Static and Dynamic Path Selection on Expander Graphs: A Random Walk Approach

Andrei Z. Broder[*] Alan M. Frieze[†] Eli Upfal[‡]

Abstract

This paper addresses the problem of virtual circuit switching in bounded degree expander graphs. We study the static and dynamic versions of this problem. Our solutions are based on the rapidly mixing properties of random walks on expander graphs.

In the static version of the problem an algorithm is required to route a path between each of K pairs of vertices so that no edge is used by more than g paths. A natural approach to this problem is through a multi-commodity flow reduction. However, we show that the random walk approach leads to significantly stronger results than those recently obtained by Leighton and Rao [13] using the multi-commodity flow setup.

In the dynamic version of the problem connection requests are continuously injected into the network. Once a connection is established it utilizes a path (a virtual circuit) for a certain time until the communication terminates and the path is deleted. Again each edge in the network should not be used by more than g paths at once.

The dynamic version is a better model for the practical use of communication networks. Our random walk approach gives a simple and fully distributed solution for this problem. We show that if the injection to the network and the duration of connections are both controlled by Poisson processes then our algorithm achieves a steady state utilization of the network which is similar to the utilization achieved in the static case situation.

1 Introduction

Communication protocols for high-speed high bandwidth networks are based on *virtual circuit switching.* The speed of the network does not allow for on-line routing of individual packets. Instead, upon establishing a connection, bandwidth is allocated

[*]Digital Systems Research Center, 130 Lytton Ave, Palo Alto, CA 94301

[†]Department of Mathematics, Carnegie-Mellon University. A portion of this work was done while the author was visiting Digital SRC. Supported in part by NSF grants CCR-9225008 and CCR-9530974.

[‡]IBM Almaden Research Center, San Jose, CA 95120, and Department of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel.

along a path connecting the two endpoints for the duration of the connection. These "virtual circuits" are set up on a per-call basis and are disconnected when the call is terminated. Efficient utilization of the network depends on the allocation of virtual circuits between pairs of nodes so that no link is overloaded beyond its capacity.

As in other routing problems we distinguish between a static and a dynamic version. In the static version all the requests are given at once and must be simultaneously satisfied. In the dynamic version requests are continuously generated. A connection once established, continues for a certain amount of time and afterwards its bandwidth can be re-used for other connections.

The static version of this problem translates into the following combinatorial question: Given a network $G = (V, E)$ and a set of K pairs of vertices in V, find for each pair (a_i, b_i), a path connecting a_i to b_i, such that no edge is used by more than g paths. For arbitrary graphs, the related decision problem is in $\mathcal{P}$ for fixed K - Robertson and Seymour [15], but is $\mathcal{NP}$-complete if K is part of the input.

In contrast to the negative results for general graphs a significant progress has been made in solving this problem for the interesting class of bounded degree expander graphs. In particular Leighton and Rao [13] have recently obtained a number of constructive and existential results for this problem based on the natural linear relaxation of the circuit switching problem to a multi-commodity flow problem. This paper explores a different approach: we study solutions for this problem based on the rapidly mixing properties of random walks on expander graphs. The method we explore here builds on the basic technique developed in [6]. Using the random walk approach we improve the Leighton-Rao results for the static case and obtain the first non-trivial results for the dynamic version of the virtual circuit switching problem.

Our first result for the static case establishes a tradeoff between the number of pairs K, and the allowed congestion g.

Logarithms are natural unless explicitly stated otherwise.

Theorem 1 *There is an explicit polynomial time algorithm that can connect any set of $K = \alpha(n)n/\log n$ pairs of vertices on a bounded degree expander so that no edge is used by more than g paths where*

$$g = \begin{cases} O\left(s + \left\lceil \dfrac{\log\log n}{\log(1/\hat{\alpha})} \right\rceil\right), & \textit{for } \alpha < 1/2; \\ O(s + \alpha + \log\log n), & \textit{for } \alpha \geq 1/2, \end{cases}$$

$\hat{\alpha} = \min(\alpha, 1/\log\log n)$, and s is the maximal multiplicity of a vertex in the set of pairs.

Since the expected distance between two random vertices on a bounded degree expander is $\Omega(\log n)$ our results are always within an additive factor of $O(\log\log n)$ from optimal. Our theorem improves the results obtained by Leighton and Rao [13] using the multi-commodity flow approach. The case $\alpha(n) = 1/(\log n)^\epsilon$ improves the bound of Theorem 1 of [13], the case $\alpha(n) = O(1)$ improves the bound of Theorem 2 in [13], and makes the result constructive.

We also note that Kleinberg and Rubinfeld [10] have recently used our result in their analysis of a greedy algorithm for finding *short* disjoint paths on expanders.

We next show constructively that $K = n/(\log n)^{2+\epsilon}$ pairs of vertices can be connected by *edge-disjoint* paths, provided the graph has sufficiently strong expansion.

Theorem 2 *Suppose* $\epsilon > 0$ *and* G *is an* r*-regular* (α, β, γ)*-expander (See Section 2 for definitions). If conditions (11),(12) and (13) below hold and* r *is sufficiently large then any* $K = n/(\log n)^{2+\epsilon}$ *pairs of vertices can be connected by edge-disjoint paths. Furthermore these paths can be explicitly found in polynomial time.*

The following non-constructive result removes the ϵ and improves Theorem 3 of [13]:

Theorem 3 *Given a bounded degree* (α, β, γ)*-expander graph (*$\alpha < 1/4$*) there exist parameters* c *and* r_0 *that depend only on* α, β, γ, *but not on* n, *such if* $r \geq r_0$ *and then any set of less than* $cn/(\log n)^2$ *disjoint pairs of vertices can be connected by edge disjoint paths.*

The advantage of the random walks approach is even more significant in the solution of the dynamic virtual circuit switching problems. In practice networks are rarely used in the "batch mode" modeled by the static problem. Real-life network performance is better modeled by a dynamic process whereby requests for connection are continuously arriving at the nodes of the network. A connection has a duration time, and once the communication has terminated its bandwidth can be used for another connection. In section 5 we formulate a model for studying the dynamic virtual circuit switching problem under stochastic assumptions about the injection rate of new requests and the duration of connections.

