TIGHT ANALYSES OF TWO LOCAL LOAD BALANCING ALGORITHMS

BHASKAR GHOSH[1] *F. T. LEIGHTON*[2] *BRUCE M. MAGGS*[3,4]

S. MUTHUKRISHNAN[5] *C. GREG PLAXTON*[6,7] *R. RAJARAMAN*[6,7]

ANDRÉA W. RICHA[3,4] *ROBERT E. TARJAN*[8] *DAVID ZUCKERMAN*[6,9]

Abstract.
This paper presents an analysis of the following load balancing algorithm. At each step, each node in a network examines the number of tokens at each of its neighbors and sends a token to each neighbor with at least $2d+1$ fewer tokens, where d is the maximum degree of any node in the network. We show that within $O(\Delta/\alpha)$ steps, the algorithm reduces the maximum difference in tokens between any two nodes to at most $O((d^2 \log n)/\alpha)$, where Δ is the global imbalance in tokens (i.e., the maximum difference between the number of tokens at any node initially and the average number of tokens), n is the number of nodes in the network, and α is the edge expansion of the network. The time bound is tight in the sense that for any graph with edge expansion α, and for any value Δ, there exists an initial distribution of tokens with imbalance Δ for which the time to reduce the imbalance to even $\Delta/2$ is at least $\Omega(\Delta/\alpha)$. The bound on the final imbalance is tight in the sense that there exists a class of networks that can be locally balanced everywhere (i.e., the maximum difference in tokens between any two neighbors is at most $2d$), while the global imbalance remains $\Omega((d^2 \log n)/\alpha)$. Furthermore, we show that upon reaching a state with a global imbalance of $O((d^2 \log n)/\alpha)$, the time for this algorithm to locally balance the network can be as large as $\Omega(n^{1/2})$. We extend our analysis to a variant of this algorithm for dynamic and asynchronous networks. We also present tight bounds for a randomized algorithm in which each node sends at most one token in each step.

Keywords: load balancing, distributed network algorithms.
AMS subject classification: 68Q22

1. Introduction. A natural way to balance the workload in a distributed system is to have each work station periodically poll the other stations to which it is connected and send some of its work to stations with less work pending. This paper analyzes the effectiveness of this local load balancing strategy in the simplified scenario in which each work station has a collection of independent unit-size jobs (called *tokens*) that can be executed on any other work station. We model a distributed system as a graph, where nodes correspond to work stations, and edges correspond to connections between stations, and we assume that in one unit of time, at most one token can be transmitted across an edge of the graph in each direction. Our analysis addresses only the static load balancing aspect of this problem: we assume that each processor has an initial collection of tokens, and that no tokens are created or destroyed while the tokens are being balanced.

We analyze the algorithms in this paper in terms of the initial *imbalance* of tokens, i.e., the maximum difference between the number of tokens at any node and the average number of tokens, which we denote Δ, the number of nodes in the graph, which we denote n, the maximum degree of the graph, d, and the node and edge expansion of the graph. We define the *node expansion* μ, of a graph G to be the largest value such that every set S of $n/2$ or fewer nodes in G has at least $\mu|S|$ neighbors outside of S. We define the *edge expansion* α, of a graph G to be the largest value such that for every set S of $n/2$ or fewer nodes in G, there are at least $\alpha|S|$ edges in G with one endpoint in S and the other not in S.

The performance of an algorithm is characterized by the time that it takes to balance the tokens, and by the final balance that it achieves. We say that an algorithm *globally balances* (or just *balances*) to within t tokens if the maximum difference in the number of tokens between any two nodes in the graph is at most t. We say that an algorithm *locally balances* to within t tokens if the maximum difference in the number of tokens between any two neighboring nodes in the graph is at most t.

We analyze two different types of algorithms in this paper: single-port and multi-port. In the *single-port* model,

[1]Informix Software, Menlo Park, CA 94025. Email: ghosh@informix.com. Work done while author was at the Department of Computer Science, Yale University, New Haven, CT 06520 and was supported by ONR Grant 4-91-1576 and a Yale/IBM joint study.

[2]Department of Mathematics and Laboratory for Computer Science, MIT, Cambridge, MA 02139. Supported by ARPA Contracts N00014-91-J-1698 and N00014-92-J-1799. Email: ftl@math.mit.edu.

[3]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Email: {aricha, bmm}@cs.cmu.edu.

[4]Supported in part by an NSF National Young Investigator Award, No. CCR-9457766, and by ARPA Contract F33615-93-1-1330.

[5]DIMACS, Rutgers University, Piscataway, NJ 08855. Supported by DIMACS, Center for Discrete Mathematics and Theoretical Computer Science, a National Science Foundation Science and Technology Center, under NSF Contract STC-8809648. Email: muthu@dimacs.rutgers.edu.

[6]Department of Computer Science, University of Texas at Austin, Austin, TX 78712.
Email: {diz, plaxton, rraj}@cs.utexas.edu.

[7]Supported by the Texas Advanced Research Program under Grant No. ARP-93-003658-461.

[8]Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08544, and NEC Research Institute. Research at Princeton University supported in part by the National Science Foundation, Grant No. CCR-8920505, and the Office of Naval Reserach, Contract No. N0014-91-J-1463. Email: ret@cs.princeton.edu.

[9]Supported in part by an NSF National Young Investigator Award, No. CCR-9457799.

a node may send and/or receive at most one token in one unit of time. In the *multi-port* model, a node may send and/or receive a token across all of its edges (there may be as many as d) in a single unit of time. Not surprisingly, the load balancing algorithms run faster in the multi-port model. In practice, however, single-port nodes may be prefered to multi-port nodes because they are easier and less costly to build.

1.1. Our results. This paper analyzes the simplest and most natural local algorithms in both the single-port and multi-port models.

In the single-port algorithm, a matching is randomly chosen at each step. First, each (undirected) edge in the network is independently selected to be a *candidate* with probability $1/(4d)$. Then each candidate edge (u, v) for which there is another candidate edge (u, x) or (y, v) is removed from the set of candidates. The remaining candidates form a matching M in the graph. For each edge (u, v) in M, if u and v have the same number of tokens, then nothing is sent across (u, v). Otherwise, a token is sent from the node with more tokens to the node with fewer. This algorithm was first analyzed in [14].

We analyze the performance of the single-port algorithm in terms of both the edge expansion and the node expansion of the graph. In terms of edge expansion, we show that the single-port algorithm balances to within $O((d \log n)/\alpha)$ tokens in $O(d\Delta/\alpha)$ steps, with high probability. In terms of node expansion, the final imbalance is $O((\log n)/\mu)$, and the time is $O(d\Delta/\mu)$, with high probability. (To compare these bounds, note that $\mu \leq \alpha \leq d\mu$.) The time bounds are tight in the sense that for many values of n, d, α, μ, and Δ, there is a graph with n nodes, maximum degree d, edge expansion α or node expansion μ, and an initial placement of tokens with imbalance Δ such that the time (for any algorithm) to balance to within even $\Delta/2$ tokens is at least $\Omega(d\Delta/\alpha)$. Similarly, in terms of node expansion, there exist classes of graphs where the time to balance to within even $\Delta/2$ tokens is at least $\Omega(d\Delta/\mu)$.

The multi-port algorithm is simpler and deterministic. At each step, a token is sent from node u to node v across edge (u, v) if at the beginning of the step node u contained at least $2d + 1$ more tokens than node v. This algorithm was first analyzed in [2].

As in the single-port case, we analyze the multi-port algorithm in terms of both edge expansion and node expansion. In terms of edge expansion, the algorithm balances to within $O((d^2 \log n)/\alpha)$ tokens in $O(\Delta/\alpha)$ steps. This bound is tight in the sense that for any network with edge expansion α, and any value Δ, there exists an initial distribution of tokens with imbalance Δ such that the time to reduce the imbalance to even $\Delta/2$ is $\Omega(\Delta/\alpha)$. In terms of node expansion, the algorithm balances to within $O((d \log n)/\mu)$ tokens in $O(\Delta/\mu)$ time. This bound is tight in the sense that for many values of d, n, and μ, and any value Δ, there exists an n-node, maximum degree d graph with node expansion μ and an initial distribution of tokens with imbalance Δ for which the time to balance to within $\Delta/2$ tokens is $\Omega(\Delta/\mu)$.

Both the single-port and multi-port algorithms will eventually locally balance the network, the single-port algorithm to within one token, and the multi-port algorithm to within $2d$ tokens. However, even after reducing the global imbalance to a small value, the time for either of these algorithms to reach a locally balanced state can be quite large. In particular, we show that after reaching a state that is globally balanced to within $O((d \log n)/\mu)$ tokens, the multi-port algorithm may take another $\Omega(n^{1/2})$ steps to reach a state that is locally balanced to within $2d$ tokens. For networks with large node expansion and small degree, e.g., $\mu = \Omega(1)$ and $d = O(1)$, and small initial imbalance, e.g., $\Delta = O((d \log^2 n)/\mu)$, the time to locally balance the network, $\Omega(n^{1/2})$, may be much larger than the time, $O(\Delta/\mu) = O((d \log^2 n)/\mu^2) = O(\log^2 n)$ to reach a state that is globally balanced to within $O((d \log n)/\mu)$ tokens. Moreover, there exist networks with diameter $\Theta(\log n/\mu)$ for which the even after reducing the global imbalance to the asymptotically best-possible value of $O(d \log n/\mu)$ tokens in optimal time, the multi-port algorithm can still take a long time to locally balance to within d tokens. We prove similar bounds for the single-port algorithm.

Thus far we have described a network model in which the nodes are synchronized by a global clock (i.e., a *synchronous* network), and in which the edges are assumed not to fail. With minor modifications, however, the load balancing algorithms can be made to work in both asynchronous and dynamic networks. In a *dynamic* network, the set of edges in the network may vary at each time step. In any time step, a *live* edge is one that can transmit one message in each direction. We assume that at each time step, each node in a synchronous dynamic network knows which of its edges are live. In an *asynchronous* network, the topology is fixed, but an adversary determines the speed at which each edge operates at every instant of time. For every undirected edge between two nodes, we allow at most two messages to be in transit at any instant in time. These messages may travel in opposite directions across the edge, or both may travel in one direction, while no message travels in the opposite direction. An edge is said to be *live* for a unit interval of time if every message that was in transit across the edge (in either direction) at the beginning of the interval is guaranteed to reach the end of the edge by the end of the interval. We analyze the performance of the multi-port load balancing algorithm under the assumption that at each time step, the set of live edges has some edge expansion α, or node expansion μ.

We also study the off-line load balancing problem, in which every node has knowledge of the global state of the network. This problem has been studied on static synchronous networks in [29]. We use their results to obtain tight bounds on off-line load balancing in terms of edge expansion and node expansion. For the single-port model, we prove that any network can be balanced off-line in $\lceil (1+\mu)\Delta/\mu \rceil$ steps so that no node has more than two tokens more than the average. This result can be used to show that any network can be balanced off-line to within three tokens in at most $2\lceil (1+\mu)\Delta/\mu \rceil$ steps in the single-port model. Moreover, there exists a network and an initial token distribution for which any single-port off-line algorithm takes more than $\lceil (1+\mu)\Delta/\mu \rceil$ steps to balance the network to within one token. Similarly, in the multi-port model, any network can be balanced off-line in at most $\lceil \Delta/\alpha \rceil$ steps so that no node contains more than d tokens more than the average. Using this result, we show that any network can be balanced to within $d+1$ tokens in at most $2\lceil \Delta/\alpha \rceil$ steps. It is easy to observe that for any network there exists an initial token distribution such that any algorithm will take at least $\lceil \Delta/\alpha \rceil$ steps to balance the network to within one token.

1.2. Previous and related work. Load balancing has been studied extensively because it comes up in a wide variety of settings including adaptive mesh partitioning [17, 39], fine-grain functional programming [16], job scheduling in operating systems [13, 25], and distributed tree searching [22, 26]. A number of models have been proposed for load balancing, differing chiefly in the amount of global information used by the load balancing algorithm [2, 11, 12, 14, 27, 31]. In these models, algorithms have been proposed for specific applications; also, proposed heuristics and algorithms have been analyzed using simulations and queuing–theoretic techniques [28, 35, 37]. In what follows, we focus on models that allow only local algorithms and on prior work that takes an analytical approach to the load balancing problem.

Local algorithms restricted to particular networks have been studied on counting networks [4, 23], hypercubes [20, 34], and meshes [17, 29]. Another class of networks on which load balancing has been studied is the class of expanders. Peleg and Upfal [32] pioneered this study by identifying certain small-degree expanders as being suitable for load balancing. Their work has been extended in [9, 18, 33]. These algorithms either use strong expanders to approximately balance the network, or the AKS sorting network [3] to perfectly balance the network. Thus, they do not work on networks of arbitrary topology. Also, these algorithms work by setting up fixed paths through the network on which load is moved and therefore cannot cope with changes in the network topology. In contrast, our local algorithm works on any arbitrary dynamic network that remains connected.

On arbitrary topologies, load balancing has been studied under two models. In the first model, any amount of load can be moved across a link in any time step [8, 12, 14, 15, 19, 36]. The second model is the one that we adopt here, namely one in which at most one unit load can be moved across a link in each time step. Load balancing algorithms for the second model were first proposed and analyzed in [2] for the multi-port variant and in [14] for the single-port variant. The upper bounds established by them are suboptimal by a factor of $\Omega(\log(n\Delta))$ or $\Omega(\sqrt{n})$, respectively. We improve these results for both single-port and multi-port variants.

As remarked earlier, our multi-port results (and those in [2]) hold even for dynamic or asynchronous networks. In general, work on dynamic and asynchronous networks has been limited. In work related to load balancing for instance, an end-to-end communication problem, namely one in which messages are routed from a single source to a single destination, has been studied in [1, 7] on dynamic networks. Our scenario is substantially more involved since we are required to move load between several sources and destinations simultaneously. Another result on dynamic networks is the recent analysis of a local algorithm for the approximate multicommodity flow problem [5, 6]. While their result has several applications including the end-to-end communication problem mentioned above, it does not seem to extend to load balancing. Our result on load balancing is related to their work in the technique; however, our algorithm and analysis are simpler and we obtain optimal bounds for our problem.

The convergence of local load balancing algorithms is related to that of random walks on Markov chains. Indeed the convergence bounds in both cases depend on the expansion properties of the underlying graph and they are established using potential function arguments. There are however two important differences. First, the analysis of the rapid convergence of random walks [21, 30] relies on averaging arbitrary probabilities across any edge. This corresponds to sending an arbitrary (possibly nonintegral) load along an edge which is forbidden in our model. In this sense, the analysis in [12] (and all references in the unbounded capacity model) are similar to the random walk analysis. Second, our argument uses an exponential potential function. The analyses in [12, 21, 30], in contrast, use quadratic potential functions. Our potential function and our amortized analysis were necessary since a number of previous attempts using quadratic potential functions yielded suboptimal results [2, 14] for local load balancing.

As mentioned earlier, we only consider the static aspect of load balancing. For a recent survey on the dynamic aspect of this problem (i.e., when tokens can be created or destroyed while the tokens are being balanced), see [40].

1.3. Outline. The remainder of this paper is organized as follows. Section 2 contains some definitions. Section 3.1 analyzes the performance of the single-port algorithm. Section 3.2 analyzes the performance of the multi-port algorithm. In Section 4, we show that the time to reach a locally balanced state can be quite large, even if the network starts in a state that is well balanced globally. Section 5 describes extensions to dynamic and asynchronous networks. Finally, Section 6 presents tight bounds on off-line load balancing.

2. Preliminaries. For any network $G = (V, E)$ with n nodes and edge expansion α, we denote the number of tokens at $v \in V$ by $w(v)$. We denote the average number of tokens by ρ, i.e., $\rho = (\sum_{v \in V} w(v))/n$. For simplicity, throughout this paper we assume that ρ is an integer. We assign a unique *rank* from $[1, w(v)]$ to every token at v. The *height* of a token is its rank minus ρ. The height of a node is the maximum among the heights of all its tokens.

Consider a partition of V given by $\{S_i\}$, where the index i is any integer (positive, negative, or zero) and S_i may be empty for any i. Let $S_{>j}$ be $\cup_{i>j} S_i$. Similarly we define $S_{\geq j}$, $S_{<j}$, and $S_{\leq j}$. We define index i to be *good* if $|S_i| \leq \alpha|S_{>i}|/2d$. An index that is not good is called a *bad* index. Thus, index i is good if there are at least $\alpha|S_{>i}|/2$ edges from nodes in $S_{>i}$ to nodes in $S_{<i}$. To observe this, note that the number of edges out of $S_{>i}$ is at least $\alpha|S_{>i}|$. On the other hand, the number of edges coming out of S_i is at most $d|S_i|$ which is at most $\alpha|S_{>i}|/2$ if i is good. Therefore, at least $\alpha|S_{>i}|/2$ edges must go from nodes in $S_{>i}$ to nodes in $S_{<i}$.

For any bad index i, it follows from the equality $|S_i| = |S_{>i-1}| - |S_{>i}|$ that $|S_{>i}| < |S_{>i-1}|/(1+\alpha/(2d))$. Consider the reduction in $|S_{>i}|$ as i increases. For each bad index, there is a reduction by a factor of $1/(1+\alpha/(2d))$. Hence, there can be at most $\lceil \log_{(1+\alpha/(2d))} n \rceil$ bad indices because $(1+\alpha/(2d))^{\log_{(1+\alpha/(2d))} n} \geq n$. It follows that at least half of the indices in $[1, 2\lceil \log_{(1+\alpha/(2d))} n \rceil]$ are good.

Note that for $0 \leq a \leq 1$, $1 + a \geq e^{a - a^2/2} \geq e^a/2$. Thus $\ln(1+a) \geq a/2$, implying $\log(1+a) = \Theta(a)$. We will use this result several times in the sections to follow, without further justification.

3. Analysis for static synchronous networks.

3.1. The single-port model. In this section, we analyze the single-port load balancing algorithm that is described in Section 1.1.

THEOREM 3.1. *For an arbitrary network G with n nodes, maximum degree d, edge expansion α, and initial imbalance Δ, the single-port algorithm balances within $O((d \log n)/\alpha)$ tokens in $O((d\Delta)/\alpha)$ steps, with high probability.*

For the sake of analysis, before every step we partition the set of nodes according to how many tokens they contain. For every integer i, we denote the set of nodes having $\rho + i$ tokens as S_i. Consider the first T steps of the algorithm, with T to be specified later. It holds that either $|S_{>0}| \leq n/2$ at the start of at least half the steps, or $|S_{<0}| \leq n/2$ at the start of at least half the steps. Without loss of generality, assume the former is true. Thus, every subset of nodes in $S_{>0}$ expands, and we will use this expansion property to show that the number of nodes that have at least $\rho + 2\log_{(1+\alpha/(2d))} n$ tokens rapidly goes to zero.

