1}^{\log_2 n} \binom{\log_2 n}{\ell} = \Theta(\log^2 n).$$

As the discrepancy of the RSW algorithm is at most $\Psi_1(G)$ after $\tau(G,K)$ rounds [14, Cor. 3], we obtain:

COROLLARY 4.1. *The discrepancy of the RSW algorithm [14] is at most $\mathcal{O}(\log^2 n)$ after $\tau(G,K) = \mathcal{O}(\log(Kn)\cdot\log^2 n)$ time steps.*

Note that the best-possible result from [14, Theorem 4] yields only a weaker bound of $\mathcal{O}(\log^3 n)$. Our result is tight since $d\cdot \mathrm{diam}(G) = (\log_2 n)^2$ is a lower bound.

5 Discussion

We presented a new diffusion-based load-balancing scheme which is very simple and avoids negative load. We show bounds on the discrepancy for general graphs depending on the local (or refined local) divergence and the eigenvalue gap of the graph. For (constant-degree) expander graphs we prove a discrepancy of $\mathcal{O}(\log\log n)$, for hypercubes of $\mathcal{O}(\log n)$, and for r-dimensional torus graphs of $\mathcal{O}(\sqrt{\log n})$.

References

[1] J. E. Boillat. Load balancing and poisson equation in a graph. *Concurrency: Pract. Exper.*, 2:289–313, 1990.

[2] J. Cooper and J. Spencer. Simulating a random walk with constant error. *Combinatorics, Probability & Computing*, 15:815–822, 2006.

[3] G. Cybenko. Load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

[4] R. Diekmann, A. Frommer, and B. Monien. Efficient Schemes for Nearest Neighbor Load Balancing. *Parallel Computing*, 25:789–812, 1999.

[5] B. Doerr and T. Friedrich. Deterministic random walks on the two-dimensional grid. *Combinatorics, Probability & Computing*, 18:123–144, 2009.

[6] D. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press, 2009.

[7] R. Elsässer and T. Sauerwald. Discrete Load Balancing is (Almost) as Easy as Continuous Load Balancing. In *29th Annual ACM Principles of Distributed Computing (PODC'10)*, pages 346–354, 2010.

[8] R. Elsässer, B. Monien, and S. Schamberger. Distributing unit size workload packages in heterogeneous networks. *Journal of Graph Algorithms and Applications*, 10:51–68, 2006.

[9] T. Friedrich and T. Sauerwald. Near-perfect load balancing by randomized rounding. In *41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pages 121–130, 2009.

[10] T. Friedrich, M. Gairing, and T. Sauerwald. Quasirandom load balancing. In *21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 1620–1629, 2010.

[11] B. Ghosh, F. T. Leighton, B. M. Maggs, S. Muthukrishnan, C. G. Plaxton, R. Rajaraman, A. W. Richa, R. E. Tarjan, and D. Zuckerman. Tight analyses of two local load balancing algorithms. *SIAM Journal on Computing*, 29:29–64, 1999.

[12] L. Lovász and P. Winkler. Mixing of random walks and other diffusions on a graph. *Surveys in combinatorics*, pages 119–154, 1995.

[13] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory Comput. Syst.*, 31:331–354, 1998.

[14] Y. Rabani, A. Sinclair, and R. Wanka. Local divergence of Markov chains and the analysis of iterative load balancing schemes. In *39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pages 694–705, 1998.

[15] R. Subramanian and I. D. Scherson. An analysis of diffusive load-balancing. In *6th ACM Symposium on Parallel Algorithms and Architectures (SPAA'94)*, pages 220–225, 1994.