Using the random walk approach we develop a simple and fully distributed protocol for dynamic path selection on bounded degree expander graphs. For the analysis we adopt the stochastic model assumed in the design of most long-distance telephone networks [12]. Requests arrive according to a Poisson process, and the duration of a connection is exponentially distributed. Our goal is to characterize the relationship between the load and congestion parameters that guarantees system stability (i.e., the expected number of requests in queue is not growing unboundly with time).

Theorem 4 *Let* $\mathbf{E}(N)$ *be the expected number of requests that arrive to the* n*-node expander network at a given step, let* $\mathbf{E}(D)$ *be the expected duration of a connection. If* $\mathbf{E}(N)\,\mathbf{E}(D) \leq \alpha(n)n/\log n$, *where* $\alpha(n) \leq 1/2$, *then there exists a constant* c *such that for* $g \geq c\lceil \log\log n/\log(1/\alpha)\rceil$ *the system running our dynamic algorithm is stable, and the expected time a request waits in a queue is* $O(n/\mathbf{E}(N))$.

Since $\mathbf{E}(N)\,\mathbf{E}(D)$ is the expected number of paths that must be active in the network at the steady state in order to keep the system stable, we see that our dynamic algorithm achieves almost the same edge utilization as that in our static results.

One should note that out approach differs from the work on admission control [4, 9, 5] in that we do not reject requests. All requests are eventually satisfied in our model, but not immediately. In contrast, in the admission control model a request is either immediately satisfied or it is rejected. Our approach better models computer communication, while the admission control approach is a better model for human (telephone) communication.

2 Preliminaries

There are various ways to define expander graphs; here we define them in terms of edge expansion (a weaker property than vertex expansion).

Let $G = (V, E)$ be a graph. For a set of vertices $S \subset V$ let $\mathrm{out}(S)$ be the set of edges with one end-point in S and one end-point in $V \setminus S$, that is

$$\mathrm{out}(S) = \Big\{ \{u, v\} \mid \{u, v\} \in E, u \in S, v \notin S \Big\}.$$

Similarly,

$$\mathrm{in}(S) = \Big\{ \{u, v\} \mid \{u, v\} \in E, u, v \in S \Big\}..$$

Definition 1 *A graph $G = (V, E)$ is a β-expander, if for every set $S \subset V$, $|S| \leq |V|/2$, we have $|\mathrm{out}(S)| \geq \beta|S|$.*

For the remainder of this paper, when β is not explicitly mentioned we will assume that it is an arbitrary constant greater than 0. For certain results we need expanders that have the property that the expansion of small sets is not too small. The form of definition given below differs slightly from that given in [6]. This allows one definition for Theorems 2 and 3.

Definition 2 *An r-regular graph $G = (V, E)$ is called an (α, β, γ)-expander if for every set $S \subset V$*

$$\mathrm{out}(S) \geq \begin{cases} (1-\alpha)r|S| & \text{if } |S| \leq \gamma|V| \\ \beta|S| & \text{if } \gamma|V| < |S| \leq |V|/2 \end{cases}$$

In particular random regular graphs and the (explicitly constructible) Ramanujan graphs of of Lubotsky, Phillips and Sarnak [11] are (α, β, γ)-expanders. (See discussion in [6].)

A *random walk* on an undirected graph $G = (V, E)$ is a Markov chain $\{X_t\} \subseteq V$ associated with a particle that moves from vertex to vertex according to the following rule: the probability of a transition from vertex i, of degree d_i, to vertex j is $1/d_i$ if $\{i, j\} \in E$, and 0 otherwise. (In case of a bi-partite graph we need to assume that we do nothing with probability 1/2 and move off with probability 1/2 only. This technicality is ignored for the remainder of the paper.) Its stationary distribution, denoted π, (or π_G) is given by $\pi(v) = d_v/(2|E|)$. Obviously, for regular graphs, the stationary distribution is uniform.

A trajectory W of length τ is a sequence of vertices $[w_0, w_1, \ldots, w_\tau]$ such that $\{w_t, w_{t+1}\} \in E$. The Markov chain $\{X_t\}$ induces a probability distribution on trajectories, namely the product of the probabilities of the transitions that define the trajectory.

Let P denote the transition probability matrix of the random walk on G, and let $P_{v,w}^{(t)}$ denote the probability that the walk is at w at step t given that it started at v. Let λ be the second largest eigenvalue of P. (All eigenvalues of P are real.) It is known that

$$|P_{v,w}^{(t)} - \pi(w)| \leq \lambda^t \sqrt{\pi(w)/\pi(v)}. \tag{1}$$

In particular, for regular graphs

$$P_{v,w}^{(t)} = \frac{1}{n} + O(\lambda^t). \tag{2}$$

To ensure rapid convergence we need $\lambda \leq 1 - \epsilon$ for some *constant* $\epsilon > 0$. This holds for all expanders (Alon [1]). In particular if G is an r-regular β-expander then Sinclair and Jerrum [16] show that

$$\lambda \leq 1 - \frac{1}{2}\left(\frac{\beta}{r}\right)^2 \tag{3}$$

It is often useful to consider the *separation* s of the distribution $P_{v,\cdot}^{(t)}$ from the limit distribution π given by

$$s(t) = \max_{v,w} \frac{\pi(w) - P_{v,w}^{(t)}}{\pi(w)}. \tag{4}$$

Then we can write

$$P_{v,\cdot}^{(t)} = (1 - s(t))\pi + s(t)\sigma$$

where σ is a probability distribution. We can then imagine that the distribution $P_{v,\cdot}^{(t)}$ is producing by choosing either σ with probability $s(t)$ or π with probability $1 - s(t)$. Hence if $\mathcal{E}$ is an event that depends only on the state of the Markov chain we have

$$(1 - s(t))\,\mathbf{Pr}(\mathcal{E} \text{ under } \pi) + s(t) \geq \mathbf{Pr}(\mathcal{E} \text{ under } P_{v,\cdot}^{(t)}) \geq (1 - s(t))\,\mathbf{Pr}(\mathcal{E} \text{ under } \pi). \tag{5}$$

We use this in the following scenario:

Experiment A: Choose $u_1 \in V$ uniformly at random and do a random walk W_1 of length τ from u_1. Let v_1 be the terminal vertex of W_1.