Recall that at least half of the indices in $[1, 2\lceil \log_{(1+\alpha/(2d))} n \rceil]$ are good in any time step. Therefore, there exists an index j in $[1, 2\lceil \log_{(1+\alpha/(2d))} n \rceil]$ that is good in at least half of those time steps in which $|S_{>0}| \leq n/2$. Hence j is good in at least $T/4$ steps.

With every token at height x we associate a potential of $\phi(x)$, where $\phi : N \to R$ is defined as follows:

$$\phi(x) = \begin{cases} 0 & \text{if } x \leq j, \\ (1+\nu)^x & \text{otherwise,} \end{cases} \tag{1}$$

where $\nu = \alpha/(cd)$, and $c > 1$ is a real constant to be specified later. The potential of the network is the sum of the potentials of all tokens in the network. While transmitting a token, every node sends its token with maximum height. Similarly, any token arriving at a node with height h is assigned height $h+1$. It follows from the definition of the potential function, and the fact that the height of a token never increases, that the potential of the network never increases. In the following, we show that during any step when j is good, the expected decrease in the potential of the network is at least an $\varepsilon\nu^2$ fraction of the potential before the step, where $\varepsilon > 0$ is a real constant to be specified later.

Before proving Theorem 3.1, we present an informal outline of the proof. For simplicity, let us assume that G is a constant-degree expander, i.e., $d = O(1)$ and $\mu = \Omega(1)$. Consider the scenario in which all of the indices greater than j are bad. In this situation, for indices greater than j, the size of the set $S_{\geq i}$ decreases exponentially with increasing i, and hence the number of tokens with height i decreases exponentially with increasing i. If the rate of growth of $\phi(x)$ with increasing x is smaller than the rate of decrease of $|S_{\geq i}|$ with increasing i, then the total potential due to tokens at height i "dominates" the total potential due to tokens at height greater than i. In such a case the potential of $S_{>j}$ is essentially a constant times the potential of tokens at height $j+1$. In addition, if the potential of tokens at height at most j is zero, then in every step when j is good, there is a constant fraction potential drop, because a constant fraction of the nodes in $S_{>j}$ send tokens to $S_{<j}$ in such a step. The exponential function we have defined in Equation (1) satisfies the properties described above for c sufficiently large.

In general, the indices greater than j may form any sequence of good and bad indices, provided that the upper bound on the number of bad indices is respected. We consider the indices greater than j in reverse order and show by an amortized analysis that for each index i we can "view" all indices greater than or equal to i as bad. If i is bad, then this view is trivially preserved; otherwise, the number of edges from $S_{>i}$ to $S_{<i}$ is at least $\alpha|S_{>i}|/2$ and hence there is a significant potential drop across the cut $(S_{\leq i}, S_{>i})$. This drop can be used to rearrange the potential of $S_{>i}$ in order to maintain the view that all indices greater than i are bad. We then invoke the argument for the case in which all indices greater than j are bad, and complete the proof.

Consider step t of the algorithm. Let Φ_t denote the potential of the network after step $t > 0$. Let M_i be the set of tokens that are sent from a node in $S_{>i}$ to a node in $S_{<i}$. Note that a token may appear in several different sets M_i. Let $m_i = |M_i|$. We say that a token p has an *i-drop* of $\phi(i+1) - \phi(i)$ if p moves from a node in $S_{>i}$ to a node in $S_{<i}$. Thus, the potential drop due to a token moving on an edge from node $u \in S_i$ to node $v \in S_{i'}$, $i > i'+1$, can be expressed as the sum of k-drops for $i' < k < i$. In Lemma 3.1, we use this notion of i-drops to relate the total potential drop in step t, Ψ, to the m_i's.

LEMMA 3.1.

$$\Psi = \left(\sum_{i>j} m_i \nu (1+\nu)^i\right) + m_j (1+\nu)^{j+1}.$$

Proof. Let M be the set of tokens that are moved from a node in $S_{>j}$. (Note that tokens that start from and end at nodes in $S_{>j}$ also belong to M.) For any token p, let $a(p)$ (resp., $b(p)$) be the height of p after (resp., before) step t.

$$\begin{aligned}
\Psi &= \sum_{p \in M} \left(\phi(b(p)) - \phi(a(p))\right) \\
&= \sum_{p \in M} \sum_{a(p) \leq i < b(p)} \left(\phi(i+1) - \phi(i)\right) \\
&= \sum_{i \geq j} \sum_{p \in M_i} \left(\phi(i+1) - \phi(i)\right) \\
&= \left(\sum_{i>j} \sum_{p \in M_i} \left(\phi(i+1) - \phi(i)\right)\right) + \sum_{p \in M_j} \left(\phi(j+1) - \phi(j)\right) \\
&= \left(\sum_{i>j} \sum_{p \in M_i} \nu (1+\nu)^i\right) + \sum_{p \in M_j} (1+\nu)^{j+1} \\
&= \left(\sum_{i>j} m_i \nu (1+\nu)^i\right) + m_j (1+\nu)^{j+1}.
\end{aligned}$$

(The second equation holds since the sum of $\phi(i+1) - \phi(i)$ over i telescopes. For the third equation, we interchange the order of summation and use the fact that $\phi(i)$ is zero for all $i \leq j$. The fourth equation is obtained by separating the case $i \geq j$ into two cases $i > j$ and $i = j$. For deriving the fifth equation, we use: (i) for all $i > j$, $\phi(i+1) - \phi(i) = \nu(1+\nu)^i$, and (ii) $\phi(j) = 0$. The last equation follows from the definition of m_i.) $\square$

We now describe the amortized analysis, which we alluded to earlier in this section, that we use to prove Theorem 3.1. We associate a charge of $\varepsilon \nu^2 \phi(h)$ with each token at height h. We show that we can pay for all of the charges using the expected potential drop $E[\Psi]$, which implies a lower bound on $E[\Psi]$. We consider the indices in $[j+1, \ell]$ in reverse order, where ℓ is the maximum token height. For every i in $[j, \ell]$, we maintain a "debt" term, given by Γ_i below, which is the difference between the charges due to tokens at height greater than i and the sum of i'-drops for $i' > i$. We will place an upper bound on $E[\Gamma_i]$ that lets us view all of the indices in $[i+1, \ell]$ as bad indicies. In other words, we upper bound $E[\Gamma_i]$ by $\varepsilon\nu|S_{\geq i}|(1+\nu)^i$. It follows from this upper bound and the informal argument outlined earlier in this section that the expected total debt can be paid for by the expected drop across index j.

Formally, for any $i > j$, we define

$$\Psi_i = \sum_{k \geq i} m_k \nu (1+\nu)^k, \text{ and}$$

$$\Gamma_i = (\varepsilon\nu^2)\left(\sum_{p:b(p)\geq i} (1+\nu)^{b(p)}\right) - \Psi_i.$$

We also define

$$\Gamma = (\varepsilon\nu^2)\left(\sum_{p:b(p)> j} (1+\nu)^{b(p)}\right) - \Psi.$$

Note that $\Phi_{t-1} = \sum_{p:b(p)>j} (1+\nu)^{b(p)}$ is the total potential of $S_{>j}$ prior to step t.

In order to prove the upper bound on $E[\Gamma_i]$, we place a lower bound on $E[m_i]$ that is obtained from the following lemma of [14].

Lemma 3.2 ([14]). *For any edge $e \in E$, the probability that e is selected in the matching is at least $1/(8d)$.* □

Lemma 3.3. *There exists a real constant $\varepsilon > 0$ such that for all $i > j$, we have $E[\Gamma_i] \leq (\varepsilon\nu)|S_{\geq i}|(1+\nu)^i$.*

Proof. The proof is by reverse induction on i. If $i > \ell$, then the claim holds trivially since Γ_i and $|S_{\geq i}|$ are both equal to zero. (Recall that ℓ denotes the maximum token height.) Therefore, for the base case we consider $i = \ell$. Since $m_\ell = 0$, we have $\Psi_\ell = 0$. Thus, $\Gamma_\ell = (\varepsilon\nu^2)|S_\ell|(1+\nu)^\ell \leq (\varepsilon\nu)|S_{\geq \ell}|(1+\nu)^\ell$, since $\nu = \alpha/(cd) \leq 1/c \leq 1$ by our choice of c.

For the induction step we consider two cases, depending on whether i is good or bad. We begin with the case when i is good. By the definition of a good index, we have $|S_i| \leq \alpha|S_{>i}|/2d$. Since each node has at most d adjacent edges, there are at most $\alpha|S_{>i}|/2$ edges adjacent to nodes in S_i. Therefore, there are at most $\alpha|S_{>i}|/2$ edges from $S_{>i}$ to S_i. By the expansion property of the graph, $S_{<i}$ has at least $\alpha|S_{<i}|$ edges to nodes in $S_{\geq i}$, so there are at least $\alpha|S_{>i}|/2$ edges from $S_{>i}$ to $S_{<i}$. By Lemma 3.2, we have $E[m_i] \geq \alpha|S_{>i}|/(16d)$.

We are now ready to place a bound on $E[\Gamma_i]$. By definition, Γ_i can be calculated by subtracting the sum of i-drops from Γ_{i+1} and adding the charges due to tokens at height i. Therefore, we have:

$$\begin{aligned}
E[\Gamma_i] &= E[\Gamma_{i+1}] + (\varepsilon\nu^2)|S_{\geq i}|(1+\nu)^i - E[m_i]\nu(1+\nu)^i \\
&\leq E[\Gamma_{i+1}] + (\varepsilon\nu^2)|S_{\geq i}|(1+\nu)^i - c\nu^2|S_{>i}|(1+\nu)^i/16 \\
&\leq E[\Gamma_{i+1}] - (\nu^2)|S_{\geq i}|(1+\nu)^i(f(c,\alpha,d) - \varepsilon) \\
&\leq (\varepsilon\nu)|S_{>i}|(1+\nu)^{i+1} - (\nu^2)|S_{\geq i}|(1+\nu)^i(f(c,\alpha,d) - \varepsilon) \\
&\leq (\varepsilon\nu)|S_{\geq i}|(1+\nu)^i((1+\nu) - \nu(f(c,\alpha,d) - \varepsilon)/\varepsilon),
\end{aligned}$$

where $f(c,\alpha,d) = c/(16(1+\alpha/(2d)))$. (In the first equation, we use the fact the number of tokens p such that $b(p) = i$ is $|S_{\geq i}|$. The second equation follows from the lower bound on $E[m_i]$. The third equation follows from the fact that $|S_{>i}| \geq |S_{\geq i}|/(1+\alpha/(2d))$ whenever i is a good index. The fourth equation follows from the induction hypothesis. The last equation follows from the fact that $|S_{>i}| \leq |S_{\geq i}|$.)

The second case is when i is bad. Thus $|S_i| > \alpha|S_{>i}|/(2d)$. We now place an upper bound on $E[\Gamma_i]$ as follows.

$$\begin{aligned}
E[\Gamma_i] &\leq E[\Gamma_{i+1}] + (\varepsilon\nu^2)|S_{\geq i}|(1+\nu)^i \\
&\leq (\varepsilon\nu)|S_{>i}|(1+\nu)^{i+1} + (\varepsilon\nu^2)|S_{\geq i}|(1+\nu)^i \\
&\leq (\varepsilon\nu)|S_{\geq i}|(1+\nu)^i((1+\nu)/(1+c\nu/2) + \nu).
\end{aligned}$$

(In the first equation, we use the fact the number of tokens p such that $b(p) = i$ is $|S_{\geq i}|$. The second equation follows from the induction hypothesis. The third equation follows from the fact that $|S_{\geq i}| > (1+\alpha/(2d))|S_{>i}|$ whenever i is a bad index.)

We now complete the induction step by determining values for c and ε such that the following equations hold.

$$((1+\nu)-\nu(f(c,\alpha,d)-\varepsilon)/\varepsilon)\leq 1 \tag{2}$$

$$(1+\nu)/(1+c\nu/2)+\nu\leq 1 \tag{3}$$

We set c to be any constant greater than or equal to $(\alpha/d)+4$ (e.g., $c=5$). For this choice of c, $\nu=\alpha/(cd)\leq(c-4)/c$, and hence $2\nu+c\nu^2/2\leq c\nu/2$. Therefore, we have:

$$\begin{aligned}(1+\nu)/(1+c\nu/2)+\nu&=(1+2\nu+c\nu^2/2)/(1+c\nu/2)\\&\leq(1+c\nu/2)/(1+c\nu/2)\\&=1.\end{aligned}$$

Thus, Equation (3) is satisfied. Since $\alpha\leq d$, we find that $f(c,\alpha,d)\geq c/24$. We now set $\varepsilon=c/48$ to establish Equation (2). (For example, $c=5$ and $\varepsilon=5/48$.) □

We are now in a position to bound $E[\Gamma]$ on those steps in which j is good. By applying Lemma 3.3 with $i=j+1$, we obtain that $E[\Gamma_{j+1}]\leq(\varepsilon\nu)|S_{\geq j+1}|(1+\nu)^{j+1}$. If j is good then by the definitions of Γ, Γ_{j+1}, and Ψ, we have:

$$\begin{aligned}E[\Gamma]&=E[\Gamma_{j+1}]-E[m_j](1+\nu)^{j+1}\\&\leq E[\Gamma_{j+1}]-\alpha|S_{>j}|(1+\nu)^{j+1}/(16d)\\&\leq(\varepsilon\nu)|S_{>j}|(1+\nu)^{j+1}-\alpha|S_{>j}|(1+\nu)^{j+1}/(16d)\\&=\nu|S_{>j}|(1+\nu)^{j+1}(\varepsilon-c/16)\\&\leq 0,\end{aligned}$$

(The second equation follows from the fact that $E[m_j]\geq\alpha|S_{>j}|/16d$ whenever j is good. The third equation follows from the upper bound on $E[\Gamma_{j+1}]$. The fifth equation holds since $c/16\geq\varepsilon$.)

We now derive a lower bound on the expected drop in the potential of the network during a sequence of T steps. By the definitions of Ψ and Γ, we have $\Phi_t=\Phi_{t-1}-\Psi$ and $\Gamma=\varepsilon\nu^2\Phi_{t-1}-\Psi$. If j is good during step t, we have $E[\Gamma]\leq 0$, and therefore, $E[\Phi_t]\leq\Phi_{t-1}(1-\varepsilon\nu^2)$, where the expectation is taken over the random matching selected in step t. Since j is good in at least $T/4$ steps, we obtain that $E[\Phi_{t+T}]\leq\Phi_t(1-\varepsilon\nu^2)^{T/4}$, where the expectation is over all the random matchings in the T steps. By setting $T=\lceil(4\ln 4)/(\varepsilon\nu^2)\rceil$, we obtain $E[\Phi_{t+T}]\leq\Phi_t/4$. By Markov's inequality, the probability that $\Phi_{t+T}\geq\Phi_t/2$ is at most $1/2$. Therefore, using standard Chernoff bounds [10], we can show that in $T'=8aT\lceil(\log\Phi_0+\log n)\rceil$ steps, $\Phi_{T'}>1$ with probability at most $O(1/(\Phi_0)^a+1/n^a)$ for any constant $a>0$.

If Δ is at most $2\log_{(1+\alpha/(2d))}n$, then the claim of the theorem holds trivially. Accordingly, we assume that Δ is greater than $2\log_{(1+\alpha/(2d))}n$ in what follows. Since Φ_0 is at least $(1+\nu)^\Delta$, Φ_0 is at least $n^{2/c}$. Therefore, $1/(\Phi_0)^a$ is inverse-polynomial in n. Since $\Phi_0\leq n(1+\nu)^{\Delta+1}/\nu$, we have $\log\Phi_0\leq(\Delta+1)(\nu)+\log n-\log\nu$. Therefore, for $T'=O(\Delta d/\alpha+d^2\log n/\alpha^2)$, we have $\Phi_{T'}<1$ with high probability which implies that after T' steps, $|S_{>2\log_{(1+\alpha/(2d))}n}|=0$ with high probability.

To establish balance in the number of tokens below the average, we use an averaging argument to show that after T' steps $|S_{<-2\log_{(1+\alpha/(2d))}n}|\leq n/2$ with high probability, and then repeat the above arguments with the potential redefined appropriately. This proves Theorem 3.1.

3.2. The multi-port model. In this section, we analyze the deterministic multi-port algorithm described in Section 1.1.

THEOREM 3.2. *For an arbitrary network G with n nodes, maximum degree d, edge expansion α, and initial imbalance Δ, the multi-port algorithm load balances to within $O((d^2\log n)/\alpha)$ tokens in $O(\Delta/\alpha)$ steps.*

The proof of Theorem 3.2 is similar to that of Theorem 3.1. We assign a potential to every token, where the potential is exponential in the height of the token. We then show by means of an amortized analysis that a suitable rearrangement of the potential reduces every instance of the problem to a special instance that we understand well.

For the sake of analysis, before every step we partition the set of nodes according to how many tokens they contain. For every integer i, we denote the set of nodes having between $\rho-d+2id$ and $\rho+d-1+2id$ tokens as S_i. (Recall that ρ is the average number of tokens per node.) Consider the first T steps of the algorithm, with T to be specified later. Without loss of generality, we assume that $|S_{>0}|\leq n/2$ holds in at least half of these steps. As shown in Section 2, there exists an index j in $[1,2\lceil\log_{(1+\alpha/(2d))}n\rceil]$ that is good in at least half of those steps in which $|S_{>0}|\leq n/2$. Hence in T steps of the algorithm, j is good in at least $T/4$ steps.

With every token at height h we associate a potential of $\phi(h)$, where $\phi : N \rightarrow R$ is defined as follows:

$$\phi(x) = \begin{cases} 0 & \text{if } x \leq 2jd, \\ (1+\nu)^x & \text{otherwise,} \end{cases}$$

where $\nu = \alpha/(cd^2)$, and $c > 0$ is a constant to be specified later. The potential of the network is the sum of the potentials of all tokens in the network.

While transmitting some number (say m) of tokens in a particular step, a node sends the m highest-ranked tokens. Similarly, if m tokens arrive at a node during a step, they are assigned the m highest ranks within the node. Thus, tokens that do not move retain their ranks after the step. We now describe what specific ranks we assign to tokens that move during any step t. Let u be a node in $S_{<i}$ with height h at the start of step t. Let A (resp., B) be the set of tokens that u receives from nodes in $S_{>i}$ (resp., $S_{\leq i}$). We assign new ranks to tokens in A and B such that the rank of every token in A is less than that of every token in B. Let C be the set of tokens in A that attain height at most $h + (d/2)$ after the step. Since $|A| \leq d$, by the choice of our ranking, we have $|C| \geq |A|/2$. We call C the set of *primary* tokens. We also note that for any node v with height h all tokens leaving v during a step are at height at least $h - d + 1$ prior to the step.