Experiment B: Choose u_2 and v_2 uniformly and independently from V and do a random walk of length τ from u_2 to v_2.

Here $\tau = c_0 \log n$ and $s(\tau) \leq N^{-3}K^{-2}$. Since G is regular, u_1 and v_1 and u_2 and v_2 have each the same (uniform) distribution in the two experiments. However v_1 depends on u_1 and therefore the distributions of W_1 and W_2 differ slightly. What we claim though is that for any event $\mathcal{E}$ depending on walks of length τ,

$$|\,\mathbf{Pr}((u_1, v_1, W_1) \in \mathcal{E}) - \mathbf{Pr}((u_2, v_2, W_2) \in \mathcal{E})| \leq s(\tau). \tag{6}$$

This follows from the stronger claim that for any $u \in V$ and any event $\mathcal{E}$ depending on walks of length τ

$$|\,\mathbf{Pr}((u_1, v_1, W_1) \in \mathcal{E} \mid u_1 = u) - \mathbf{Pr}((u_2, v_2, W_2) \in \mathcal{E} \mid u_2 = u)| \leq s(\tau),$$

which follows from (5).

The notation $B(m, p)$ stands for the binomial random variable with parameters m = number of trials, and p = probability of success.

3 Static routing with bounded congestion

In this section we present an algorithm for static routing with bounded congestion. We first use a flow algorithm to randomize the endpoints. We then connect each pair of (new) endpoints by a random path. At this point, most of the edges have a limited congestion but some edges are overloaded. We then remove all paths that use overloaded edges. With high probability the number of disconnected pairs is sufficiently small that we can use an algorithm for finding edge-disjoint paths [6] to reconnect them. More formally our algorithm is:

Algorithm

Input: An r-regular β-expander $G = (V, E)$. A collection of $K = \alpha(n)n/\log n$ pairs of vertices denoted $\{(a_1, b_1), \ldots, (a_K, b_K)\}$ such that no vertex in V participates in more than s pairs.

Output: A set of K paths, $\{P_1, \ldots, P_K\}$ such that P_i connects a_i to b_i and the maximum congestion g on any edge is bounded by

$$g = \begin{cases} O\left(s + \left\lceil \dfrac{\log\log n}{\log(1/\hat{\alpha})} \right\rceil\right), & \text{for } \alpha < 1/2; \\[2ex] O(s + \alpha + \log\log n), & \text{for } \alpha \geq 1/2. \end{cases}$$

where $\hat{\alpha} = \min(\alpha, 1/\log\log n)$.

Phase 1. Choose independently (with replacement) uniformly at random, two multisets R_A and R_B of $2K$ vertices each in V.

Phase 2. Select multisets $Q_A \subset R_A$ and $Q_B \subset R_B$ of K vertices, such that every element in Q_A has multiplicity at most $\max(10eK/n, 1)$ and every element in Q_B has multiplicity at most $\max(10eK/n, 1)$. If such sets cannot be found, then **stop**. The algorithm has failed.

Phase 3. Let $S_A = \{a_1, \ldots, a_K\}$ and $S_B = \{b_1 \ldots, b_K\}$. Using a flow algorithm in G twice, connect in an arbitrary manner the vertices of S_A (resp. S_B for the second flow) to the vertices of Q_A (resp. Q_B) by K paths as follows:

- Assume that every edge in G has a capacity equal to $\max(s, 20eK/n, 2)/\beta$.
- View each vertex in S_A (resp. S_B) as a source with capacity equal to its multiplicity in S_A (resp. S_B) and similarly every vertex in Q_A (resp. Q_B) as a sink with capacity equal to its multiplicity in Q_A (resp. Q_B) .

The expansion properties of G ensure that such flows always exist.

Phase 4. Let $\tilde{a}_i$ (resp. $\tilde{b}_i$) denote the vertex in Q_A (resp. Q_B) that was connected to the original end-point a_i (resp. b_i). Choose $x_1, x_2, \ldots, x_K$ uniformly at random in V and then choose trajectories W_i (resp. W_i') of length $\tau = c_0 \log n$ that go from $\tilde{a}_i$ to x_i (resp. $\tilde{b}_i$ to x_i) according to the distribution on trajectories, conditioned on $w_{i,0} = \tilde{a}_i$ and $w_{i,\tau} = x_i$. (The constant c_0 is discussed in the analysis.)

Let $\nu(e)$ be the number of trajectories that use the edge e. Let $g_{\max} = g_{\max}(\alpha)$ be a parameter defined in the analysis below. For every edge e with $\nu(e) > g_{\max}$ delete all the trajectories using it. For all i such that both W_i and W_i' survived, connect $\tilde{a}_i$ to $\tilde{b}_i$ using W_i followed by W_i' with loops removed. (This will lead to congestion at most $g_{\max}$.)

If the numbers of disconnected pairs is "too large" (see analysis), then **stop**. The algorithm has failed. Otherwise reconnect them using the algorithm in [6] on G.

The final path from a_i to b_i is the union of the paths from a_i to $\tilde{a}_i$, and from b_i to $\tilde{b}_i$ found in Phase 3, and the path from $\tilde{a}_i$ to $\tilde{b}_i$ selected here, with loops removed.

End Algorithm

We will show that the algorithm above succeeds with probability greater than $1/3$ for any given input. Thus by repeating it $O(\log n)$ times we prove

Theorem 1 *There is an explicit polynomial time algorithm that with high probability can connect any set of $K = \alpha(n)n/\log n$ pairs of vertices on a bounded degree expander so that no edge is used by more than g paths where*

$$g = \begin{cases} O\left(s + \left\lceil \dfrac{\log\log n}{\log(1/\hat{\alpha})} \right\rceil\right), & \text{for } \alpha < 1/2; \\ O(s + \alpha + \log\log n), & \text{for } \alpha \geq 1/2, \end{cases}$$

$\hat{\alpha} = \min(\alpha, 1/\log\log n)$, *and* s *is the maximal multiplicity of a vertex in the set of pairs.*

Proof: We first discuss the existence of Q_A, Q_B. Note first that the multiplicity of an element in R_A or R_B has distribution $B(2K, 1/n)$.

Let X_v denote the multiplicity of $v \in V$ in the multi-set R_A. Let

$$Z_t = \sum_{v \in V} X_v 1_{X_v \geq t}$$

where

$$t = \max\left\{2, \frac{10eK}{n}\right\}.$$

Then

$$\begin{aligned}
\mathbf{E}(Z_t) &\leq \sum_{v \in V} \sum_{s \geq t} \binom{2K}{s} n^{-s} \\
&\leq \sum_{v \in V} \sum_{s \geq t} \left(\frac{2eK}{sn}\right)^s \\
&\leq \frac{5n}{4} \left(\frac{2eK}{tn}\right)^t.
\end{aligned}$$

Case 1 $K \geq n/5e$.

$$\mathbf{E}(Z_t) \leq n/20 \leq 3K/4.$$

Changing one of our $2K$ choices for R_A changes Z_t by at most t and so applying the Azuma-Hoeffding martingale tail inequality (see Alon and Spencer [3], Chapter 6) we see that for any $u > 0$,

$$\mathbf{Pr}(Z_t \geq \mathbf{E}(Z_t) + u) \leq \exp\left\{-\frac{2u^2}{2Kt^2}\right\}. \tag{7}$$

Putting $u = K/4$ we see that the RHS of (7) is o(1) provided $K = o(n^2)$. (For larger K, the result is trivial, since then the Chernoff bounds imply that each X_v is sharply concentrated around its mean $2K/n$.)

Let S_0 be the set of vertices of multiplicity at least $10eK/n$ in R_A. The total multiplicity of S_0 is **whp** at most K and so $V \setminus S_0$ contains a multi-set Q_A as required. Similarly for Q_B.

Case 2 $K < n/5e$.

$$\mathbf{E}(Z_t) \leq \frac{5n}{4}\left(\frac{eK}{n}\right)^2 = \left(\frac{5e^2K}{4n}\right) K \leq \frac{eK}{4}.$$

For $K \geq n^{3/4}$ we can use (7) to show concentration of Z_t around its mean. For $K \leq n^{3/4}$ we can use the fact that $\mathbf{E}(Z_t) = o(K)$ and apply the Markov inequality to show that **whp** $Z_t \leq K$. After this we proceed as in Case 1.

We now continue with the analysis of Phase 3. A straightforward application of the Max-Flow Min-Cut Theorem shows that the flow phase always suceeds in finding paths between the vertices in S_A (resp. S_B) to the vertices in Q_A (resp. Q_B). Furthermore, the flow is computed in polynomial time.

Assume for a moment that we start random walks of length $\tau = c_0 \log n$ from every element of R_A, without any conditioning on their other endpoint. Let $\nu'(e)$ be the number of walks that use the edge e in this case. Then $\nu'(e) \leq B(2K, 2\tau/(rn))$ in distribution. (Each walk starts at an independently chosen vertex and moves to an independently chosen destination. The steady state of a random walk on G is uniform and so at each stage of a walk, each edge of G is equally likely to be crossed.) Thus for any $t > 0$

$$\mathbf{Pr}(\nu'(e) = t) \leq \binom{2K}{t} \left(\frac{2\tau}{rn}\right)^t \leq \left(\frac{4Ke\tau}{trn}\right)^t = \left(\frac{4e\alpha(n)c_0}{tr}\right)^t.$$

Now let $\nu_1(e)$ (resp. $\nu_2(e)$) be the number of trajectories W_i (resp. W_i') in phase 4 that use e. We choose c_0 such that the separation between the distribution of the endpoint of a walk of length τ and the uniform distribution is less than (say) $1/(n^3K^2)$. Then in view of the paragraph above, we have

$$\mathbf{Pr}(\nu_1(e) = t) \leq \left(\frac{4e\alpha(n)c_0}{tr}\right)^t + \frac{1}{n^3K^2}, \tag{8}$$

where the error term comes from the fact that the endpoint of each trajectory is chosen uniformly at random rather than according to the distribution of the endpoint of the walk. Note also that we have to consider ν_1 and ν_2 separately because we want the endpoints to be chosen independently.

The results of [6] imply that given any n vertex, bounded degree, regular β-expander G and given any set of $q \leq n/(\log n)^\kappa$ disjoint pairs of vertices in G, it is possible to find with high probability a set of paths in G connecting the q pairs, such that each edge in G participates in no more than $1 + 1/\beta$ paths. The parameter κ depends only on the expansion properties of G. Fix it to be the value corresponding to the input graph here. We now consider two cases:

Case 1: $\alpha \leq 1/2$
Define

$$t_0 = \left\lceil 4e^2 c_0 + \frac{3\kappa \log\log n}{\log(1/\hat{\alpha})} \right\rceil$$
$$g_{\max} = 2t_0$$

Observe that

$$\sum_{t=t_0}^{\infty} t \left(\frac{4e\alpha c_0}{tr}\right)^t \leq 2t_0 \left(\frac{4e\alpha c_0}{t_0 r}\right)^{t_0} \leq \left(\frac{4e^2\alpha c_0}{t_0 r}\right)^{t_0} \leq \frac{1}{4r}\left(\frac{\alpha}{t_0}\right)^{t_0/2} \leq \frac{1}{4r(\log n)^\kappa}. \tag{9}$$

Suppose that we delete all those trajectories W_i or W_i' which use an edge with $\max\{\nu_1(e), \nu_2(e)\} \geq t_0$. The number of pairs of vertices which become disconnected is at most

$$\Delta = \sum_{t_0 \leq t \leq K} t|\{e : \nu_1(e) = t\}| + \sum_{t_0 \leq t \leq K} t|\{e : \nu_2(e) = t\}|.$$