It follows from the definition of the potential function and the fact that the height of a token never increases that the network potential never increases. In the following we show that whenever j is good the potential of $S_{>j}$ decreases by a factor of $\varepsilon\nu^2 d^2$, where $\varepsilon > 0$ is a real constant to be specified later. (For the sake of simplicity, we assume that d is even. If d is odd, we can replace d by $d+1$ in our argument without affecting the bounds by more than constant factors.)

For any token p, let $a(p)$ (resp., $b(p)$) be the index i such that S_i contains p after (resp., before) the step. (Note that the indexing is done prior to the step.) Let M_i be the set of primary tokens received by nodes in $S_{<i}$. Let $m_i = |M_i|$. Note that m_i is at least half the number of edges connecting nodes in $S_{<i}$ and nodes in $S_{>i}$. This is because a token is sent along every one of the edges connecting $S_{<i}$ and $S_{>i}$ and at least half the tokens received by any node in $S_{<i}$ from nodes in $S_{>i}$ are primary tokens. Lemma 3.4 establishes the relationship between the total potential drop Ψ in step t and the m_i's.

LEMMA 3.4.

$$\Psi \geq \left(\frac{1}{2} \sum_{i>j} m_i \nu d (1+\nu)^{(2i-1)d} \right) + m_j (1+\nu)^{2jd+1}.$$

Proof. Let M be the set of primary tokens that are moved from nodes in $S_{>j}$. (Note that primary tokens that start from a node in $S_{>j}$ and end at a node in $S_{>j}$ are in M.) Let p be a token in M. By the definition of a primary token, the height of p prior to the step is at least $2b(p)d - 2d + 1$ and the height after the step is at most $2a(p)d + 3d/2$. Moreover, p belongs to M_i for all i such that $a(p) < i < b(p)$.

$$\begin{aligned}
\Psi &\geq \sum_{p \in M} [\phi(2b(p)d - 2d + 1) - \phi(2a(p)d + 3d/2)] \\
&\geq \sum_{p \in M} \sum_{a(p) < i < b(p)} [\phi(2(i+1)d - 2d + 1) - \phi(2(i-1)d + 3d/2)] \\
&= \sum_{i \geq j} \sum_{p \in M_i} [\phi(2(i+1)d - 2d + 1) - \phi(2(i-1)d + 3d/2)] \\
&= \sum_{i > j} \sum_{p \in M_i} [\phi(2(i+1)d - 2d + 1) - \phi(2(i-1)d + 3d/2)] \\
&\quad + \sum_{p \in M_j} [\phi(2(j+1)d - 2d + 1) - \phi(2(j-1)d + 3d/2)] \\
&\geq \left(\frac{1}{2} \sum_{i>j} \sum_{p \in M_i} \nu d (1+\nu)^{2id-d} \right) + \sum_{p \in M_j} (1+\nu)^{2jd+1} \\
&\geq \left(\frac{1}{2} \sum_{i>j} m_i \nu d (1+\nu)^{2id-d} \right) + m_j (1+\nu)^{2jd+1}.
\end{aligned}$$

(The first equation follows from the lower bound (resp., upper bound) on the height of a token p in M before (resp., after) the step. For the second equation, note that $2id - 2d + 1 \le 2(i-1)d + 3d/2$. Therefore, $\phi(2id - 2d + 1) \le \phi(2(i-1)d + 3d/2)$. The second equation now follows since the sum telescopes. The third equation is obtained by interchanging the sums and noting that $\phi(x)$ is 0 for $x \le 2jd$. The fourth equation is obtained by partitioning M into the subsets $M \setminus M_j$ and M_j. The fifth equation is derived using the following calculations: (i) $\phi(2id+1) - \phi(2id - d/2) \ge ((1+\nu)^{d/2} - 1)(1+\nu)^{2id-d/2} \ge \nu d(1+\nu)^{2id-d}/2$, (ii) $\phi(2jd+1) = (1+\nu)^{2jd+1}$, and (iii) $\phi(2jd - d/2) = 0$. The last equation follows from the definition of m_i.) $\square$

We establish Theorem 3.2 by means of an amortized analysis similar to the one used in Section 3.1. We associate a charge of $\varepsilon\nu^2 d^2 \phi(h)$ with every token at height h. We show that we can pay for all of the charges using the potential drop Ψ and thus place a lower bound on Ψ. We consider the sets S_i in reverse order and maintain a "debt" term Γ_i for each i. Informally, Γ_i indicates the difference between the total charges due to tokens at height at least $2id - d$ and the current upper bound on the potential drop. Our amortized analysis terminates by showing that the total debt Γ is at most zero.

We now formally define Γ_i and Γ. For any token p, let $h(p)$ denote the height of p prior to the step. Thus $2b(p)d - d \le h(p) \le 2b(p)d + d - 1$. For $i > j$ and for a suitable constant $\varepsilon > 0$ to be specified later, we define

$$\Psi_i = \frac{1}{2}\sum_{k \ge i} m_k \nu d(1+\nu)^{2kd-d}, \text{ and}$$

$$\Gamma_i = (\varepsilon\nu^2 d^2)\left(\sum_{p:h(p)\ge 2id-d} (1+\nu)^{h(p)}\right) - \Psi_i.$$

We also define

$$\Gamma = (\varepsilon\nu^2 d^2)\left(\sum_{p:h(p)>2jd} (1+\nu)^{h(p)}\right) - \Psi.$$

For any step t', let $\Phi_{t'}$ denote the total potential after step t'. Thus, $\Phi_{t-1} = \sum_{p:h(p)>2jd} (1+\nu)^{h(p)}$ is the total potential prior to step t.

Lemma 3.5. *There exists a real constant $\delta > 0$ such that for all $i > j$, we have*

$$\Gamma_i \le (\delta\nu d^2)|S_{\ge i}|(1+\nu)^{2id-d}.$$

Proof. The proof is by reverse induction on i. Let ℓ be the maximum token height. Consider first the case when $i > \lfloor(\ell+d)/2d\rfloor$. Since $2id - d > \ell$, there is no token with height at least $2id - d$. Hence $\Gamma_i \le 0$ and $|S_{\ge i}| = 0$. Thus, the desired claim holds. We now consider $i = \lfloor(\ell+d)/2d\rfloor$. Since $\Psi_i = 0$, we have

$$\begin{aligned}\Gamma_i &\le (2\varepsilon\nu^2 d^3)|S_{\ge i}|(1+\nu)^{\ell}\\ &\le (2\varepsilon\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2d(i+1)-d}\\ &\le (\delta\nu d^2)|S_{\ge i}|(1+\nu)^{2id-d}.\end{aligned}$$

(The first equation holds because: (i) each node in S_i has at most $2d$ tokens with height at least $2id - d$, and (ii) $h(p) \le \ell$ for each token p. The second equation follows from the fact that $\ell < 2(i+1)d - d$. The third equation is obtained by choosing δ and ε such that $\delta > 2\varepsilon\nu d(1+\nu)^{2d}$. Note that for c sufficiently large, $(1+\nu)^{2d}$ can be set to an arbitrarily small constant.)

For the induction step we consider two cases. If i is good, then $|S_i| \le \alpha|S_{>i}|/(2d)$ and $m_i \ge \alpha|S_{>i}|/4$. Therefore, we have

$$\begin{aligned}\Gamma_i &\le \Gamma_{i+1} + (2\varepsilon\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2id+d-1} - m_i\nu d(1+\nu)^{2id-d}/2\\ &\le \Gamma_{i+1} + (2\varepsilon\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2id+d-1} - c\nu^2 d^3|S_{>i}|(1+\nu)^{2id-d}/8\\ &\le \Gamma_{i+1} - (\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2id-d}(f(c,\alpha,d) - 2\varepsilon(1+\nu)^{2d})\\ &\le (\delta\nu d^2)|S_{>i}|(1+\nu)^{2(i+1)d-d} - (\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2id-d}(f(c,\alpha,d) - 4\varepsilon)\\ &\le (\delta\nu d^2)|S_{\ge i}|(1+\nu)^{2id-d}((1+\nu)^{2d} - \nu d(f(c,\alpha,d) - 4\varepsilon)/\delta),\end{aligned}$$

where $f(c,\alpha,d) = c/(8(1+\alpha/(2d)))$. (The first equation holds because: (i) each node in S_i has at most $2d$ tokens with height at least $2id-d$, and (ii) $h(p) \le 2id+d-1$ for each token p that contributes to Γ_i and not to Γ_{i+1}. The third equation follows from the fact that $|S_{>i}| \ge |S_{\ge i}|/(1+\alpha/(2d))$. The fourth equation follows from the induction hypothesis and the equation $(1+\nu)^{2d} \le 2$ for c sufficiently large. The last equation is derived using straightforward algebra.)

The second case is when i is bad. Thus $|S_i| > \alpha|S_{>i}|/(2d)$. We have

$$\begin{aligned}\Gamma_i &\le \Gamma_{i+1} + (2\varepsilon\nu^2 d^3)|S_{\ge i}|(1+\nu)^{2id+d-1}\\ &\le (\delta\nu d^2)|S_{>i}|(1+\nu)^{2(i+1)d-d} + 2\varepsilon\nu^2 d^3|S_{\ge i}|(1+\nu)^{2id+d-1}\\ &\le (\delta\nu d^2)|S_{\ge i}|(1+\nu)^{2id-d}((1+\nu)^{2d}/(1+\alpha/(2d)) + 2\varepsilon\nu d(1+\nu)^{2d}/\delta).\end{aligned}$$

We now set c, δ, and ε such that $c > 4$, $c/12 - 4\varepsilon \ge 4\delta$, and $c/4 - 2\varepsilon/\delta \ge 4$. (One set of choices is $c = 50$, $\delta = 1$, and $\varepsilon = 1/24$.) Since $\alpha \le d$, we have $f(c,\alpha,d) \ge c/12$. Since $c > 4$, we have $2\nu d < 1/2$, and hence $(1+\nu)^{2d} \le 1 + \sum_{i>0}(2\nu d)^i = 1 + 2\nu d/(1-2\nu d) \le 1 + 4\nu d$. Thus,

$$((1+\nu)^{2d} - \nu d(f(c,\alpha,d) - 4\varepsilon)/\delta) \le 1 + 4\nu d - 4\nu d \le 1.$$

Since $\alpha/(2d) \le 1/2$, we have $1/(1+\alpha/(2d)) \le 1 - \alpha/2d + (\alpha/2d)^2 \le 1 - \alpha/(2d) + \alpha/(4d) = 1 - \alpha/(4d)$, and hence,

$$\begin{aligned}(1+\nu)^{2d}/(1+\alpha/(2d)) + 2\varepsilon\nu d(1+\nu)^{2d}/\delta &= (1+\nu)^{2d}(1/(1+\alpha/(2d)) + 2\varepsilon\nu d/\delta)\\ &\le (1+\nu)^{2d}(1 - \alpha/4d + 2\varepsilon\nu d/\delta)\\ &= (1+\nu)^{2d}(1 - c\nu d/4 + 2\varepsilon\nu d/\delta)\\ &\le (1+4\nu d)(1-4\nu d) < 1.\end{aligned}$$

(The second equation follows from the upper bound on $1/(1+\alpha/(2d))$. The fourth equation follows from the upper bound of $(1+4\nu d)$ on $(1+\nu)^{2d}$.)

Thus, in both cases, $\Gamma_i \le (\delta\nu d^2)|S_{\ge i}|(1+\nu)^{2id-d}$. This completes the induction step. □

Corollary 3.5.1. *If j is good on step t then we have $\Psi \ge \varepsilon\nu^2 d^2 \Phi_{t-1}$.*

Proof. Applying Lemma 3.5 with $i = j+1$, it follows that $\Gamma_{j+1} \le (\delta\nu d^2)|S_{\ge j+1}|(1+\nu)^{2(j+1)d-d}$. If j is good then $|S_{\ge j}| \le (1+\alpha/(2d))|S_{>j}| \le 3|S_{>j}|/2$ and $m_j \ge \alpha|S_{>j}|/2$. Therefore,

$$\begin{aligned}\Gamma &\le \Gamma_{j+1} + \varepsilon\nu^2 d^3|S_{\ge j}|(1+\nu)^{2jd+d-1} - \alpha|S_{>j}|(1+\nu)^{2jd+1}/2\\ &\le (\delta\nu d^2)|S_{>j}|(1+\nu)^{2(j+1)d-d} + (3\varepsilon\nu^2 d^3)|S_{>j}|(1+\nu)^{2jd+d-1}/2 - \alpha|S_{>j}|(1+\nu)^{2jd+1}/2\\ &\le (\nu d^2)|S_{>j}|(1+\nu)^{2(j+1)d-d}(\delta + 3\varepsilon\alpha/(2cd) - c/4)\\ &\le 0,\end{aligned}$$

for c, δ, and ε chosen above. (In the first equation, the term $\varepsilon\nu^2 d^3|S_{\ge j}|(1+\nu)^{2jd+d-1}$ is an upper bound on the contribution to Γ_j by tokens in $S_{\ge j}$ since: (i) tokens with height at least $2jd+d$ contribute to Γ_{j+1}, and (ii) each node in $S_{\ge j}$ has $d-2 \le d$ tokens with height in the interval $[2jd+1, 2jd+d-1]$. Also, the third term in the first equation is the second term in the right-hand side of the equation of Lemma 3.4. In the second equation, we use the upper bounds on Γ_{j+1} and $|S_{\ge j}|$. The third equation follows from the choice of c, δ, and ε, and the fact that for $c > 4$, we have $(1+\nu)^d \le (1+\alpha/(cd^2))^d \le (1+1/(cd))^d < (1+1/(4d))^d \le e^{1/4} \le 2$.)

By the definitions of Γ and Ψ, we have $\Phi_t \le \Phi_{t-1} - \Psi$ and $\Gamma = \varepsilon\nu^2 d^2\Phi_{t-1} - \Psi$. If j is good during step t, then $\Gamma \le 0$, and the desired claim follows. □

By Corollary 3.5.1, if j is good during step t then we have

$$\Phi_t \le \Phi_{t-1}(1 - \varepsilon\nu^2 d^2).$$

After $T = \lceil 4\ln\Phi_0/(\varepsilon\nu^2 d^2)\rceil$ steps, we have $\Phi_T \le \Phi_0(1-\varepsilon\nu^2 d^2)^{T/4} < 1$. Since the height of each node is at most Δ initially, $\Phi_0 \le n\sum_{2jd<i\le\Delta}(1+\nu)^i \le n(1+\nu)^{\Delta+1}/\nu$, $\ln\Phi_0 = O(\Delta\nu + \log n)$. Substituting $\alpha/(cd^2)$ for ν, we obtain that within $O(\Delta/\alpha + d^2\ln n/\alpha^2)$ steps, $|S_{>2\log_{(1+\alpha/(2d))} n}| \le |S_{>j}| = 0$.

We use an averaging argument to show that after T steps, $|S_{<-2\log_{(1+\alpha/(2d))} n}| \le n/2$. By redefining the potential function and repeating the above analysis in the other direction, we obtain that in another T steps $|S_{<-4\log_{(1+\alpha/(2d))} n}| = 0$. This completes the proof of Theorem 3.2.

3.3. Results in terms of node expansion. The proofs of Theorems 3.1 and 3.2 can be easily modified to analyze the algorithm in terms of the node expansion μ of the graph instead of the edge expansion α. Recall that μ and α are related by the following inequalities: $\alpha/d \le \mu \le \alpha$. The primary modifications that need to be done to obtain bounds in terms of node expansion are to change the definition of a good index and to set ν appropriately. We call index i good if $|S_i| \le \mu|S_{>i}|/2$. We set $\nu = \mu/c$ (resp., $\nu = \mu/(cd)$) for the single-port model (resp., multi-port model).

By an argument similar to the one used in Section 2, we obtain that the number of bad indices is at most $\lceil \log_{(1+\mu)} n \rceil$. (In fact, the argument in Section 2 uses α/d as a lower bound on μ.) This bound on the number of bad indices leads to an upper bound of $O((\log n)/\mu)$ (resp., $O(d(\log n)/\mu)$) on the final imbalance obtained by the single-port algorithm (resp., multi-port algorithm). For a bound on the number of steps, note that while deriving a bound on the potential drop in Sections 3.1 and 3.2, we use the edge expansion α to obtain a lower bound on the number of tokens leaving sets $S_{>i}$. Since the best lower bound on α in terms of node expansion is μ, our time bounds here are obtained by substituting μ for α in the time bounds of Theorems 3.1 and 3.2, respectively. We thus obtain Theorems 3.3 and 3.4. Finally, Corollary 3.3.1 (resp., Corollary 3.4.1) follows from Theorems 3.1 and 3.3 (resp., Theorems 3.2 and 3.4).

THEOREM 3.3. *For an arbitrary network G with n nodes, maximum degree d, node expansion μ, and initial imbalance Δ, the single-port algorithm balances to within $O((\log n)/\mu)$ tokens in $O(d\Delta/\mu)$ steps with high probability.*

COROLLARY 3.3.1. *If $\Delta \ge (d \log n)/\mu$, the single-port algorithm balances to within $O(\log n/\mu)$ tokens in $O((d\Delta)/\alpha)$ steps with high probability. If $\Delta < (d \log n)/\mu$, the single-port algorithm balances to within $O(\log n/\mu)$ tokens in $O((d\Delta)/\mu)$ steps with high probability.*

THEOREM 3.4. *For an arbitrary network G with n nodes, maximum degree d, node expansion μ, and initial imbalance Δ, the multi-port algorithm load balances to within $O((d \log n)/\mu)$ tokens in $O(\Delta/\mu)$ steps.*

COROLLARY 3.4.1. *If $\Delta \ge (d^2 \log n)/\mu$, the multi-port algorithm balances to within $O((d \log n)/\mu)$ tokens in $O(\Delta/\alpha)$ steps. If $\Delta < (d^2 \log n)/\mu$, the multi-port algorithm balances to within $O((d \log n)/\mu)$ tokens in $O(\Delta/\mu)$ steps.*

4. Local load balancing can be expensive. Here we show that upon reaching a state with small global imbalance, the algorithms presented in this paper may still take many steps until they reach a locally balanced state. More specifically, in Section 4.1, we show that locally load balancing to within $2d$ tokens using the multi-port algorithm of [2] described in Section 1.1 can take $\Omega(\sqrt{n})$ more time than globally load balancing to within $O((d \log n)/\mu)$ tokens. We extend this bound to the single-port algorithm presented in [14]; i.e., upon reaching a state where the network is globally balanced to within $O(\log n/\mu)$ tokens, the expected number of additional steps this algorithm may take to perform local balancing (to within one token) is $\Omega(d\sqrt{n})$. Furthermore, in Section 4.2 we show that in the single-port case, the network may be one step away from being locally balanced to within one token but have an expected running time of $\Omega(\mu\sqrt{n})$ for reaching a locally balanced state. Finally, we prove upper bounds on the time each algorithm takes to reach a locally balanced state in Section 4.3.