It follows from (8) and (9) (there are $rn/2$ edges) that $\mathbf{E}(\Delta) \leq n/(4(\log n)^\kappa) + 1/n$ and so with probability at most 1/2 we find that $\Delta \leq n/(\log n)^\kappa$. We now re-link these pairs using the algorithm of [6] at an additional congestion cost of $1 + 1/\beta$. Thus the total congestion on an edge is at most

$$\frac{\max(s, 20eK/n, 2)}{\beta} + 2t_0 + \left(1 + \frac{1}{\beta}\right) = O\left(s + \frac{\log\log n}{\log(1/\hat{\alpha})}\right).$$

Case 2: $\alpha \geq 1/2$. This time we take

$$t_0 = \left\lceil 4\alpha e^2 c_0 + 3\kappa \log\log n \right\rceil = O(\alpha + \log\log n),$$
$$g_{\max} = 2t_0$$

and proceed as before. The congestion now is $O(s + \alpha + \log\log n)$. □

4 Edge Disjoint Paths

The main ideas are similar to those described in the previous section, except that we need to partiton the edge set E into $2p$ disjoint subsets $E_1, E_2, \ldots, E_{2p}$, for a suitable p. Let $G_i = (V, E_i)$. We try to connect the K_i pairs which are left unconnected from the $i-1$'th routing phase using the edges $E_{2i-1} \cup E_{2i}$. It is likely that K_p is sufficently small that all the K_p pairs can be connected using the algorithm of [6].

The value of p and the required expansion properties are given next.

$$p = 1 + \lceil \log_2 \epsilon^{-1} + 10 \log_2 r \rceil \tag{10}$$

$$\beta \geq \frac{2p}{\psi(\epsilon)} \gamma^{-1} H(\gamma) \tag{11}$$

$$\beta > \frac{2p}{1-\epsilon} \tag{12}$$

$$\alpha < \frac{1}{8p} \tag{13}$$

where

$$H(\gamma) = ((1-\gamma)^{1-\gamma}\gamma^\gamma)^{-1}$$
$$\psi(\epsilon) = (1-\epsilon)\ln(1-\epsilon) + \epsilon$$

The above conditions look complex but they will hold for the Ramanujuan graphs of Lubotsky, Phillips and Sarnak [14] and for random regular graphs when r is large. For the former, it follows from Lemma 2.3 of Alon and Chung [2] that

$$|X| = \delta n \text{ implies } \mathrm{out}(X) \geq r(1-\lambda)(1-\delta)|X|,$$

where λ is the second largest eigenvalue of the transition probability matrix associated with the random walk on G. If G is one of the Ramanujan graphs then $\lambda = 2\sqrt{r-1}/r$ and if G is a large random r-regular graph then $\lambda \approx 2/\sqrt{r}$ (see Friedman, Kahn, and Szemerédi [8]). We see then that in these cases (10) to (13) hold with $\alpha = 1/(100 \log r)$, $\beta = r/3$ and $\gamma = 1/(200 \log r)$.

Algorithm

Input: An r-regular (α, β, γ)-expander $G = (V, E)$ for which (10) – (13) hold. A collection of $K = n/(\log n)^{2+\epsilon}$ pairs of vertices denoted $\{(a_1, b_1), \ldots, (a_K, b_K)\}$ such that no vertex in V participates in more than one pair.

Output: A set of K paths, $\{P_1, \ldots, P_K\}$ such that P_i connects a_i to b_i.

Phase 0. Using Algorithm **Partition** below, divide E into $E_1 \cup E_2 \cup \cdots E_{2p}$ (p defined in (10))

Let $K_1 = K$, $S_{A,1} = \{a_1, a_2, \ldots, a_K\}$ and $S_{B,1} = \{b_1, b_2, \ldots, b_K\}$.

For $i = 1$ **to** $p-1$ **do**

Phase i.1. Choose independently (with replacement) two multisets $R_{A,i}$ and $R_{B,i}$ of $2K_i$ vertices each in V. These vertices are individually chosen according to the steady state distribution $\mathcal{D}_i$ of random walks on G_{2i} i.e. proportional to degrees in this graph.

Phase i.2. Select *sets* $Q_{A,i} \subset R_{A,i}$ and $Q_{B,i} \subset R_{B,i}$ of K_i vertices. If such sets cannot be found, then **stop**. The algorithm has failed.

Phase i.3. Let $S_{A,i} = \{a_{1,i}, \ldots, a_{K_i,i}\}$ and $S_{B,i} = \{b_{1,i} \ldots, b_{K_i,i}\}$. Using a flow algorithm in G_{2i-1} twice, connect in an arbitrary manner the vertices of $S_{A,i}$ (resp. $S_{B,i}$ for the second flow) to the vertices of $Q_{A,i}$ (resp. $Q_{B,i}$) by K_i paths as follows:

- Assume that every edge in G_{2i-1} has a capacity equal to 1.
- View each vertex in $S_{A,i}$ (resp. $S_{B,i}$) as a source with capacity 1 and similarly every vertex in $Q_{A,i}$ (resp. $Q_{B,i}$) as a sink with capacity equal 1.

The expansion properties of G_{2i-1} ensure that such flows always exist.

Phase i.4. Let $\tilde{a}_{k,i}$ (resp. $\tilde{b}_{k,i}$) denote the vertex in $Q_{A,i}$ (resp. $Q_{B,i}$) that was connected to the original end-point $a_{k,i}$ (resp. $b_{k,i}$). Choose $x_{1,i}, x_{2,i}, \ldots, x_{K_i,i}$ independently with distribution $\mathcal{D}_i$ in V and then choose trajectories $W_{k,i}$ (resp. $W'_{k,i}$) of length $\tau_i = c_i \log n$ that go from $\tilde{a}_{k,i}$ to $x_{k,i}$ (resp. $\tilde{b}_{k,i}$ to $x_{k,i}$) according to the distribution on trajectories, conditioned on $w_{k,i,0} = \tilde{a}_{k,i}$ and $w_{k,i,\tau} = x_{k,i}$. (The constant c_i is discussed in the analysis.)