All results in this section are stated in terms of the node expansion of the network, rather than in terms of its edge expansion. This is done for the sake of making our arguments more intuitive and clear. Similar bounds can be derived in terms of edge expansion.

For any positive n and any μ, $0 < \mu < 1/72$, we present an example of a graph $G = (V, E)$ on n nodes, with node expansion at least μ, and with maximum degree d that depends on μ, where, given some initial distribution of tokens, locally balancing is difficult.

First we define the node set V of G. Let μ_0 be equal to $\sqrt{8\mu}$ (note that $0 < \mu_0 < 1/3$). Let V be equal to $(\cup_{i=0}^{k} L_i) \cup (\cup_{i=0}^{k-1} R_i)$, where L_i and R_i are disjoint sets of $(1+\mu_0)^i$ nodes each, and k will be specified shortly. For simplicity, we shall ignore integrality constraints on the number of nodes in each set. We could be more formal by setting the size of each set L_i or R_i to be $\lceil (1+\mu_0)^i \rceil$, but this would only make the calculations in this section more involved, without changing the asymptotic results. For convenience, let L_{-1} (resp., R_{-1}) denote R_0 (resp., L_0) and R_k denote L_k. Note that each L_i and each R_i has size $(1+\mu_0)^i = \mu_0(\sum_{j=0}^{i-1} |L_j|) + 1 = \mu_0(\sum_{j=0}^{i-1} |R_j|) + 1$. Thus, setting $n = \sum_{j=0}^{k} |L_j| + \sum_{j=0}^{k-1} |R_j|$ and solving for k, we have $k = \Theta((\log n)/\log(1+\mu_0)) = \Theta((\log n)/\log(1+\mu)) = \Theta((\log n)/\mu)$. For simplicity, we assume that $k = (\log n)/\log(1+\mu)$ and that k is even.

Given μ, we can choose the maximum degree d of G, independent of n, such that the following construction of the edges in G is possible. We obtain a similar structure for the R_i's by replacing L_j by R_j below. Let the only node in L_0 be adjacent to every node in L_1. For all i, $0 \le i \le k$, we insert the edges between nodes in L_i such that (i) there are at most $d/2$ such edges adjacent to any node in L_i, and (ii) every subset S of L_i of size less than or equal to $2|L_i|/3$ has at least $\mu_0|S|$ neighbors in $L_i \setminus S$ (see [24, 38] for a proof that such a construction is possible). Also, let each node in L_i have exactly $d(1+\mu_0)/(2(2+\mu_0))$ neighbors in L_{i+1} and each node in L_{i+1} have exactly $d/(2(2+\mu_0))$ neighbors in L_i, $0 < i \leq k-1$. For simplicity, we again ignore integrality constraints. Let S be any subset of L_i. There are exactly $(|S|d(1+\mu_0))/(2(2+\mu_0))$ edges between S and L_{i+1}, and each node in L_{i+1} has exactly $d/(2(2+\mu_0)$ neighbors in L_i. Thus, S has at least $(1+\mu_0)|S|$ neighbors in L_{i+1}.

Now we consider how L_{i+1} "expands" into L_i. We can use an approach similar to that of [24, 38] to show that we can choose the edges between L_i and L_{i+1}, respecting the degree constraints, such that any subset S of L_{i+1} of size less than or equal to $3|L_{i+1}|/(4(1+\mu_0))$ has at least $(1+\mu_0)|S|$ neighbors in L_i. This construction is possible since $(1+\mu_0)3|L_{i+1}|/(4(1+\mu_0)) = 3(1+\mu_0)|L_i|/4 < |L_i|$. The same analysis as in [24], but for a bipartite graph with node sets of sizes $|L_{i+1}|$ and $|L_{i+1}|/(1+\mu_0)$, and of regular node degrees $d/(2(2+\mu_0))$ and $d(1+\mu_0)/(2(2+\mu_0))$, respectively, applies here.

To complete the edge construction of G, let u be the only node in L_0 and let v be the only node in R_0, and add the edge $e = (u, v)$ to the set E. Note that the diameter of G is $\Theta(k) = \Theta(\log n/\mu)$.

We give a pictorial representation of the sets R_i's and L_i's in Figure 2. The initial distribution of tokens in Figure 2 (given by the quantities above the ovals representing each set L_i or R_i) may be ignored for the moment.

We still need to show that the graph G has node expansion at least μ, as claimed.

THEOREM 4.1. *The graph G, constructed as described, has node expansion at least μ.*

Proof. We will show how to account for the node expansion of any subset of G of at most $n/2$ nodes. Let U be a subset of V of size at most $n/2$. We will show that there exists a set of at least $\mu|U| = \mu_0^2|U|/8$ nodes outside of U that are all adjacent to nodes in U. For any subsets X and Y of V, we define the neighborhood of X in Y, $N_Y(X)$, as the subset of nodes in Y, but not in X, that are adjacent to some node in X, i.e., $N_Y(X) = \{y \in (Y \setminus X) : (x,y) \in E, x \in X\}$. If the set Y is not specified, assume $Y = V$. Let $U_i^L = U \cap L_i$ and $U_i^R = U \cap R_i$, for all $0 \leq i \leq k$. We consider two cases, according to whether the size of U_k^L is greater than $2|L_k|/3$ or not.

(1) If $|U_k^L| > 2|L_k|/3$, then let W^L (resp., W^R) be the union of the sets U_j^L (resp., U_j^R) of size greater than $2|L_j|/3$ (resp., $2|R_j|/3$) such that there is no U_q^L (resp., U_q^R), $q > j$, of size less than or equal to $2|L_q|/3$ (resp., $2|R_q|/3$). Let ℓ (resp., r) be the minimum index of a set U_j^L in W^L (resp., U_t^R in W^R). In case no such j (resp., t) exists, let $\ell = 0$ (resp., $r = 0$). Let $\mathcal{S}$ denote $\left(\cup_{j=\ell}^k L_j\right) \cup \left(\cup_{j=r}^{k-1} R_j\right)$, and let $W = W^L \cup W^R$. Note that since $|U_j^L| > 2|L_j|/3$, for all $j \geq \ell$, and $|U_j^R| > 2|R_j|/3$, for all $j \geq r$, there are at most $3|W|/2$ nodes in $\mathcal{S}$. Furthermore, since the set U has at most $n/2$ nodes, there are at most $3n/4$ nodes in $\mathcal{S}$. Hence, there are at least $n/4$ nodes that are not in $\mathcal{S}$, and so we must have either $\sum_{i=0}^{\ell-1}|L_i| \geq n/8$ or $\sum_{i=0}^{r-1}|R_i| \geq n/8$. Assume w.l.o.g. that the former is true. This implies that $|L_\ell| > \mu_0 n/8$.

We will account for the node expansion of U using the neighborhood of U_ℓ^L in $L_{\ell-1} \setminus U_{\ell-1}^L$. If $|U_\ell^L| < 3|L_\ell|/(4(1+\mu_0))$ (implying that $\mu_0 < 1/8$, since $|U_\ell^L| > 2|L_\ell|/3$), then

$$|N_{L_{\ell-1}}(U_\ell^L) \setminus U_{\ell-1}^L| \geq \frac{(1+\mu_0)2|L_\ell|}{3} - \frac{2|L_{\ell-1}|}{3} = \frac{2|L_\ell|}{3}\left((1+\mu_0) - \frac{1}{(1+\mu_0)}\right)$$
$$> \frac{2|L_\ell|\mu_0(2+\mu_0)}{3(1+\mu_0)} > \mu_0|L_\ell| > \frac{\mu_0^2 n}{8}$$

(the second to last inequality follows from $(2+\mu_0)/(1+\mu_0) > 3/2$). Else, any subset of $|U_\ell^L|$ of size $3|L_\ell|/(4(1+\mu_0))$ has at least $(1+\mu_0)3|L_\ell|/(4(1+\mu_0))$ neighbors in $L_{\ell-1}$. Thus

$$|N_{L_{\ell-1}}(U_\ell^L) \setminus U_{\ell-1}^L| \geq \frac{(1+\mu_0)3|L_\ell|}{4(1+\mu_0)} - \frac{2|L_{\ell-1}|}{3} = |L_\ell|\left(\frac{3}{4} - \frac{2}{3(1+\mu_0)}\right)$$
$$> \frac{9\mu_0|L_\ell|}{12(1+\mu_0)} > \frac{\mu_0|L_\ell|}{2} > \frac{\mu_0^2 n}{16}$$

We obtained the second to last inequality by substituting $(1+\mu_0)$ by $4/3$ in the denominator of the left-hand side of the inequality.

Hence, in case (1) we have at least $\mu_0^2 n/16 \geq \mu_0^2|U|/8$ nodes not in U that are adjacent to the nodes in U.

(2) If $|U_k^L| \leq 2|L_k|/3$, then each set U_i^L and each set U_j^R is considered in exactly one of the following sub-cases. We prove the results that follow for the sets U_j^L's only. Similar results hold if we replace U_j^L by U_j^R, and L_j by R_j in the two sub-cases below.

(2.1) Let i be the maximum index such that $|U_i^L| > 2|L_i|/3$ and $|U_{i+1}^L| \leq 2|L_{i+1}|/3$. If $|U_{i+1}^L| \leq |L_{i+1}|/3$, then the neighbors of U_i^L in L_{i+1} that are not in U_{i+1}^L can account for the node expansion of $(\cup_{j=0}^{i+1} U_j^L)$, because

$$
\begin{aligned}
|N_{L_{i+1}}(U_i^L) \setminus U_{i+1}^L| &> \frac{(1+\mu_0)2|L_i|}{3} - \frac{|L_{i+1}|}{3} = \frac{|L_{i+1}|}{3} \\
&> \frac{\mu_0}{3}\left(\sum_{j=0}^{i} |L_j|\right) \geq \frac{\mu_0}{4}\left(\left(\sum_{j=0}^{i} |L_j|\right) + |L_{i+1}| - 1\right) \\
&> \frac{\mu_0}{4}\left(\sum_{j=0}^{i+1} |U_j^L|\right)
\end{aligned}
$$

(the second to last inequality follows from the fact that $|L_{i+1}| = \mu_0(\sum_{j=0}^{i} |L_j|) + 1 \leq (\sum_{j=0}^{i} |L_j|)/3$, since $1/n < \mu_0 < 1/3$). Else, $|L_{i+1}|/3 < |U_{i+1}^L| \leq 2|L_{i+1}|/3$, and the neighborhood of U_{i+1}^L in L_{i+1} can account for the node expansion of $(\cup_{j=0}^{i+1} U_j^L)$, since

$$
\begin{aligned}
|N_{L_{i+1}}(U_{i+1}^L)| \geq \mu_0 |U_{i+1}^L| &> \frac{\mu_0 |L_{i+1}|}{3} > \frac{\mu_0^2}{3}\left(\sum_{j=0}^{i} |L_j|\right) \\
&\geq \frac{\mu_0^2}{4}\left(\sum_{j=0}^{i} |L_j| + |L_{i+1}| - 1\right) > \frac{\mu_0^2}{4}\left(\sum_{j=0}^{i+1} |U_j^L|\right)
\end{aligned}
$$

(2.2) Now we consider every U_j^L, $i+2 \leq j \leq k$, that we did not account for in case (2.1). Any set U_j^L that was not considered in (2.1) must have size less than or equal to $2|L_j|/3$, by the choice of i in (2.1). Thus the neighborhood of each U_j^L in L_j, $i < j \leq k$, has size at least $\mu_0 |U_j^L|$, and so it accounts for the node expansion of U_j^L.

It follows from cases (1) and (2) that U has at least $(\mu_0^2 |U|)/8 = \mu |U|$ neighbors outside of U in G. ∎

We group the sets R_i's and L_i's into $\mathcal{L}$ and $\mathcal{R}$, groups of $k/2$ consecutive sets, and $\mathcal{M}$, a group of $k+1$ consecutive sets (note that we have $2k+1$ distinct sets). Let $\mathcal{L} = \{L_0, L_1, \ldots, L_{k/2-1}\}$, $\mathcal{R} = \{R_0, R_1, \ldots, R_{k/2-1}\}$ and $\mathcal{M} = \{L_{k/2}, L_{k/2+1}, \ldots, L_{k-1}, L_k (= R_k), R_{k-1}, \ldots, R_{k/2}\}$. Our choice for $\mathcal{L}, \mathcal{M}$ and $\mathcal{R}$ is such that the number of sets in $\mathcal{L}$ is close to half the number of sets in $\mathcal{M}$.

4.1. It may be expensive to locally balance G. We give an initial distribution of tokens on G that has global imbalance of $\Theta((d \log n)/\mu)$. Then we will show that the multi-port algorithm will take $\Omega(\sqrt{n})$ steps to locally balance G to within $2d$ tokens. Suppose we have the following initial distribution of tokens on G: for every node z in $\mathcal{R}$, $w(z) = m+1$, where m is an integer such that $m \geq 2kd$; for all z in R_i, R_i in $\mathcal{M}$, let $w(z) = m - 2(i - k/2)d$; for all z in L_i, L_i in $\mathcal{M}$, let $w(z) = m - 2(3k/2 - i)d$; for all z in L_i, L_i in $\mathcal{L}$, let $w(z) = m - 2(i + k/2 + 1)d$. Then w is globally balanced to within $\Theta(dk) = \Theta((d \log n)/\mu)$ tokens, but it is not locally balanced to within $2d$ tokens, since $w(v) - w(u) = (k+2)d + 1 \geq 2d + 1$. See Figure 2.

We will maintain the invariant that at any step of the multi-port algorithm, every node in L_i (resp., R_i) has exactly the same number of tokens, for all $0 \leq i \leq k$. The following lemma shows that this invariant holds.

LEMMA 4.1. *Suppose every node in L_i (resp., R_i) had the same number of tokens at the start of the multi-port algorithm, for all $0 \leq i \leq k$. Then every node in L_i (resp., R_i) has exactly the same number of tokens at any step of the algorithm, for all $0 \leq i \leq k$.*

Proof. We prove this lemma using induction, and w.l.o.g., we will prove it for the sets L_i only. Suppose that every node in L_i had the same number of tokens at time step $t-1$. A node x in L_i only sends a token to one of its neighbors y in L_{i+1} if it has at least $2d+1$ more tokens than y. Thus if at time t, x sends a token to some y in L_{i+1}, then it sends a token to all of its neighbors in L_{i+1}, since all of them had the same number of tokens at time $t-1$. Note that x has at least $2d+1$ tokens, and x has at most d neighbors. Hence at time t, every edge between L_i and L_{i+1} is traversed by a token. Since every node in L_{i+1} is adjacent to exactly the same number of nodes in L_i, they all receive the same number of tokens from L_i. We can use a similar argument for tokens that move from L_i to L_{i-1}. For $i = k$, only consider tokens moving from $L_k (= R_k)$ to L_{k-1} (and R_{k-1}). No token moves between any two nodes in L_i, since all nodes in L_i had the same number of tokens at time $t-1$ (thus we can ignore the edges inside each set L_i and R_i). ∎

Now we prove the main theorem in this section for the multi-port model:

THEOREM 4.2. *The multi-port algorithm may take $\Omega(\sqrt{n})$ steps to locally balance G, even if G is globally balanced to within $\Theta((d \log n)/\mu)$ tokens initially.*

Proof. Assume we have a initial token distribution on G as defined above. The number of nodes in $\mathcal{R}$, as well as in $\mathcal{L}$, is proportional to $|R_{k/2-1}| = (1+\mu_0)^{k/2-1} = (1+\mu_0)^{\frac{\log n}{2\log(1+\mu)}-1} > \sqrt{n}/(1+\mu_0) > \sqrt{n}/2$. We claim that in order for G to be locally balanced to within $2d$ tokens, we need to move at least $\sqrt{n}/2$ tokens from $\mathcal{R}$ to $\mathcal{L}$ across edge e. Since at most one token at a time can traverse e, this will require time $\Omega(\sqrt{n})$. Our proof proceeds as follows:

(1) Since every node in R_j (resp., L_j), for $0 \leq j \leq k/2-1$, is identical with respect to both the number of tokens it has (by Lemma 4.1) and the number of neighbors it sees in R_{j-1} and R_{j+1} (resp., L_{j-1} and L_{j+1}), we observe that the tokens in G flow as follows: Tokens will be sent from v to u across edge e until u has $2d+1$ more tokens than a node in L_1. Then, every node in L_1 receives a token from u. This process continues until the nodes in L_1 have each at least $2d+1$ more tokens than a node in L_2. Then every node in L_1 will send a token to each of its neighbors in L_2 (by Lemma 4.1, every node in L_2 receives the same number of tokens from the nodes in L_1). Continuing in this fashion, the flow of tokens in $\mathcal{L}$ will proceed only from left to right, i.e., tokens never move from L_i to L_{i-1}, or inside L_i, for all L_i in $\mathcal{L}$. In parallel, as the number of tokens in v gets small, the nodes in R_1 will all send a token to v. When the nodes in R_1 have sent $2d+1$ tokens to v each, the nodes in R_2 will all send a token to each of its neighbors in R_1, etc. Thus, as in $\mathcal{L}$, the tokens also flow only from left to right in $\mathcal{R}$ (i.e., tokens never move from R_i to R_{i+1}, or inside R_i, for all R_i in $\mathcal{R}$). Thus, no token ever moves from $\mathcal{R}$ to $\mathcal{M}$, or from $\mathcal{M}$ to $\mathcal{L}$.

(2) Now we show that only after $\sqrt{n}/2$ steps have elapsed, can we have (i) $w(x)-w(y) \leq 2d$, for all x in L_i, for all y in L_{i+1}, for all L_i in $\mathcal{L}$ – i.e., $\mathcal{L}$ is locally balanced – and (ii) $w(u) > m-(k+2)d$. Suppose we reach such a configuration at some time t. Then every node in $\mathcal{L}$ has at least one more token than it had initially (since $w(u) = m-(k+2)d$ and $w(x)-w(y) = 2d$, for all x in L_i, y in L_{i+1}, and L_i in $\mathcal{L}$, initially). That is, we have at least $|\mathcal{L}| \geq \sqrt{n}/2$ "extra" tokens in $\mathcal{L}$ at time t, all of which must have reached $\mathcal{L}$ by traversing e from v to u, since no token moves from $\mathcal{M}$ to $\mathcal{L}$. Hence $t \geq \sqrt{n}/2$.

(3) We also show that only after $\sqrt{n}/2$ steps have elapsed, can we have (i) $w(y)-w(x) \leq 2d$, for all x in R_i, for all y in R_{i+1}, for all R_i in $\mathcal{R}$ – i.e., $\mathcal{R}$ is locally balanced – and (ii) $w(v) \leq m-kd$. A counting argument (similar to the one above) on the number of tokens in $\mathcal{R}$ and the fact that no token is ever sent from $\mathcal{R}$ to $\mathcal{M}$ is sufficient to show this.

From (2) and (3), we conclude that on each of the first $\sqrt{n}/2$ steps the following holds: Either $w(u) \leq m-(k+2)d$ and $w(v) > m-kd$, and so v sends a token to u, or the subnetwork induced by $\mathcal{L} \cup \mathcal{R}$ is not $2d$-locally balanced. Thus G is not locally balanced before these first $\sqrt{n}/2$ steps. Hence the algorithm will take $\Omega(\sqrt{n})$ time to locally balance G. ☐

A similar result holds for the single-port model. Assume we have the following initial distribution of tokens: for every node z in $\mathcal{R}$, $w(z) = m+1$, where m is an integer such that $m \geq k$; for all z in R_i, R_i in $\mathcal{M}$, let $w(z) = m+k/2-i$, and for all z in L_i, L_i in $\mathcal{M}$, let $w(z) = m-(3k/2-i)$; for all z in L_i, L_i in $\mathcal{L}$, let $w(z) = m-(i+k/2+1)$.

The arguments used in the proof of Theorem 4.2 can be easily modified to hold for the single-port model with the initial distribution of tokens defined above, since the lower bound on the number of steps required to reach a locally balanced state is given only in terms of how many tokens must traverse the edge e. Lemma 4.1, which is used to show that no token moves from $\mathcal{M}$ to $\mathcal{L}$ without traversing edge e, no longer holds in the single-port model. Instead, we prove Lemma 4.2, which implies that no token moves from $\mathcal{M}$ to $\mathcal{L}$ without traversing edge e, as stated in Corollary 4.2.1. Recall that in the single-port algorithm, a token moves node x to node y only if edge (x, y) is selected in the matching, and x has at least two more tokens than y.

We first prove Lemma 4.2, from which we derive Corollary 4.2.1. Let M denote the set of tokens in $\mathcal{M}$ that were either initially in $\mathcal{M}$ or that moved from $\mathcal{R}$ to $\mathcal{M}$ without using the edge e (i.e., tokens that moved from $\mathcal{R}$ to $\mathcal{M}$ through some node in $R_{k/2}$), at any step of the single-port algorithm. W.l.o.g., assume that if a node in $\mathcal{M}$ sends a token at step t of the algorithm, it will send a token that is not in M if it has one, for all t.

LEMMA 4.2. *At any step of the single-port algorithm, for any node x in $\mathcal{M}$, the number of tokens on x that belong to M is at most the total number of tokens on x initially.*

Proof. By definition, a token in M is either a token that was in $\mathcal{M}$ initially, or a token that moved from $\mathcal{R}$ to $\mathcal{M}$ through some node in $R_{k/2}$. Suppose, for the sake of contradiction, that at step t, a node x in $\mathcal{M}$ has one more token in M than it had initially. Assume x had b tokens initially. There must exist a sequence of nodes $x = x_1, \ldots, x_p$ such that (i) x_i is adjacent to x_{i+1} in G, (ii) x_i had at least $b+i$ tokens at time t_i, (iii) $t_p < \ldots < t_1 = t$, and (iv) x_p is in $R_{k/2}$ and x_i, $i \neq p$, is not in $R_{k/2}$. There are two cases to consider:

(1) If no x_i is in $\mathcal{L}$ (i.e., every node x_i is in $\mathcal{M}$), then let q be the distance in $\mathcal{M}$ from x to x_p. Thus x_p has at least $b+p \geq b+q+1 = m+1$ tokens at time t_p. But no node in $R_{k/2}$ can have $m+1$ tokens, since $m+1$ is the maximum number of tokens in G initially, and no node in $R_{k/2}$ had that many tokens initially.

(2) Else, let x_j (resp., $x_{j'}$) be the first (resp., last) node in the sequence that is not in $\mathcal{M}$. Then x_{j-1} and $x_{j'+1}$ must be in $L_{k/2}$. Let q be the distance from v to x_{j-1} in $\mathcal{M}$. Then $x_{j'+1}$ has at least $b+p \geq b+q+2 \geq (m-k)+q+1$ tokens at step $t_{j'+1}$. Thus x_p must have at least $b+q+k+2 \geq m+q+1 \geq m+1$ tokens at step t_p: a contradiction (see item (1)). □

COROLLARY 4.2.1. *No token initially in $\mathcal{M} \cup \mathcal{R}$ ever moves from $\mathcal{M}$ to $\mathcal{L}$ without traversing edge e.*

Proof. By Lemma 4.2, no node x in $L_{k/2}$ will ever have more tokens in M than it had initially. Since the number of tokens on x at the beginning of the algorithm, $m-k$, is minimum (on the entire network), it follows that x will have exactly $m-k$ tokens that belong to M at any step of the algorithm. Thus x will never send a token that belongs to M to any other node in G. Since a token can move from $\mathcal{M}$ to $\mathcal{L}$ only through some node in $L_{k/2}$, the corollary follows. □

Any edge is selected independently with probability $O(1/d)$ at each iteration of the single-port algorithm. Thus an edge e is selected, on average, an $O(1/d)$ fraction of the time. Hence, we can show that it will take $\Omega(d\sqrt{n})$ expected time for G to be locally balanced to within 1 token in the single-port model, even if initially G is globally balanced to within $O((\log n)/\mu)$ tokens, as stated in the theorem below.

THEOREM 4.3. *The single-port algorithm may take $\Omega(d\sqrt{n})$ expected number of steps to locally balance G to within one token, even if G is globally balanced to within $O((\log n)/\mu)$ tokens initially.* □

4.2. The single-port algorithm may diverge from an almost locally balanced state. In this section we consider the single-port model only. Suppose we have the following distribution of tokens on G: for all z in $R_{k/2-1}$, let $w(z) = m+1$ (where m is an integer such that $m \geq k$), and for all z in R_i, $i \leq k/2-2$ (note that $R_i \in \mathcal{R}$), let $w(z) = m-(k/2-i-1)$; for all z in $R_{k/2}$, let $w(z) = m-1$, for all z in R_i, $i \geq k/2+1$ (note that $R_i \in \mathcal{M}$), let $w(z) = m-(i-k/2)$, for all z in L_i, L_i in $\mathcal{M}$, let $w(z) = m-(3k/2-i)$; for all z in L_i, L_i in $\mathcal{L}$, let $w(z) = m-(k/2+i)$. Thus w is globally balanced to within $O((\log n)/\mu)$ tokens but it is not locally balanced to within 1 token, since $w(x)-w(y) = 2$, for any x in $R_{k/2-1}$ and y in $R_{k/2} \cup R_{k/2-2}$. See Figure 3.

The intuition for this case is that if all tokens move in the "right direction" initially, we reach a locally balanced state in a single time step. Otherwise, if a large number of tokens move in the "wrong direction" in the first step, it will take many steps until we reach such a state. If every node in $R_{k/2-1}$ is matched with some node in $R_{k/2}$ (we can construct G such that every node in $R_{k/2-1}$ has a distinct neighbor in $R_{k/2}$), G reaches local balance in a single time step. On the other hand, if some tokens move across a matching between the nodes in $R_{k/2-1}$ and $R_{k/2-2}$, then these tokens will continue moving "down" (non-decreasing indices of R_i), and never move "up". The expected size of such a matching will be $\Omega(|R_{k/2-1}|/d)$ (each node in $R_{k/2-1}$ has exactly $d/(2(2+\mu_0)) \geq d/5$ neighbors in $R_{k/2-2}$). Using an analysis similar to that of Section 4.1 for the single-port model, we see that no token that was initially in $\mathcal{M} \cup \mathcal{R}$ moves from $\mathcal{M}$ to $\mathcal{L}$ without traversing edge e, and that either $w(v) > m-k/2+1$ and $w(u) = m-k/2$, or $\mathcal{L} \cup \mathcal{R}$ is not locally balanced, for each of the $\Omega(|R_{k/2-1}|/d)$ initial steps. Thus, once any of the tokens that moved from $R_{k/2-1}$ to $R_{k/2-2}$ in the first step reaches v, it will eventually traverse e onto u.

Hence, eventually all tokens that moved from $R_{k/2-1}$ to $R_{k/2-2}$ in the initial step will reach u. Since $|R_{k/2-1}| \geq \mu\sqrt{n}/((1+\mu_0)^2)$ and $1 < (1+\mu_0)^2 < 2$, the expected number of edges between nodes in $R_{k/2-1}$ and nodes in $R_{k/2-2}$ in a selected matching is $\Omega(|R_{k/2-1}|/d) = \Omega(\mu\sqrt{n}/d)$. The expected time for edge e to be in a selected matching is at least d. Thus the expected time for G to be locally balanced to within one token is $\Omega(\mu\sqrt{n})$. This result is stated in the following theorem.

THEOREM 4.4. *There exists an initial distribution of tokens from which the network G can be locally balanced to within one token in one step, but for which the expected number of steps required by the single-port algorithm to locally balance G to within one token is $\Omega(\mu\sqrt{n})$.* □

4.3. Convergence to a locally balanced state. We now prove that if the graph G is globally balanced to within Δ tokens, in $O(n\Delta^2/d)$ subsequent steps the multi-port algorithm locally balances G to within $2d$ tokens. Define the potential Φ of the network as $\sum_{v \in V}(w(v)-\rho)^2$. If the network is globally balanced to within Δ tokens, then $\Phi = O(n\Delta^2)$. At any step, if there exists an edge (u,v) such that $|w(u)-w(v)| \geq 2d+1$, then a token is transmitted along (u,v) resulting in a potential drop of at least d. Thus, within $O(n\Delta^2/d)$ steps the network is globally balanced. Similarly, it is not difficult to show that if the graph G is globally balanced to within Δ tokens, then the single-port algorithm locally balances to within one token in $O(nd\Delta^2)$ subsequent steps with high probability, since a token transmitted along an edge results in a potential drop of at least one, and an edge is selected with probability at least $1/d$.

5. Extension to dynamic and asynchronous networks. In this section, we extend our results of Section 3.2 for the multi-port model to dynamic and asynchronous networks. We first prove that a variant of the local multi-port algorithm is optimal on dynamic synchronous networks in the same sense as for static synchronous networks. We then use a result of [2] that relates the dynamic synchronous and asynchronous models to extend our results to asynchronous networks.

In the dynamic synchronous model, the edges of the network may fail or succeed dynamically. An edge $e \in E$ is *live* during step t if e can transmit a message in each direction during step t. We assume that at each step each node knows which of its adjacent edges are live. The local load balancing algorithm for static synchronous networks can be modified to work on dynamic synchronous networks. The algorithm presented here is essentially the same as in [2].

Since edges may fail dynamically, a node u may have no knowledge of the height of a neighboring node v and hence may be unable to decide whether to send a token to v. In our algorithm, which we call **DS**, every node u maintains an estimate $e^u(v)$ of the number of tokens at v for every neighbor v of u. (The value of $e^u(v)$ at the start of the algorithm is arbitrary.) In every step of the algorithm, each node u performs the following operations:

(1) For each live neighbor v of u, if $w(u) - e^u(v) > 12d$, u sends a message consisting of $w(u)$ and a token; otherwise, u sends a message consisting only of $w(u)$. Next, $w(u)$ is decreased by the number of tokens sent.

(2) For each message received from a live neighbor v, $e^u(v)$ is updated according to the message and if the message contains a token, $w(u)$ is increased by one.

Unlike the algorithm for static networks, the above algorithm may (temporarily) worsen the imbalance since a node may have an old estimate of the height of one of its neighbors. Two anamolies may occur while executing **DS**: (i) a token sent by u to v may gain height as it is possible for $w(u) - e^u(v)$ to be greater than $12d$ even if $w(u)$ is at most $w(v)$, and (ii) node u may not send a token to v as it is possible for $w(u) - e^u(v)$ to be at most $12d$ even if $w(u) - w(v)$ is much larger than $12d$. Consequently, the analysis for dynamic networks is more difficult than for static networks. We employ a more complicated amortized analysis to account for the above anomalies.

For every integer i, let S_i denote the set of nodes that have at least $\rho - 12d + 24id$ and at most $\rho + 12d - 1 + 24id$ tokens. Consider T steps of **DS**. We assume without loss of generality that $|S_{>0}| \le n/2$ at the start of at least $T/2$ steps. As shown in Section 2, there exists an index j in $[1, 2\lceil \log_{(1+\alpha/(2d))} n \rceil]$ that is good in at least half of those steps in which $|S_{>0}| \le n/2$. (Recall that index i is good if $|S_i| \le \alpha |S_{>i}|/2d$.) If index j is good at the start of step t, we call t a *good* step. For any token p, let $h_t(p)$ denote the height of p after step t, $t > 0$. For convenience, we denote the height of p at the start of **DS** by $h_0(p)$. Similarly, for $t \ge 0$, we define $h_t(u)$ for every node u and $e^u_t(v)$ for every edge (u, v).

With every token at height h, we associate a potential of $\phi(h)$, where $\phi : N \to R$ is defined as follows:

$$\phi(x) = \begin{cases} 0 & \text{if } x \le 24jd - 11d, \\ (1+\nu)^x & \text{otherwise,} \end{cases}$$

where $\nu = \alpha/(cd^2)$ and $c > 0$ is a constant to be specified later. Let Φ_t denote the total potential of the network after step t. Let Ψ_t denote the potential drop during step t.

We analyze **DS** by means of an amortized analysis over the steps of the algorithm. Let E_t be the set $\{(u, v) : (u, v)$ is live during step t, $u \in S_{>j}$ and $h_{t-1}(u) - h_{t-1}(v) \ge 24d\}$. For every step t, we assign an amortized potential drop of

$$\hat{\Psi}_t = \frac{1}{50} \sum_{\substack{(u,v) \in E_t \\ h_{t-1}(u) > h_{t-1}(v)}} (\phi(h_{t-1}(u) - d) - \phi(h_{t-1}(v) + d)).$$

The definition of $\hat{\Psi}_t$ is analogous to the amount of potential drop that we use in step t in the argument of Section 3.2 for the static case. By modifying that argument slightly and choosing appropriate values for the constants c and ε, we show the following lemma.

LEMMA 5.1. *If the live edges of G have an edge expansion of α during every step of* **DS**, *then for every good step t we have $\hat{\Psi}_t \ge \varepsilon \nu^2 d^2 \Phi_{t-1}$, where ε is an appropriately chosen constant.*

Proof Sketch. Let M_i denote the set of live edges between nodes in $S_{<i}$ and nodes in $S_{>i}$. Let $m_i = |M_i|$. For any node u, let $g(u)$ represent the group to which u belongs prior to step t. We now place a lower bound on $\hat{\Psi}_t$

which is analogous to that on Ψ in Lemma 3.4 of Section 3.2. By the definition of $\hat{\Psi}_t$, we have

$$\begin{aligned}
\hat{\Psi}_t &= \frac{1}{50} \sum_{\substack{(u,v)\in E_t \\ h_{t-1}(u)>h_{t-1}(v)}} (\phi(h_{t-1}(u)-d) - \phi(h_{t-1}(v)+d)) \\
&\geq \frac{1}{50} \sum_{\substack{(u,v)\in E_t \\ h_{t-1}(u)>h_{t-1}(v)}} \sum_{g(v)<i<g(u)} (\phi(24(i+1)d-13d) - \phi(24(i-1)d+13d)) \\
&= \frac{1}{50} \sum_{i\geq j} \sum_{\substack{(u,v)\in M_i \\ h_{t-1}(u)>h_{t-1}(v)}} (\phi(24(i+1)d-13d) - \phi(24(i-1)d+13d)) \\
&= \frac{1}{50} \sum_{i> j} \sum_{\substack{(u,v)\in M_i \\ h_{t-1}(u)>h_{t-1}(v)}} (\phi(24(i+1)d-13d) - \phi(24(i-1)d+13d)) \\
&\quad + \frac{1}{50} \sum_{\substack{(u,v)\in M_j \\ h_{t-1}(u)>h_{t-1}(v)}} \phi(24(j+1)d-13d) \\
&\geq \frac{22}{50} \sum_{i> j} \sum_{\substack{(u,v)\in M_i \\ h_{t-1}(u)>h_{t-1}(v)}} \nu d(1+\nu)^{24id-11d} + \frac{1}{50} \sum_{\substack{(u,v)\in M_j \\ h_{t-1}(u)>h_{t-1}(v)}} (1+\nu)^{24jd+11d} \\
&\geq \frac{22}{50} \sum_{i>j} m_i \nu d(1+\nu)^{24id-11d} + \frac{1}{50} m_j (1+\nu)^{24jd+11d}.
\end{aligned}$$

(For the second equation, note that $24id - 13d \leq 24(i-1)d + 13d$. Therefore, $\phi(24id-13d) \leq \phi(24(i-1)d+13d)$. The second equation now follows since the sum telescopes. The third equation is obtained by interchanging the sums and noting that $\phi(x)$ is zero for $x \leq 24jd - 11d$. The fourth equation is obtained by partitioning the set M into subsets $M \setminus M_j$ and M_j. The fifth equation uses the following calculations: (i) $\phi(24id+11d) - \phi(24id-11d) \geq ((1+\nu)^{22d}-1)(1+\nu)^{24id-11d} \geq 22d(1+\nu)^{24id-11d}$, (ii) $\phi(24jd+11d) = (1+\nu)^{24jd+11d}$, and (iii) $\phi(24jd-11d) = 0$. The last equation follows from the definition of m_i.)

We next establish claims similar to Lemma 3.5 and Corollary 3.5.1 of Section 3.2 by modifying the constants in the proofs. Thus we have $\hat{\Psi}_t \geq \varepsilon\nu^2 d^2 \Phi_{t-1}$ for an appropriately chosen constant ε. □

The following lemma relates the amortized potential drops to the actual potential drops.

LEMMA 5.2. *For any initial load distribution and any step $t' > 0$, we have*

$$\sum_{t\leq t'} \Psi_t \geq \left(\sum_{t\leq t'} \hat{\Psi}_t\right) - 2\Phi_0 - n^2\phi(24jd). \tag{4}$$

In order to prove Lemma 5.2, we need to address two issues that arise in the dynamic setting: (i) potential gains, i.e., when a token gains height, and (ii) the lack of a potential drop across edges that join nodes differing by at least $24d$ tokens. We show that for any of the above events to occur, "many" tokens should have lost height in previous steps. We use a part of this prior potential drop to account for (i) and (ii). At a high level, our proof follows the lines of Lemma 3 of [2]. However, since the potential functions involved are different, the two proofs differ considerably in the details. We have included a complete proof of Lemma 5.2 in Appendix B.