Let $\nu_i(e)$ be the number of trajectories that use the edge e. For every edge e with $\nu_i(e) \geq 2$ delete all the trajectories using it. For all k such that both $W_{k,i}$ and $W'_{k,i}$ survived, connect $\tilde{a}_{k,i}$ to $\tilde{b}_{k,i}$ using $W_{k,i}$ followed by $W'_{k,i}$ with loops removed. For such survivors, the final path from $a_{k,i}$ to $b_{k,i}$ is the union of the paths from $a_{k,i}$ to $\tilde{a}_{k,i}$, and from $b_{k,i}$ to $\tilde{b}_{k,i}$ found in Phase i.3, and the path from $\tilde{a}_{k,i}$ to $\tilde{b}_{k,i}$ selected here, with loops removed.

If the number of disconnected pairs K_{i+1} is "too large" (see analysis), then **stop**. The algorithm has failed.

End i loop

Phase 5. Use the algorithm of [6] to connect the final K_p pairs via the edges of the graphs G_{2p-1}, G_{2p}.

End Algorithm

We will show that the algorithm above succeeds with probability greater than $1/3$ for any given input. Thus by repeating it $O(\log n)$ times we prove Theorem 2.

We now describe our algorithm for partitioning E. It is a simple generalisation of Algorithm Split of [6]. A desription is necessary in order to check that each G_i has sufficient expansion.

Algorithm Partition

Input: An r-regular (α, β, γ)-expander $G = (V, E)$ for which (10) – (13) hold.

Output: $2p$ spanning subgraphs $G_i = (V, E_i), 1 \leq i \leq 2p$ such that $E_1, E_2, \ldots, E_{2p}$ is a partition of E and each G_i is a θ-expander where

$$\theta = \min\{(\lfloor r/(4p) \rfloor - \alpha r), (1-\epsilon)\beta/(2p)\}.$$

1. Orient the edges of G so that $|outdegree(v) - indegree(v)| \leq 1$ for all $v \in V$.
2. For each $v \in V$ randomly partition the edges directed out of v into $2p$ sets $X_{v,1}, \ldots, X_{v,2p}$ each of size $\lfloor r/4p \rfloor$ or $\lceil r/4p \rceil$. Let $E_i = \bigcup_{v \in V} X_{v,i}$, for $1 \leq i \leq 2p$.

Note that the degrees $d_{v,i}$ of vertices $v \in V$ in G_i satisfy

$$\left\lfloor \tfrac{r}{4p} \right\rfloor \le d_{v,i} \le \frac{r}{2}.$$

If $\mathrm{out}_i(S)$ denotes the number of G_i edges leaving a set S then (see Case 1 of Theorem 4.1 of [6]) for $|S| \le \gamma n$,

$$\begin{aligned}\lfloor r/(4p) \rfloor |S| &\le 2\,\mathrm{in}(S) + \mathrm{out}_i(S)\\ r|S| &\ge 2\,\mathrm{in}(S) + (1-\alpha)r|S|\end{aligned}$$

yielding

$$\mathrm{out}_i(S) \ge (\lfloor r/(4p) \rfloor - \alpha r)|S|.$$

Now consider $|S| > \gamma n$. By following the proof of Case 2 of Theorem 4.1 ($\gamma n < |S| \le n/2$) and replacing $2s$ by $r/2$ and $k/2$ by $k/(2p)$ we see that provided (11) holds then **whp**

$$\mathrm{out}_i(S) \ge (1-\epsilon)\beta|S|/(2p).$$

Thus Partition succeeds **whp**.

Theorem 5 *Suppose $\epsilon > 0$ and G is an r-regular (α, β, γ)-expander. If conditions (11),(12) and (13) hold and r is sufficiently large then any $K = n/(\log n)^{2+\epsilon}$ pairs of vertices can be connected by edge-disjoint paths. Furthermore these paths can be explicitly found in polynomial time.*

Proof: Assume Partition succeeds. Now assume inductively, that conditional on the success of Partition, with probability at least $1-(i-1)/(2p)$ we start Phase $i, i \ge 1$ with

$$K_i \le \left(\frac{Cr^3p^2}{\theta^4}\right)^{2^{i-1}-1} \frac{n}{(\log n)^{2+2^{i-1}\epsilon}} \tag{14}$$

unsatisfied pairs, for some absolute constant $C > 0$.

This is trivially true for $i = 1$.

First consider the existence of $Q_{A,i}$. We note that if $\mathcal{D}_i = (\pi_{v,i},\, v \in V)$ then for large r

$$\frac{1}{3n} \le \pi_{v,i} \le \frac{2p}{n}. \tag{15}$$

Let $S_{A,i}$ be the set of vertices of multiplicity 1 in $R_{A,i}$. Then

$$\begin{aligned}\mathbf{E}(|S_{A,i}|) &= \sum_{v \in V} 2K_i \pi_{v,i}(1-\pi_{v,i})^{2K_i-1}\\ &\ge 2K_i \sum_{v \in V} \pi_{v,i}\left(1 - \frac{4pK_i}{n}\right) \qquad \text{using (15)}\\ &= 2K_i\left(1 - \frac{4pK_i}{n}\right).\end{aligned}$$

Changing one choice in $R_{A,i}$ changes $|S_{A,i}|$ by at most 1 and so the Azuma-Hoeffding inequality implies that $|S_{A,i}|$ is sharply concentrated around $(2-o(1))K_i$ and $Q_{A,i}$ exists **whp**. Similarly for $Q_{B,i}$.

A simple application of the Max-Flow Min-Cut Theorem shows that the flow part Phase $i.3$ will succeed. We use the fact that the expansion of G_{2i-1} is at least 1, assuming $\theta > 1$, which is true when (12), (13) hold.