The main result follows from Lemmas 5.1 and 5.2. We first show that within $O(1/(\varepsilon\nu^2 d^2))$ steps, there is a step when the actual potential of the network either decreases by a factor of 2 or falls below a threshold value.

LEMMA 5.3. *Let t be any integer such that at least $7/(\varepsilon\nu^2 d^2)$ of the first t steps are good. There exists $t' \leq t$ such that $\Phi_{t'} \leq \max\{\Phi_0/2, n^2\phi(24jd)\}$.*

Proof. If $\Phi_0 \leq n^2\phi(24jd)$, then the claim is proved for $t = 0$. For the remainder of the proof, we assume that $\Phi_0 \geq n^2\phi(24jd)$. If $\Phi_{t'} \leq \Phi_0/2$ for any $t' < t$, the claim is proved. Otherwise, for all $t' < t$, we have $\Phi_{t'} > \Phi_0/2$. In this case, we obtain

$$\Phi_t = \Phi_0 - \sum_{t'<t} \Psi_{t'}$$

$$
\begin{aligned}
&\leq 3\Phi_0 + n^2\phi(24jd) - \sum_{t'<t} \hat{\Psi}_{t'} \\
&\leq 4\Phi_0 - \sum_{\substack{t'<t \\ t'\text{good}}} (\varepsilon\nu^2 d^2)\Phi_{t'} \\
&\leq \Phi_0/2.
\end{aligned}
$$

(To obtain the second equation, we invoke Lemma 5.2. For the third equation, we invoke Lemma 5.1 and use the inequalities $\Phi_0 \geq n^2\phi(24jd)$, and $\hat{\Psi}_{t'} \geq 0$ for every t'. For the last equation, we use the fact that at least $7/(\varepsilon\nu^2 d^2)$ of the t steps are good, and the equation $\Phi_{t'} > \Phi_0/2$ for every $t' < t$.) □

THEOREM 5.1. *For an arbitrary network G with n nodes, degree d, and initial imbalance Δ, if the live edges at every step t of G have edge expansion α, then the dynamic synchronous multi-port algorithm load balances to within $O(d^2(\log n)/\alpha)$ tokens in $O(\Delta/\alpha)$ steps.*

Proof. We first place an upper bound on the number t of steps such that the height of each node at the end of step t is $O(d^2(\log n)/\alpha^2)$. If Δ is at most $d^2(\log n)/\alpha^2$, then a trivial bound is 0.

We now consider the case when Δ is at least $d^2(\log n)/\alpha^2$. By repeatedly invoking Lemma 5.3, we obtain that within $T = \lceil(7/(\varepsilon\nu^2 d^2))\rceil\lceil\log\Phi_0\rceil$ good steps, there exists a step after which the potential of the network is at most $n^2\phi(24jd)$. (Note that the fact that Lemma 5.2 holds for arbitrary initial values of the estimates, the $e^u(v)$'s, is crucial here.) Since at least $T/4$ of the first T steps are good, there exists $t \leq 4\lceil(7/(\varepsilon\nu^2 d^2))\rceil\lceil\log\Phi_0\rceil$ such that $\Phi_t \leq n^2\phi(24jd)$. Since $\Phi_0 \leq n(1+\nu)^{(\Delta+1)}/\nu$, we have $\log\Phi_0 \leq \log n + (\Delta+1)\log(1+\nu) - \log\nu$. Since $\nu = \alpha/(cd^2)$ and $\log(1+\nu) < \nu$, we have $t = O((\Delta/\alpha) + d^2(\log n)/\alpha^2) = O(\Delta/\alpha)$.

Let h be the maximum height of any node after step t. We thus have

$$
\begin{aligned}
\phi(h) &\leq \Phi_t \\
&\leq n^2(1+\nu)^{24jd}.
\end{aligned}
$$

Therefore, if $\phi(h) > 0$, then $h \leq \log_{(1+\nu)}(n^2(1+\nu)^{24jd})$. If $\phi(h) = 0$, then $h \leq 24jd - 11d$. In either case,

$$
\begin{aligned}
h &\leq 24jd + (2\log n)/\log(1+\nu) \\
&\leq 24jd + (4\log n)/\nu \\
&= O((d^2\log n)/\alpha)
\end{aligned}
$$

(The right hand side of the first equation is an expansion of $\log_{(1+\nu)}(n^2(1+\nu)^{24jd})$. The second equation holds since $\log(1+\nu) < \nu/2$ for c appropriately large. The final inequaility follows from the relations $\nu = \alpha/(cd^2)$ and $j = O((d\log n)/\alpha)$.)

Thus, at the end of step t, no node has more than $a = \rho + h$ tokens. We now prove by contradiction that for every step after step t, no node has more than $a + d$ tokens. Let t' be the first step after step t such that there exists some node u with more than $a + d$ tokens. (If no such t' exists, the claim holds trivially.) Of the $d+1$ highest tokens received by u after step t, at least 2 tokens (say p and q) were last sent by the same neighbor v of u. Without loss of generality, we assume that p arrived at u before q. Let t_1 be the step when p was last sent by v to u. Therefore, we have $e^v_{t_1}(u) \geq h_{t_1}(p) - d \geq a - d$. Hence q can be sent to u only when v has height at least $a + 11d$, which contradicts our choice of t'.

We have shown that after $O(\Delta/\alpha)$ steps, no node ever has more than $\rho + O((d^2\log n)/\alpha)$ tokens. An easy averaging argument shows that there exists $k = O((d\log n)/\alpha)$ such that after every step $t' \geq t$, $|S_{<-k}| \leq n/2$. By defining an appropriate potential function for tokens with heights below the average and repeating the analysis done for $S_{>j}$, we show that in another $O(\Delta/\alpha)$ steps, all nodes have more than $\rho - O(d^2(\log n)/\alpha)$ tokens. □

As suggested in [2], a simple variant of **DS** can be defined for asynchronous networks. As shown in [2], the analysis for the dynamic synchronous case can be used for asynchronous networks to yield the same time bounds. Hence, the multi-port local load balancing algorithm balances to within $O(d^2\log n/\alpha)$ tokens in time $O(\Delta/\alpha)$ on aynchronous networks.

6. Tight bounds on off-line load balancing. In this section, we analyze the load balancing problem in the off-line setting for both single-port and multi-port models. We derive nearly tight bounds on the minimum number of steps required to balance on arbitrary networks in terms of the node and edge expansion of the networks. We assume that the network is synchronous.

We first consider the network $G = (V, E)$ under the single-port model. For any subset X of V, let $\overline{X}$ denote $V \setminus X$, $m(X)$ denote the number of edges in a maximum matching between X and $\overline{X}$, $A(X)$ denote the set $\{v \in \overline{X} : \exists x \in X \text{ such that } (x, v) \in E\}$, and $B(X)$ denote the set $\{x \in X : \exists y \in A(X) \text{ such that } (x, y) \in E\}$. For subsets X and Y of V, let $M(X, Y)$ denote the set of edges with one endpoint in X and the other in Y.

LEMMA 6.1. *For any network $G = (V, E)$ with node expansion μ and any subset X of V, we have $m(X) \le \mu \min\{|X|, |\overline{X}|\}/(1 + \mu)$. Moreover, for any subset X of V, $m(X \cup A(X)) \le |A(X)|$.*

Proof. Without loss of generality, assume that $|X| \le |\overline{X}|$. Consider the bipartite graph $H = (B(X), A(X), M(X, \overline{X}))$. A maximum matching in H is equal to a maximum flow in the graph $I = (B(X) \cup A(X) \cup \{s, t\}, M(X, \overline{X}) \cup \{(s, x) : x \in B(X)\} \cup \{(x, t) : x \in A(X)\})$ from source s to sink t. (All of the edges of I have unit capacity.) We will show that every cut C of I separating s and t is of cardinality at least $\mu|X|/(1 + \mu)$.

Consider any cut $C = (S, T)$ with $s \in S$ and $t \in T$. The set of edges in C is $M(S, T)$. Let $Y = T \cap B(X)$ and $Z = T \cap A(X)$. The capacity of C, given by $|M(S, T)|$, can be lower bounded as follows.

$$\begin{aligned}
|M(S,T)| &= |Y| + |M(Y, A(X) \setminus Z)| + |M(B(X) \setminus Y, Z)| + |A(X) \setminus Z| \\
&\ge |Y| + |M(B(X) \setminus Y, Z)| + |A(X) \setminus Z| \\
&\ge |A(X \setminus Y)| \\
&\ge \mu|X \setminus Y| \\
&= \mu(|X| - |Y|) \\
&\ge \mu|X|/(1 + \mu).
\end{aligned}$$

(For the third equation, see Figure 1. Three subsets of nodes contribute to the set $A(X \setminus Y)$: (i) the set of nodes in Y that have an edge to a node in $X \setminus Y$, (ii) the set of nodes in Z that have an edge to a node in $X \setminus Y$, and (iii) the set of nodes in $A(X) \setminus Z$ that have an edge to a node in $X \setminus Y$. The size of the three sets is bounded by $|Y|$, $|M(B(X) \setminus Y, Z)|$, and $|A(X) \setminus Z|$, respectively. The fourth equation follows from the definition of $A(X \setminus Y)$. The fifth equation holds since Y is a subset of X. The last equation holds since $|Y| \le |M(S, T)|$.) For the second part of

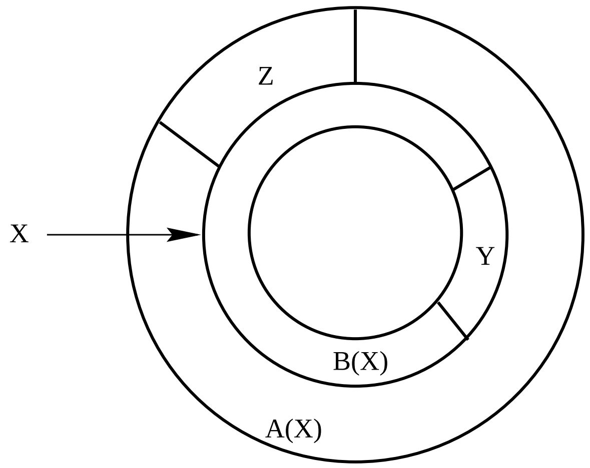

FIG. 1. *The sets X, Y, Z, $A(X)$, and $B(X)$ in the proof of Lemma 6.1.*

the lemma, we note that since all of the neighbors of X are in $A(X)$, any node in $X \cup A(X)$ that connects to some node outside of $X \cup A(X)$ is in $A(X)$. Therefore, $m(X \cup A(X)) \le |A(X)|$. □

Theorem 1 of [29] obtains tight bounds on the off-line complexity of load balancing in terms of the function m. We restate the theorem using our notation and terminology. Before stating the theorem, we need one additional notation. For any subset X of nodes of any network, let $I(X)$ denote the number of tokens held by nodes in X in the initial distribution.

THEOREM 6.1 ([29]). *Consider a network $G = (V, E)$ in the single-port model. The network G can be balanced in at most $\max_{\emptyset \subsetneq X \subsetneq V} \lceil (I(X) - \rho|X|)/m(X) \rceil$ steps so that every node has at most $\lceil \rho \rceil + 1$ tokens. Moreover, any algorithm takes at least $\max_{\emptyset \subsetneq X \subsetneq V} \lceil (I(X) - \rho|X|)/m(X) \rceil$ steps to balance the network so that every node has at most $\lceil \rho \rceil$ tokens.* □

Theorem 6.1 and Lemma 6.1 imply the following result.

LEMMA 6.2. *Assume the single-port model. Any network G with node expansion μ and initial imbalance Δ can be balanced in at most $\lceil \Delta(1+\mu)/\mu \rceil$ steps so that every node has at most $\lceil \rho \rceil + 1$ tokens. Moreover, there exists a network G and an initial load distribution with imbalance Δ such that any algorithm takes at least $\lceil \Delta(1+\mu)/\mu \rceil$ steps to balance G such that every node has at most $\lceil \rho \rceil$ tokens.*

Proof. If $I(X)$ is the the total number of tokens belonging to nodes in X in the initial distribution distribution, then we have: $-\Delta|X| \le I(X) - \rho|X| \le \Delta|X|$ for all X. Moreover, $|I(X) - \rho|X|| = |I(\overline{X}) - \rho|\overline{X}||$. Therefore, for all X, $|I(X) - \rho|X|| = \Delta \min\{|X|, |\overline{X}|\}$. By Lemma 6.1, $m(X)$ is at least $\mu \min\{|X|, |\overline{X}|\}/(1+\mu)$ for all X. Thus, the first claim of Theorem 6.1 establishes the first claim of the desired lemma.

For the second claim of the lemma, given any μ, we construct the following network $G = (V, E)$ with node expansion μ. The node set V is partitioned into 3 sets X, Y, and Z such that: (i) $|Y| = \mu|X|$, and (ii) $|Z| = |X|(1+\mu)^2/(1-\mu)$. Let n and x denote $|V|$ and $|X|$, respectively. Thus, n equals $x(1+\mu+(1+\mu)^2/(1-\mu)) = 2x(1+\mu)/(1-\mu)$. The edge set E is the union of the sets $X \times X$, $X \times Y$, $Y \times Y$, $Y \times Z$, and $Z \times Z$.

We now show that the node expansion of G is μ. Consider any nonempty subset U of V of size at most $n/2$ and let X', Y', and Z' denote $U \cap X$, $U \cap Y$, and $U \cap Z$, respectively. Let $n(U)$ denote the number of neighbors of U that lie outside of U. We need to show that $n(U)$ is at least $\mu|U|$.

We consider two cases: (i) Y' and Z' are both empty, and (ii) Y' is nonempty or Z' is nonempty. In the first case, $U = X'$. Therefore, $n(U) \ge |Y| = \mu x \ge \mu|U|$. In the second case, we have:

$$\begin{aligned} n(U) &\ge |Z| - |Z'| \\ &\ge |Z| - |U| \\ &\ge x((1+\mu)^2/(1-\mu) - (1+\mu)/(1-\mu)) \\ &= x\mu(1+\mu)/(1-\mu) \\ &\ge \mu|U|. \end{aligned}$$

(The second equation holds since Z' is a subset of U. For the third equation, note that $|U| \le n/2 = x(1+\mu)/(1-\mu)$. The last equation follows from the upper bound of $x(1+\mu)/(1-\mu)$ on $|U|$.)

We now apply the second claim of Lemma 6.1 to the subset X. Since $A(X) = Y$, $m(X \cup Y) = \mu x = \mu|X \cup Y|/(1+\mu)$. Given any Δ, consider the initial token distribution in which each node in $X \cup Y$ has $\rho + \Delta$ tokens, and each node in Z has $\rho - \Delta(1-\mu)/(1+\mu)$ tokens, where ρ is any integer that is at least $\Delta(1-\mu)/(1+\mu)$. (Note that the average number of tokens is ρ.) By applying the second claim of Theorem 6.1, we obtain that the number of steps to balance G so that each node has at most ρ tokens is at least $(I(X \cup Y) - \rho|X \cup Y|)/m(X \cup Y) \ge \Delta|X \cup Y|/m(X \cup Y) \ge \Delta(1+\mu)/\mu$. Since the number of steps is an integer, the desired claim follows. □

By using the techniques of [29], we can modify the proof of Lemma 6.2 to show that any network G with node expansion μ and initial imbalance Δ can be globally balanced to within 3 tokens in at most $2\lceil \Delta(1+\mu)/\mu \rceil$ steps. The extra factor of 2 is required because even after balancing the network so that each node has at most $\lceil \rho \rceil + 1$ tokens, there may exist a node with considerably fewer than ρ tokens. It takes an additional $\lceil \Delta(1+\mu)/\mu \rceil$ steps to bring the network to a state in which the global imbalance is at most 3.

Lemma 6.2 implies that the time bound achieved by the single-port algorithm (see Theorems 3.1 and 3.3) is not optimal for all networks. An example of a network for which the single-port algorithm is not optimal is the hypercube whose maximum degree is $\log n$, edge expansion is 1, and node expansion is $\Theta(1/\sqrt{\log n})$. The local algorithm balances in $\Omega(\Delta \log n)$ time, while there exists an $O(\Delta\sqrt{\log n} + \log^2 n)$ time load balancing algorithm for the hypercube [34] which is optimal for Δ sufficiently large. For the class of constant-degree networks, however, the time taken by the single-port algorithm to reduce the global imbalance to $O(\log n/\mu)$ (see Theorem 3.3) is within a constant factor of the time taken by any algorithm to completely balance the network (see Lemma 6.2).

The proofs of Theorem 1 of [29] and Lemma 6.2 can be modified to establish the following result for the multi-port model.

LEMMA 6.3. *Assume the multi-port model. Any network G with edge expansion α and initial imbalance Δ can be balanced in at most $\lceil \Delta/\alpha \rceil$ steps so that every node has at most $\lceil \rho \rceil + d$ tokens. Moreover, for every network G, there exists an initial load distribution with imbalance Δ such that any algorithm takes at least $\lceil \Delta/\alpha \rceil$ steps to balance G so that every node has at most $\lceil \rho \rceil$ tokens.*

Proof Sketch. We prove that there exists an off-line algorithm that balances to within d tokens in at most $T = \max_{\emptyset \subset X \subset V} \left\lceil \frac{|I(X) - \rho|X||}{|M(X,\overline{X})|} \right\rceil$ steps. For all $X \subseteq V$, we have (i) $|I(X) - \rho|X|| \le \Delta \min\{|X|, |\overline{X}|\}$ (see proof of Lemma 6.2), and (ii) $|M(X, \overline{X})| \ge \alpha \min\{|X|, |\overline{X}|\}$. It follows from (i) and (ii) that $T \le \lceil \Delta/\alpha \rceil$.

We modify the proofs of Theorem 1 and Lemma 4 of [29] (where the single-port model was assumed) to establish the desired claims for the multi-port model. We transform the load balancing problem on G to a network flow problem on a directed graph $H = (V', E')$ which is constructed as follows. Let V_i be $\{\langle v, i\rangle : v \in V\}$, $0 \le i \le T$. Let E_i be $\{(\langle u, i\rangle, \langle v, i+1\rangle) : (u, v) \in E \text{ or } u = v\}$, $0 \le i < T$. We set V' to $\{s\} \cup \bigcup_{0 \le i \le T} V_i \cup \{t\}$, and E' to $\{(s, \langle v, 0\rangle) : v \in V\} \cup \bigcup_{0 \le i < T} E_i \cup \{(\langle v, T\rangle, t) : v \in V\}$. For any v in V, the capacity of the edge $(s, \langle v, 0\rangle)$ is $w(v)$. For any (u, v) in E, the capacity of any edge $(\langle u, i\rangle, \langle v, i+1\rangle)$, $0 \le i < T$, is 1. For any v in V, the capacity of any edge $(\langle v, i\rangle, \langle v, i+1\rangle)$, $0 \le i < T$, is ∞. For any v in V, the capacity of the edge $(\langle v, T\rangle, t)$ is $\lceil \rho \rceil + d$.

We show that the value of the maximum integral flow in H is equal to the total number of tokens N in V, from which it follows that there exists an off-line algorithm that balances to within d tokens in T steps. Consider any cut $C = (S, T)$ of H separating $s \in S$ and $t \in T$. Let $S_i = S \cap V_i$ and $D(S_i) = \{v \in V : \langle v, i\rangle \in S_i\}$. If $S_0 = \emptyset$, or $S_T = V_T$, or there is an edge of infinite capacity, then the capacity of C is at least N. Otherwise, the number of edges from V_i to V_{i+1} that belong to the cut is at least $|M(D(S_i), \overline{D(S_i)})| - d(|S_{i+1}| - |S_i|)$. Moreover, since there is no edge with infinite capacity in C, $D(S_i)$ is a subset of $D(S_{i+1})$. Thus the capacity of C is at least

$$
\begin{aligned}
& I(D(V_0) \setminus D(S_0)) + \left(\sum_{i=0}^{T-1} \left(|M(D(S_i), \overline{D(S_i)})| - d(|S_{i+1}| - |S_i|) \right) \right) + (\lceil \rho \rceil + d)|S_T| \\
& \ge I(D(V_0) \setminus D(S_0)) + \left(\sum_{i=0}^{T-1} ((I(D(S_i)) - \rho|S_i|)/T - d(|S_{i+1}| - |S_i|)) \right) + (\lceil \rho \rceil + d)|S_T| \\
& \ge I(D(V_0) \setminus D(S_0)) + \left(\sum_{i=0}^{T-1} ((I(D(S_0)) - \rho|S_T|)/T - d(|S_T| - |S_0|)) \right) + (\lceil \rho \rceil + d)|S_T| \\
& \ge I(D(V_0) \setminus D(S_0)) + I(D(S_0)) - \rho|S_T| + d|S_0| + \lceil \rho \rceil |S_T| \\
& \ge N.
\end{aligned}
$$

(In the first equation: (i) $I(D(V_0) \setminus D(S_0))$ is the capacity of the edges from s to V_0 that belong to the cut, (ii) $|M(D(S_i), \overline{D(S_i)})| - d(|S_{i+1}| - |S_i|)$ is the capacity of the edges from V_i to V_{i+1} that belong to the cut, and (iii) $(\lceil \rho \rceil + d)|S_T|$ is the capacity of the edges from S_T to t that belong to the cut. The second equation follows from the definition of T and the fact that $|D(S_i)| = |S_i|$. For the third equation, note that $D(S_0)$ is a subset of $D(S_i)$ for all i and $|S_T| \ge |S_i|$ for all i. The fourth equation holds since the sum of $|S_{i+1}| - |S_i|$ telescopes. The final equation is obtained since $I(D(V_0)) = N$.) Since the capacity of the cut $(\{s\}, V' \setminus \{s\})$ equals N, the maximum flow in H is N.

To prove the second part of the lemma, given any network G with a partition (V_1, V_2) of its nodes such that $|V_1| \le n/2$ and $|M(V_1, V_2)| = \alpha |V_1|$, we define an initial load distribution with average ρ in which each node in V_1 has $\rho + \Delta$ tokens and each node in V_2 has $\rho - \Delta|V_1|/|V_2|$ tokens. The desired claim holds since at least $\Delta|V_1|$ tokens need to leave the set V_1. □

Lemma 6.3 implies that the local multi-port algorithm is asymptotically optimal for *all* networks. As in the single-port case, we can modify the above proof to obtain upper bounds on the off-line complexity of globally balancing a network. We can show that any network G with edge expansion α and initial imbalance Δ can be globally balanced to within $d + 1$ tokens in at most $2\lceil \Delta/\alpha \rceil$ steps.

Appendix A. Some Technical Inequalities. Let ν equal $\alpha/(cd^2)$. For the following we set c large enough so that $(1+\nu)^{12d} \le 3/2$. The function ϕ is defined in Section 5.

LEMMA A.1. *For any integer x, if $\phi(x) > 0$, then $\phi(x + 12d) \le 3\phi(x)/2$.*

Proof. Since $\phi(x) > 0$, we have $\phi(x + 12d) = (1+\nu)^{12d}\phi(x) \le 3\phi(x)/2$. (Note that if $\phi(x) = 0$ then $\phi(x + 12d)$ may not equal $(1+\nu)^{12d}\phi(x)$.) □

LEMMA A.2. *For any integer x we have*

$$\max\{\phi(24jd), \phi(x - 12d)\} \ge 2\phi(x)/3$$

Proof. If $\phi(x - 12d) > 0$, then $2\phi(x)/3 \le \phi(x - 12d)$ by Lemma A.1. Otherwise, $x - 12d \le 24jd - 11d$, which implies that $x \le 24jd + d$. Therefore, $\phi(x) \le \phi(24jd + d) \le \phi(24jd)(1+\nu)^d \le 3\phi(24jd)/2$. □

LEMMA A.3. *For any integers x and y, if $\phi(x) > 0$ and $x - y \ge 11d$, then we have $\phi(x) - \phi(y) \ge 2(\phi(x + 11d) - \phi(y))/5$.*

Proof.

$$\begin{aligned}2(\phi(x+11d)-\phi(y))/5 &= 2(\phi(x+11d)-\phi(x))/5\\ &\quad +2(\phi(x)-\phi(y))/5\\ &\leq 2(1+\nu)^{11d}(\phi(x)-\phi(x-11d))/5\\ &\quad +2(\phi(x)-\phi(y))/5\\ &\leq 2(1+\nu)^{11d}(\phi(x)-\phi(y))/5\\ &\quad +2(\phi(x)-\phi(y))/5\\ &\leq \phi(x)-\phi(y).\end{aligned}$$

(In the second equation we use: $x-11d \geq y$. In the last equation we use: $(1+\nu)^{11d} \leq 3/2$.) ☐

Appendix B. Proof of Lemma 5.2. We define a notion of *goodness* of the tokens. Initially, all tokens are unmarked. After any step t, for every token p that is moved along an edge, p is marked *good* if $h_{t-1}(p)-h_t(p) \geq 6d$; otherwise p is marked *bad*. The marking of tokens that do not move is unchanged.

LEMMA B.1. *For any two bad tokens p_1 and p_2 present at any node v at the start of any step t, if p_1 and p_2 are last sent to v by the same neighbor u of v, then $|h_t(p_1)-h_t(p_2)| > 4d$.*

Proof. Let t_1 (resp., t_2) be the steps during which p_1 (resp., p_2) are last sent to v. Without loss of generality, we assume $t_1 < t_2 < t$. Thus we have $h_t(p_1) < h_t(p_2)$. Since u's estimate of the number of tokens at v is updated in step t_1, we have $e^u_{t_1}(v) \geq \rho + h_{t_1}(p_1) - d$. (Note that $e^u_{t_1}(v)$ is u's estimate of the number of tokens at v after step t_1.) Since p_1 remains at v during the interval $[t_1, t_2)$, we find that $e^u_{t'}(v) \geq \rho + h_{t'}(p_1) - d$ for every step t' in $[t_1, t_2)$. In particular, we have $e^u_{t_2-1}(v) \geq \rho + h_{t_2-1}(p_1) - d$. Since u sends p_2 to v in step t_2, $h_{t_2-1}(p_2) \geq h_{t_2-1}(u) - d \geq e^u_{t_2-1}(v) - \rho + 11d \geq h_{t_2-1}(p_1) + 10d$. Since p_2 is bad, we also have $h_{t_2}(p_2) > h_{t_2-1}(p_2) - 6d \geq h_{t_2-1}(p_1) + 4d$. Since $h_t(p_2) = h_{t_2}(p_2)$ and $h_t(p_1) = h_{t_2-1}(p_1)$, the lemma follows. ☐

COROLLARY B.1.1. *At any time, for any node u and integer $i > 0$, there are at most d bad tokens with heights in $(i, i+4d]$.* ☐

Proof of Lemma 5.2. Consider an arbitrary step t of the algorithm. For every token p transferred from u to v in step t, we assign some credit to every edge adjacent to u or v. Specifically, if p is marked good after step t we assign an *outgoing credit* of $9(\phi(h_{t-1}(p)) - \phi(h_t(p)))/(20d)$ units to every edge adjacent to u and an *incoming credit* of the same amount to every edge adjacent to v. If p is marked bad we assign an outgoing credit of $(\phi(h_t(p)+d) - \phi(h_t(p)))/(20d) + (\phi(h_{t-1}(p)) - \phi(h_{t-1}(p)-d))$ units to every edge adjacent to u and an incoming credit of the same amount to every edge adjacent to v. Also, for each edge (u,v), we assign an *initial credit* of $2\max\{\phi(24jd), (\phi(h_0(u)-d)+\phi(h_0(v)-d))\}$ units at the start of the analysis. The total initial credit I is bounded as follows:

$$\begin{aligned}I &\leq 2\binom{n}{2}\phi(24jd) + \sum_{(u,v)\in E} 2(\phi(h_0(u)-d)+\phi(h_0(v)-d))\\ &\leq n^2\phi(24jd) + \sum_{u\in V}\sum_{0\leq \ell<d} 2\phi(h_0(u)-\ell)\\ &\leq n^2\phi(24jd) + 2\Phi_0.\end{aligned}$$

(The first equation follows from the fact that the maximum of two quantities is at most the sum of the particular quantities. We also note that each undirected edge (u,v) appears at most once in the summation. For the second equation, we note that each node has at most d edges. Hence for any node u, the term $2\phi(h_0(u)-d)$ appears in at most d terms of the sum. We complete the derivation of the second equation by observing that $\phi(h_0(u)-\ell)$ is at least $\phi(h_0(u)-d)$ for $0 \leq \ell < d$. The third equation is obtained by the fact that $\sum_{0\leq\ell<d}\phi(h_0(u)-\ell)$ is at most $\phi(u)$.) The above bound on I corresponds to the negative term in Equation (4).

We now show that using the above accounting method, we can account for the amortized potential drop of $(\phi(h_{t-1}(u)-d) - \phi(h_{t-1}(v)+d))/50$ units at step t for every edge $(u,v) \in E_t$. To accomplish this, for every live edge (u,v) ((u,v) not necessarily in E_t), we consider three cases: (i) a token p sent from u to v is marked good, (ii) a token p sent from u to v is marked bad, (iii) no token is sent from u to v.

We first consider case (i). When a token p is marked good after being sent along (u,v), we use the actual potential drop of p to pay for the amortized drop D_1 associated with (u,v) as well as the total credit D_2 assigned to the edges adjacent to u or v due to the transfer of a good token.

$$\begin{aligned} D_1 + D_2 &\le (\phi(h_{t-1}(u) - d) - \phi(h_{t-1}(v) + d))/50 + 2d[9(\phi(h_{t-1}(p)) - \phi(h_t(p)))]/(20d) \\ &\le (\phi(h_{t-1}(p)) - \phi(h_t(p)))/50 + 9(\phi(h_{t-1}(p)) - \phi(h_t(p)))/10 \\ &\le \phi(h_{t-1}(p)) - \phi(h_t(p)). \end{aligned}$$

(The first term in the right-hand side of the first equation is the amortized potential drop. The second term is an upper bound on D_2 since the number of edges adjacent to either u or v is at most $2d$. The second equation follows from the fact that $h_{t-1}(p)$ is at least $h_{t-1}(u) - d$ and $h_t(p)$ is at most $h_t(u) + d$.)

We now consider case (ii). In this case we need to account for: (1) if $h_t(p) > h_{t-1}(p)$, an amount equal to the potential increase of $D_1 = \phi(h_t(p)) - \phi(h_{t-1}(p))$ units, and (2) a credit of at most $(\phi(h_t(p) + d) - \phi(h_t(p)))/10 + (\phi(h_{t-1}(p)) - \phi(h_{t-1}(p) - d))/10$ units. We pay for $(\phi(h_{t-1}(p)) - \phi(h_t(p)))/10$ units of the credit using the potential change. The remainder of the credit we need to account for is at most the sum of $D_2 = (\phi(h_t(p) + d) - \phi(h_t(p)))/10$ and $D_3 = (\phi(h_t(p)) - \phi(h_{t-1}(p) - d))/10$. (Note that this is true regardless of whether the potential of p decreases or not in step t.)

We have two subcases, depending on whether t is the first step (u, v) is live (subcase (a)) or not (subcase (b)). In subcase (a), if $h_0(u) \ge h_t(p) - d$, the initial credit C_0 associated with (u, v) is at least $2\max\{\phi(24jd), \phi(h_t(p) - 2d)\}$. Since $\phi(h_t(p) - 2d) \ge \phi(h_t(p) - 12d)$, it follows from Lemma A.2 that $3C_0/4 \ge \phi(h_t(p)) \ge D_1$. Since $\phi(h_t(p) - 2d) \ge \phi(h_t(p) - 11d)$, $C_0/4 \ge \phi(h_t(p) + d)/3 \ge \phi(h_t(p) + d)/10 + \phi(h_t(p))/10 \ge D_2 + D_3$. Therefore, we have $C_0 \ge D_1 + D_2 + D_3$.

We now consider subcase (a) under the assumption that $h_0(v) \le h_t(p) - d$. In order to do the accounting, we use part of the incoming credit associated with the edge (u, v) due to the set X of good tokens of v with heights in the interval $(h_0(v), h_t(p) - d]$. (Note that each token in X is marked and thus, has contributed incoming credit to all edges adjacent to v.) Since each token x in X is good, the height of the token before the transfer to node v was at least $h_t(q) + 6d$. Therefore, the incoming credit assigned to (u, v) by a token x in X is at least $9(\phi(h_t(q) + 6d) - \phi(h_t(q)))/(20d)$ units. For each token x in X, we use $c_x = 8(\phi(h_t(q) + 6d) - \phi(h_t(q)))/(20d)$ units of this incoming credit. Let C_1 denote $\sum_{x \in X} c_x$. We obtain the following lower bound C_1. By invoking Corollary B.1.1, we obtain:

$$\begin{aligned} C_1 &\ge \frac{8}{20d} \sum_{1 \le i \le \lfloor \frac{h_t(p) - d - h_0(v)}{4d} \rfloor} \sum_{1 \le k \le 3d} (\phi(h_t(p) - d - 4id + k + 6d) - \phi(h_t(p) - d - 4id + k + d)) \\ &\ge \frac{8}{20d} \sum_{1 \le k \le 3d} \sum_{1 \le i \le \lfloor \frac{h_t(p) - d - h_0(v)}{4d} \rfloor} (\phi(h_t(p) - d - 4id + k + 6d) - \phi(h_t(p) - 4id + k)) \\ &\ge \frac{8}{20d} \sum_{1 \le k \le 3d} \left(\phi(h_t(p) - d - 4d + k + 6d) - \phi(h_t(p) - 4d \lfloor \frac{h_t(p) - d - h_0(v)}{4d} \rfloor + k) \right) \\ &\ge \frac{8}{20d} \sum_{1 \le k \le 3d} (\phi(h_t(p) + d) - \phi(h_0(v) + 8d)) \\ &= 6(\phi(h_t(p) + d) - \phi(h_0(v) + 8d))/5. \end{aligned}$$

(In the first equation we partition the interval $(h_0(v), h_t(p) - d]$ into subintervals of $4d$ consecutive integers starting from $h_t(p) - d$. The last subinterval may have fewer than $4d$ integers; if so, we ignore the last subinterval in the sum. The second summation in the first equation is a lower bound on the sum of c_x over each good token x in each subinterval. To obtain the second summation, we invoke Corollary B.1.1 which implies that there are at least $3d$ good tokens in every subinterval of $4d$ tokens. The second equation is obtained by interchanging the order of sums. For the third equation, we use the fact that $\phi(h_t(p) - d - 4(i-1)d + k + 6d) \ge \phi(h_t(p) - 4id + k)$ and then note that the sum telescopes. For the fourth inequaility, note that: (i) the index k is at least 0 and at most $3d$, and (ii) $h_t(p) - 4d\lfloor \frac{h_t(p) - d - h_0(v)}{4d} \rfloor \le h_0(v) + 5d$.)

Since p is marked bad after step t, we have $h_t(p) > h_{t-1}(p) - 6d$. Therefore,

$$\begin{aligned} C_0 + C_1 &\ge 2\max\{\phi(24jd), \phi(h_0(v) - d)\} + 6(\phi(h_t(p) + d) - \phi(h_0(v) + 8d))/5 \\ &\ge 6\phi(h_t(p) + d)/5 \\ &\ge \phi(h_t(p)) - \phi(h_{t-1}(p)) + (\phi(h_t(p) + d) - \phi(h_t(p)))/10 + (\phi(h_t(p)) - \phi(h_{t-1}(p) - d))/10 \\ &\ge D_1 + D_2 + D_3. \end{aligned}$$

(The first equation states the lower bounds on C_0 and C_1 obtained above. For the second equation, we invoke Lemma A.2 as follows: $2\max\{\phi(24jd), \phi(h_0(v)-d)\} \geq 4\phi(h_0(v)+11d)/3 \geq 6\phi(h_0(v)+8d)/5$. The third equation is obtained from the following three observations: (i) $\phi(h_t(p)+d) \geq \phi(h_t(p)) - \phi(h_{t-1}(p))$, (ii) $\phi(h_t(p)+d)/10 \geq (\phi(h_t(p)+d)-\phi(h_t(p)))/10$, and (iii) $\phi(h_t(p)+d)/10 \geq (\phi(h_t(p)) - \phi(h_{t-1}(p)-d))/10$.)

We use a similar argument as above to handle subcase (b) where t is not the first step in which (u,v) is live. The set X is the set of good tokens of v with heights in the interval $(e^u_{t-1}(v)-\rho, h_t(p)-d]$. Let c_x and C_1 be defined as in subcase (a). That is, c_x equals $8(\phi(h_t(x)+6d)-\phi(h_t(x)))/(20d)$ units of the incoming credit assigned to (u,v) by a token x in X, and C_1 equals $\sum_{x \in X} c_x$. We will show that $11C_1/12 \geq D_1 + D_3$, and $C_1/12 \geq D_2$, and hence obtain that $C_1 \geq D_1 + D_2 + D_3$.