Now consider the random walks Phase $i.4$. Let λ_i denote the second largest eigenvalue of the transition probability matrix of the random walk on G_{2i}. We choose

$$c_i = \frac{3}{\log \lambda_i^{-1}}. \tag{16}$$

This implies that the separation (4) between the distribution of the endpoints $x_{k,i}$ of our walks and their steady state is at most $1/(nK^3)$. For $e \in E_{2i}$ let $\nu_i(e) = \nu_{i,1}(e) + \nu_{i,2}(e)$ where $\nu_{i,1}(e)$ (resp. $\nu_{i,2}(e)$) is the number of trajectories $W_{k,i}$ (resp. $W'_{k,i}$) which use e. Arguing as in Section 3 we see that for $e \in E_{2i}$

$$\mathbf{Pr}(\nu_{i,1}(e) \geq t) \leq \binom{2K_i}{t}\left(\frac{c_i \log n}{|E_{2i}|}\right)^t + \frac{1}{nK}.$$

Then

$$\begin{aligned}
\mathbf{E}(K_{i+1}) &\leq 2|E_{2i}| \sum_{t \geq 2} t \binom{2K_i}{t}\left(\frac{c_i \log n}{|E_{2i}|}\right)^t + \frac{2}{n} \\
&\leq \frac{9K_i^2 c_i^2 (\log n)^2}{2|E_{2i}|} + \frac{2}{n} \\
&\leq \frac{20K_i^2 c_i^2 p (\log n)^2}{rn}.
\end{aligned}$$

So with probability at least $1 - 1/(2p)$

$$K_{i+1} \leq \frac{40K_i^2 c_i p^2 (\log n)^2}{rn}. \tag{17}$$

The *conductance* – see [16] – of the walk is at least θ/r. It follows (as in (3)) that

$$\lambda_i \leq 1 - \frac{\theta^2}{2r^2}.$$

It then follows from (16) that

$$c_i \leq \frac{6r^2}{\theta^2}.$$

So from (17) we obtain

$$K_{i+1} \leq \frac{CK_i^2 r^3 p^2 (\log n)^2}{\theta^4 n}$$

for some absolute constant $C > 0$.

Going back to (14) completes the induction. Thus with conditional probability at least $1/2$ there are at most

$$\left(\frac{Cr^3p^2}{\theta^4}\right)^{2^{p-1}-1} \frac{n}{(\log n)^{2+2^{p-1}\epsilon}}$$

unmatched pairs going into the final phase. Let $\kappa = 2 + 2^{p-1}\epsilon$. We must check that this is sufficiently large for the algorithm of [6]. The conditions (13) – (15) of [6] are satisfied when

$$\kappa > \frac{13(\log r)^2}{(\log \lambda_p^{-1})^2}.$$

Thus it suffices to ensure that

$$2 + 2^{p-1}\epsilon > 52r^4(\log r)^2/\theta^4.$$

Going back to our definition of p in (10) we see that the above holds for large r and this completes the proof of Theorem 5.

5 Dynamic Selection of Paths

We define a stochastic model for studying a dynamic version of the circuit switching problem. In our model new requests for establishing paths arrive continuously at nodes according to a discrete Poisson process. Requests wait in the processor's queue until the requested path is established. The duration of a path is exponentially distributed.

Our model is characterized by three parameters:

- P_1 is an upper bound on the probability that a new request arrives at a given node at a given step.
- P_2 is the probability that a given existing path is terminated in a given step. A path *lives* from the time it is established until it is terminated.
- g is the maximum congestion allowed on any edge.

We assume that the destinations of path requests are chosen uniformly at random among all the graph vertices.

Our goal is to characterize the relationship between these parameters that ensures stability of the system. (By stability, we mean that the expected length of any queue does not grow unboundly in time). We also estimate the expected delay incurred by a request in the steady state distribution.

We study a simple and fully distributed algorithm for this problem. In our algorithm each processor at each step becomes active with probability $P_1' > P_1$ (P_1'

is defined later). An inactive processor does not try to establish a path even if there are requests in its queue. One effect of this is to moderate the injection rate for vertices with large queues. Also, just for the analysis, we imagine that if a process is active but its queue is empty, then it tries to construct a *ghost path* to a random vertex. The lifetime of a ghost path is distributed as that of real paths.

Our algorithm is simply this: Assume that a is active at step t, and the first request in a's queue is for b. Processor a tries to establish a path to b by choosing a random trajectory of length $\tau = c_0 \log n$ connecting a to b. If the path does not use any edge with congestion greater than $g-1$, the path is established, otherwise the request stays in the queue.

Theorem 6 *Let*
$$\Phi = \min\left\{\frac{1}{\log(grn)}, \frac{rg}{\tau^{(g+1)/g}}\right\}.$$
There exists a constant γ such that if $P_1 \le \gamma\Phi P_2$, then the system is stable and the expected wait of a request in the queue is $O(1/P_1)$.

Before giving the proof let us see the consequence of this theorem. Let $\mathbf{E}(N) = nP_1$ be the expected number of new requests that arrive at the system at a given step, and let $\mathbf{E}(D) = 1/P_2$ be the expected duration of a connection. For the system to be stable, the expected number of simultaneously active paths in the steady state must be at least $\mathbf{E}(N)\,\mathbf{E}(D) = nP_1/P_2$. Plugging $g = \log\log n/\log\omega$ for some ω in the range $[1, \log n]$ in the definition of Φ we get
$$\Phi = \Omega\left(\frac{1}{\omega \log n}\right).$$
Thus the theorem above implies that for such a congestion g, the system remains stable even if we choose P_1 and P_2 such that
$$\mathbf{E}(N)\,\mathbf{E}(D) = n\frac{P_1}{P_2} = \gamma n \Phi = \Omega\left(\frac{n}{w \log n}\right),$$
in which case the dynamic algorithm utilizes the edges of the network almost as efficiently as the static algorithm analyzed in section 3 (there seems to be an efficiency gap of maximum order $\log\log\log n$ for $\omega \le \log\log n$).