We first show that $11C_1/12 \geq D_1 + D_3$. If $h_t(p) \leq h_{t-1}(p) - d$, then D_1, and D_3 are both nonpositive and hence the desired claim holds trivially. We now assume that $h_t(p) > h_{t-1}(p) - d$. Let y denote $e^u_{t-1}(v) - \rho + 8d$. We observe that since u sent a token to v during step t, $y = e^u_{t-1}(v) - \rho + 8d \leq h_{t-1}(u) - 4d \leq h_{t-1}(p) - 3d$. Since p is a bad token, we have $y \leq h_{t-1}(p) - 3d < h_t(p) - 2d$. As in subcase (a), we divide the interval $(e^u_{t-1}(v)-\rho, h_t(p)-d]$ into subintervals consisting of $4d$ consecutive integers. Note that $e^u_{t-1}(v) - \rho \leq h_t(p) - 11d$, and hence the number of subintervals is at least 1. We obtain the following lower bound on $11C_1/12$.

$$\begin{aligned}
11C_1/12 &\geq (11/12)\cdot 6(\phi(h_t(p)+d)-\phi(y))/5 \\
&\geq 11(\phi(h_t(p)+d) - \phi(h_{t-1}(p)-2d))/10 \\
&\geq (\phi(h_t(p)) - \phi(h_{t-1}(p))) + (\phi(h_t(p)) - \phi(h_{t-1}(p)-d))/10 \\
&= D_1 + D_3
\end{aligned}$$

(The first equation is obtained in the same manner as the upper bound on C_1 in subcase (a). While the interval considered in subcase (a) is $(h_0(v), h_t(p)-d]$, we consider here the interval $(e^u_{t-1}(v)-\rho, h_t(p)-d] = [y-8d, h_t(p)-d]$. Hence, the term $\phi(h_0(v)+8d)$ obtained in the lower bound on C_1 in subcase (a) is replaced by $\phi(y)$ above. The second equation is obtained by the upper bound on y.)

We now show that $C_1/12 \geq D_2$. Since a token is sent by u to v in step t, $e^u_{t-1}(u) - \rho \leq h_{t-1}(u) - 12d \leq h_{t-1}(p) - 11d$. Moreover, since p is a bad token, $h_{t-1}(p) \leq h_t(p) - 6d$. Therefore, $e^u_{t-1}(u) - \rho \leq h_t(p) - 5d$. It follows that $(h_t(p)-5d, h_t(p)-d]$ is a subinterval of $(e^u_{t-1}(u)-\rho, h_t(p)-d]$. Hence, C_1 can be lower bounded by adding c_x over all good tokens x whose height is in $(h_t(p)-5d, h_t(p)-d]$. By Corollary B.1.1, at least $3d$ of the tokens in $[h_t(p)-5d, h_t(p)-d]$ are good. We thus obtain:

$$\begin{aligned}
C_1/12 &\geq (3d/12)\cdot 8(\phi(h_t(p)+d)-\phi(h_t(p)-d))/(20d) \\
&= (\phi(h_t(p)+d) - \phi(h_t(p))/10 \\
&\geq D_2.
\end{aligned}$$

(For the first equation, note that $c_x = 8(\phi(h_t(x)+6d)-\phi(h_t(x)))/(20d) \geq 8(\phi(h_t(p)+d)-\phi(h_t(p)-d))/(20d)$, for $h_t(x)$ in $[h_t(p)-5d, h_t(p)-d]$. The last equation follows from the definition of D_2.)

To complete the proof for case (ii), we show that for any token x of v, any incoming credit assigned by x to edge (u,v) that is used at step t for case (ii) is not used again for case (ii). To prove this, we note that for any x in X, for every further step $t' > t$ until x is transferred by u, we have $h_{t'}(x) \geq e^u_{t'-1}(v) - \rho$. While establishing case (ii) for step t, we only use the incoming credit assigned by tokens in $(e^u_{t'-1}(v)-\rho, h_{t'}(p)-d]$. Hence the incoming credit assigned by x to edges adjacent to u that is used at step t will never be used again.

We need to consider case (iii) only under the assumption that $(u,v) \in E_t$, i.e., (u,v) is live in step t. In this case we account for $D = (\phi(h_{t-1}(u)-d) - \phi(h_{t-1}(v)+d))/50$ units of potential. Again we consider two subcases depending on whether t is the last step in which (u,v) is live (subcase (a)) or not (subcase (b)). We first consider subcase (a). If $h_0(u) \geq h_{t-1}(u) - 12d$, then we use $C_0 = 2\max\{\phi(24jd), \phi(h_0(u)-d)\}$ units of the initial credit associated with (u,v). Since $h_{t-1}(u) - d \leq h_0(u) - d + 12d$, it follows from Lemma A.2 that $C_0 \geq 4\phi(h_{t-1}(u)-d)/3 \geq \phi(h_{t-1}(u)-d)/50 \geq D$.

We now consider subcase (a) of case (iii) under the assumption that $h_0(u) < h_{t-1}(u) - 12d$. In addition to C_0, we also use part of the incoming credit associated with the set of tokens $Y = \{y : y \text{ is a token of } u \text{ and } h_0(u) < h_t(y) \leq h_{t-1}(u)\}$. Specifically, for every token y in Y, we use $(\phi(h_t(y)+d) - \phi(h_t(y)))/(20d)$ units of incoming credit that is assigned to (u,v) by y. Note that since $h_t(y) > h_0(u)$, token y has moved and hence has assigned some incoming credit to (u,v). If y is good, this credit is at least $9(\phi(h_t(y)+6d) - \phi(h_t(y)))/(20d)$ units; otherwise, this credit is at least $(\phi(h_t(y)+d) - \phi(h_t(y)))/(20d)$. (u,v). Moreover, if y is a good token at most $8(\phi(h_t(y)+6d) - \phi(h_t(y)))/(20d)$ units of incoming credit were used in the analysis of case (ii). If y is a bad token, none of the incoming credit was used in the analysis of case (ii). In either case, at least $(\phi(h_t(y)+d)-\phi(h_t(y)))/(20d)$ units of incoming credit still remain. Let this credit be denoted C_1. We obtain the following lower bound on C_0+C_1.

$$\begin{aligned}
C_0+C_1 &\geq C_0+\sum_{h_0(u)<k\leq h_{t-1}(u)}(\phi(k+d)-\phi(k))/(20d)\\
&= C_0+\frac{1}{20d}\sum_{1\leq i\leq d}(\phi(h_{t-1}(u)+i)-\phi(h_0(u)+i))\\
&\geq C_0+(\phi(h_{t-1}(u))-\phi(h_0(u)+d))/20\\
&\geq \phi(h_{t-1}(u))/20\\
&\geq D.
\end{aligned}$$

(The second equation holds since the sum in the first equation can be expressed as a sum of d telescoping sums. For the third equation we invoke Lemma A.2 and obtain that $C_0 \geq 2\phi(h_0(u)+11d)/3 \geq \phi(h_0(u)+d)/20$.)

We now consider subcase (b) of (iii). Recall that by the definition of E_t, u is in $S_{>j}$ at the start of step t. Therefore, $h_{t-1}(u) \geq 24(j+1)d-12d \geq 24jd+12d$. Since no token was sent along (u,v) in step t, we have $e^u_{t-1}(v)-\rho > h_{t-1}(u)-12d$ $(\geq 24jd)$. By the definition of E_t, we also have $h_{t-1}(u) \geq h_{t-1}(v)+24d$. It follows that $e^u_{t-1}(v)-\rho > h_{t-1}(v)+12d$. Since the last step in which (u,v) was live, at least $e^u_{t-1}(v)-\rho-h_{t-1}(v)$ tokens have left v. We use the outgoing credit assigned to (u,v) due to these token transmissions. Consider a token x that is transmitted by v in step t'. If x is marked good after the step, then the outgoing credit assigned by x to (u,v) is at least $9(\phi(h_{t'-1}(p))-\phi(h_{t'}(p)))/(20d) \geq 9(\phi(h_{t'-1}(p))-\phi(h_{t'-1}(p)-6d))/(20d)$ units. Otherwise, the outgoing credit assigned by x to (u,v) is at least $(\phi(h_{t'-1}(p))-\phi(h_{t'-1}(p)-d))/(20d)$ units. In either case, the outgoing credit is at least $(\phi(h_{t'-1}(p))-\phi(h_{t'-1}(p)-d))/(20d)$ units. We thus obtain the following lower bound on the total outgoing credit C_2 assigned to (u,v) by the at least $e^u_{t-1}(v)-\rho-h_{t-1}(v)$ tokens.

$$\begin{aligned}
C_2 &\geq \sum_{h_{t-1}(v)<k\leq e^u_{t-1}(v)-\rho}(\phi(k)-\phi(k-d))/(20d)\\
&= \frac{1}{20d}\sum_{1\leq i\leq d}(\phi(e^u_{t-1}(v)-\rho-d+i)-\phi(h_{t-1}(v)-d+i))\\
&\geq (\phi(e^u_{t-1}(v)-\rho-d)-\phi(h_{t-1}(v)))/20\\
&\geq (\phi(e^u_{t-1}(v)-\rho+11d)-\phi(h_{t-1}(v)+d))/50\\
&\geq (\phi(h_{t-1}(u)-d)-\phi(h_{t-1}(v)+d))/50\\
&= D.
\end{aligned}$$

(The second equation holds since the sum in the first equation can be expressed as a sum of d telescoping sums. For the third and fourth inequalities, we first note that since no token was sent by u to v in step t, we have $e^u_{t-1}(v)-\rho > h_{t-1}(u)-12d \geq 24jd-d$. The third equation now follows from Lemma A.3 and the fact that $\phi(e^u_{t-1}(v)-\rho-d) > 0$. The fourth equation holds follows directly from the lower bound on $e^u_{t-1}(v)-\rho$.)

We note that the outgoing credit assigned to edge (u,v) in the above analysis of case (iii) is used at most once in case (iii). To prove this, we observe that after step t, the value of $e^u(v)$ is updated by u to $h_{t-1}(v)+\rho$. Therefore, if case (iii) of the analysis subsequently uses any outgoing credit assigned by a token x that leaves v and whose height in v is in $(h_{t-1}(v), e^u_{t-1}(v)]$, then x must have arrived into v after step t. Hence, the outgoing credit assigned by the $e^u_{t-1}(v)-h_{t-1}(v)$ tokens that are used in the analysis for step t are not used again for a later step. □

REFERENCES

[1] Y. Afek, E. Gafni, and A. Rosen. The slide mechanism with applications in dynamic networks. In *Proceedings of the 11th ACM Symposium on Principles of Distributed Computing*, pages 35–46, August 1992.

[2] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao. Approximate load balancing on dynamic and asynchronous networks. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pages 632–641, May 1993.

[3] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c\log n$ parallel steps. *Combinatorica*, 3:1–19, 1983.

[4] J. Aspnes, M. Herlihy, and N. Shavit. Counting networks and multiprocessor co-ordination. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 348–358, May 1991.

[5] B. Awerbuch and T. Leighton. A simple local-control approximation algorithm for multi-commodity flow. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 459–468, October 1993.

[6] B. Awerbuch and T. Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pages 487–496, May 1994.

[7] B. Awerbuch, Y. Mansour, and N. Shavit. End-to-end communication with polynomial overhead. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 358–363, October 1989.

[8] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[9] A. Broder, A. M. Frieze, E. Shamir, and E. Upfal. Near-perfect token distribution. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, pages 308–317, July 1992.

[10] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[11] Y. C. Chow and W. Kohler. Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Transactions on Computers*, C–28(5):57–68, November 1980.

[12] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.

[13] D. Eager, E. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, SE–12(5):662–675, 1986.

[14] B. Ghosh and S. Muthukrishnan. Dynamic load balancing on parallel and distributed networks by random matchings. In *Proceedings of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 226–235, June 1994.

[15] B. Ghosh, S. Muthukrishnan, and M. H. Schultz. First and second order diffusive methods for rapid, coarse, distributed load balancing. In *Proceedings of the 8th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 72–81, June 1996.

[16] B. Goldberg and P. Hudak. Implementing functional programs on a hypercube multiprocessor. In *Proceedings of the 4th Conference on Hypercubes, Concurrent Computers and Applications*, volume 1, pages 489–503, 1989.

[17] A. Heirich and S. Taylor. A parabolic theory of load balance. Research Report Caltech-CS-TR-93-25, Caltech Scalable Concurrent Computation Lab, Pasadena, CA, March 1993.

[18] K. T. Herley. A note on the token distribution problem. *Information Processing Letters*, 28:329–334, 1991.

[19] S. H. Hosseini, B. Litow, M. Malkawi, J. McPherson, and K. Vairavan. Analysis of a graph coloring based distributed load balancing algorithm. *Journal of Parallel and Distributed Computing*, 10(2):160–166, 1990.

[20] J. JáJá and K. W. Ryu. Load balancing and routing on the hypercube and related networks. *Journal of Parallel and Distributed Computing*, 14(4):431–435, 1992.

[21] M. R. Jerrum and A. Sinclair. Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 235–244, May 1988.

[22] R. Karp and Y. Zhang. A randomized parallel branch-and-bound procedure. *Journal of the ACM*, 40:765–789, 1993.

[23] M. Klugerman and C. G. Plaxton. Small depth counting networks. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, pages 417–428, May 1992.

[24] T. Leighton, C. E. Leiserson, and D. Kravets. Theory of parallel and VLSI computation. Research Seminar Series Report MIT/LCS/RSS 8, MIT Laboratory for Computer Science, May 1990.

[25] F. C. H. Lin and R. M. Keller. The gradient model load balancing method. *IEEE Transactions on Software Engineering*, SE–13(1):32–38, 1987.

[26] R. Lüling and B. Monien. Load balancing for distributed branch and bound algorithms. In *Proceedings of the 6th International Parallel Processing Symposium*, pages 543–549, March 1992.

[27] R. Lüling and B. Monien. A dynamic distributed load balancing algorithm with provable good performance. In *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 164–172, June 1993.

[28] R. Lüling, B. Monien, and F. Ramme. Load balancing in large networks: A comparative study. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing*, pages 686–689, December 1991.

[29] F. Meyer auf der Heide, B. Oesterdiekhoff, and R. Wanka. Strongly adaptive token distribution. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming*, pages 398–409, July 1993.

[30] M. Mihail. Conductance and convergence of Markov chains – a combinatorial treatment of expanders. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 526–531, October 1989.

[31] L. M. Ni, C. Xu, and T. B. Gendreau. Distributed drafting algorithm for load balancing. *IEEE Transactions on Software Engineering*, SE–11(10):1153–1161, October 1985.

[32] D. Peleg and E. Upfal. The generalized packet routing problem. *Theoretical Computer Science*, 53:281–293, 1987.

[33] D. Peleg and E. Upfal. The token distribution problem. *SIAM Journal on Computing*, 18:229–243, 1989.

[34] C. G. Plaxton. Load balancing, selection and sorting on the hypercube. In *Proceedings of the 1989 ACM Symposium on Parallel Algorithms and Architectures*, pages 64–73, June 1989.

[35] J. Stankovic. Simulations of three adaptive, decentralized controlled, job scheduling algorithms. *Computer Networks*, 8:199–217, 1984.

[36] R. Subramanian and I. D. Scherson. An analysis of diffusive load balancing. In *Proceedings of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 220–225, June 1994.

[37] A. N. Tantawi and D. Towsley. Optimal static load balancing in distributed computer systems. *Journal of the ACM*, 32:445–465, April 1985.

[38] E. Upfal. An $O(\log N)$ deterministic packet routing scheme. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 241–250, May 1989.

[39] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and Experience*, 3(5):457–481, 1991.

[40] C. Z. Xu and F. C. M. Lau. Iterative dynamic load balancing in multicomputers. *Journal of the Operational Research Society*, 45(7):786–796, 1994.

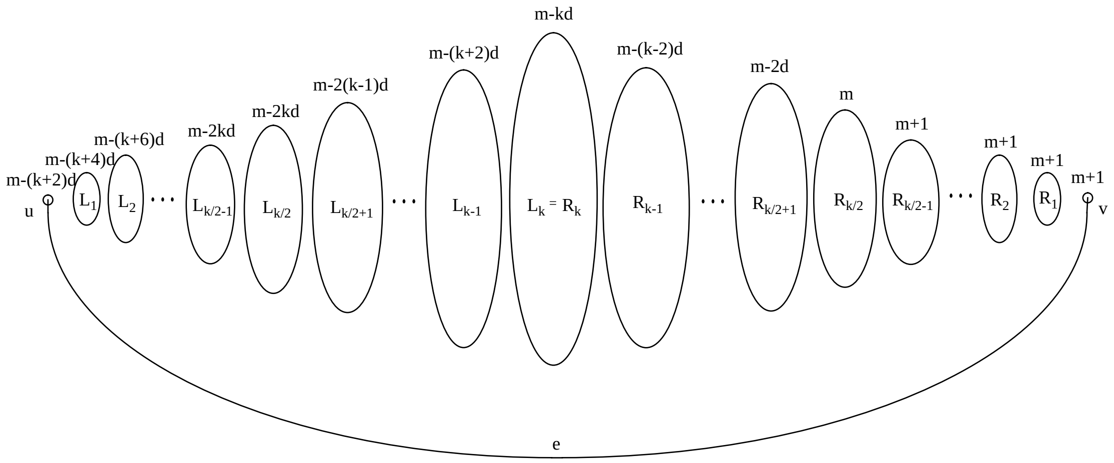

FIG. 2. *The initial distribution of tokens on G for the first case.*

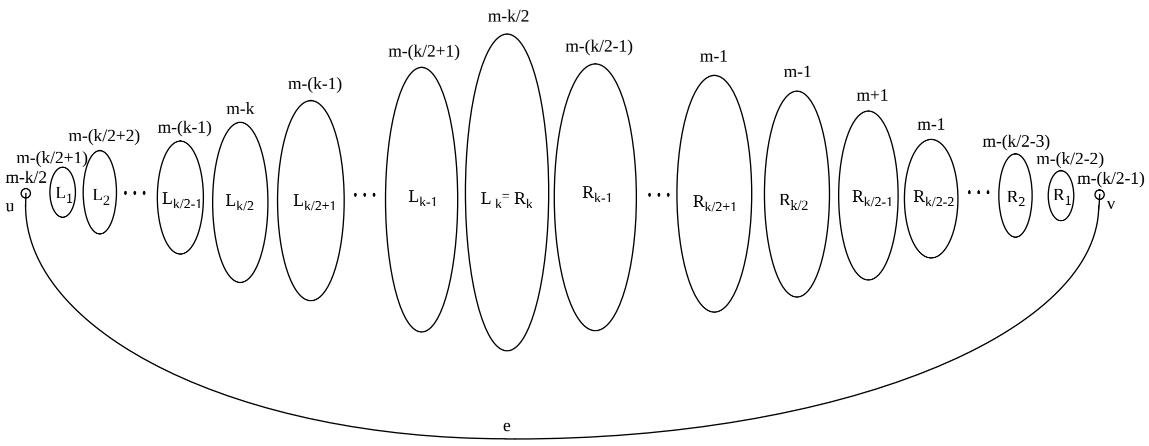

FIG. 3. *The initial distribution of tokens on G for the second case.*