Proof of the theorem: Partition time into intervals of length T (to be determined). Let H_t denote the history of the system during the first t time intervals. Define the event
$$\mathcal{E}(v,t) = \left\{\begin{matrix}\text{If the queue of processor } v \text{ was not empty at the beginning of} \\ \text{interval } t \text{ then } v \text{ served at least one request during interval } t\end{matrix}\right\}$$
Our goal is to show that for all v and t,
$$\mathbf{Pr}(\mathcal{E}(v,t) \mid H_{t-2}) > \frac{1}{2}.$$

To this end consider the following four events:

- $\mathcal{E}_1$ defined as

$$\mathcal{E}_1(v,t) = \{\text{Processor } v \text{ was not active in any step of interval } t. \qquad \}$$

Then

$$\mathbf{Pr}(\mathcal{E}_1(v,t) \mid H_{t-2}) \leq (1-P_1')^T \leq e^{-TP_1'} \leq \frac{1}{10}, \tag{18}$$

provided that

$$TP_1' \geq \log 10. \tag{19}$$

- $\mathcal{E}_2$ defined as

$$\mathcal{E}_2(t) = \left\{ \begin{array}{l} \text{At the beginning of interval } t \text{ there is a path in the network} \\ \text{that is still alive although it was established before the be-} \\ \text{ginning of interval } t-1. \end{array} \right\}$$

Clearly, at the start of time interval $t-1$ there are at most grn live paths. So

$$\mathbf{Pr}(\mathcal{E}_2(t) \mid H_{t-2}) \leq grn(1-P_2)^T \leq grne^{-P_2T} \leq \frac{1}{10}, \tag{20}$$

provided that

$$TP_2 \geq \log(10grn). \tag{21}$$

- $\mathcal{E}_3$ defined as

$$\mathcal{E}_3(v,t) = \left\{ \begin{array}{l} \text{There are more than } 2nP_1'/P_2 \text{ live paths that were created} \\ \text{within intervals } t-1 \text{ and } t \text{ when } v \text{ makes its first attempt in} \\ \text{interval } t \end{array} \right\}$$

To evaluate the probability of this event, we overestimate the number of paths in the network when v makes its first attempt at establishing a path in time interval t. We include in the count ghost paths, and paths or ghost path attempts which could not be established because of congestion. The life of such failed paths will also be geometric with parameter P_2. In this count we exclude paths attempts from before the start of period $t-1$. Suppose that period $t-2$ finished j_0 time steps ago. Then our estimate is

$$X = \sum_{w=1}^{n} \sum_{j=0}^{j_0} X_{w,j}$$

where $X_{w,j}$ is the 0/1 indicator variable for the event

$$\left\{ \begin{array}{l} j \text{ time steps ago, a path attempt was made from } w \text{ and this} \\ \text{path is still alive.} \end{array} \right\}$$

Then $\mathbf{Pr}(X_{w,j} = 1) = P_1'(1-P_2)^j$ and so

$$E(X \mid H_{t-2}) = \sum_{w=1}^{n} \sum_{j=0}^{j_0} P_1'(1-P_2)^j \leq n\frac{P_1'}{P_2}.$$

An easy argument shows the concentration of X and therefore

$$\mathbf{Pr}(\mathcal{E}_3(v,t) \mid H_{t-2}) = o(1). \tag{22}$$

- $\mathcal{E}_4$ defined as

$$\mathcal{E}_4(v,t) = \left\{\begin{array}{l}\text{The first path that processor } v \text{ tries in interval } t \text{ includes an} \\ \text{edge used by at least } g \text{ other paths that were created within} \\ \text{intervals } t-1 \text{ and } t.\end{array}\right\}$$

We break the ensuing analysis into 2 cases:

Case 1: $2\tau\left(\frac{4e\tau}{rgP_2}\right)^g > 1/10$.

Then

$$\begin{aligned}&\mathbf{Pr}(\mathcal{E}_4(v,t) \mid H_{t-2}, X \leq 2nP_1'/P_2) \\ &\qquad \leq \tau \sum_{h \geq g} \binom{2n\frac{P_1'}{P_2}}{h} \left(\frac{2\tau}{rn} + \frac{1}{n^3K^2}\right)^h \leq 2\tau\left(\frac{4e\tau P_1'}{rgP_2}\right)^g \leq \frac{1}{10},\end{aligned} \tag{23}$$

provided that

$$P_1' \leq \frac{rgP_2}{4e\tau}\left(\frac{1}{20\tau}\right)^{1/g}. \tag{24}$$

In the above calculation we have implicitly used the fact that the X path attempts are a collection of X random walks between randomly chosen pairs. We include unsuccessful path attempts in order to avoid conditioning problems caused by paths blocking paths. Since

$$\bar{\mathcal{E}}(v,t) \subseteq \mathcal{E}_1(v,t) \cup \mathcal{E}_2(t) \cup \mathcal{E}_3(v,t) \cup \mathcal{E}_4(v,t)$$

we see that

$$\mathbf{Pr}(\bar{\mathcal{E}}(v,t) \mid H_{t-2}) \leq \frac{3}{10} + o(1),$$

and we conclude that that in any segment of $2T$ steps processor v is serving at least one request with probability at least $1/2$. The number of new arrivals in this time interval has a Binomial distribution with expectation at most $2TP_1 < 1/2$, provided that

$$TP_1 \leq \frac{1}{4}. \tag{25}$$

Thus, under these conditions the queue is dominated by an $M/M/1$ queue with expected inter-arrival distribution greater than $4T$, and expected service time smaller than $4T$. The queue is stable, and the expected wait in the queue is $O(1/T) = O(1/P_1)$.

It remains to check that if γ is small then we can choose T and $P_1 < P_1' < 1$ to satisfy (19), (21), (24), (25). We take

$$T = \frac{1}{4\gamma P_2 \Phi}$$
$$P_1' = \min\left\{\frac{rgP_2}{1000\tau^{(g+1)/g}}, 1\right\}.$$

This completes the proof for Case 1.

Case 2: $2\tau\left(\frac{4e\tau}{rgP_2}\right)^g \leq 1/10$.

Here we take T as above and $P_1' = 1$. Here (23) holds by the case hypothesis. $\square$

6 Existential results regarding edge-disjoint paths

In this section we show how to use the Lovász Local Lemma [7] to prove the *existence* of a large number of edge disjoint paths in any r-regular (α, β, γ)-expanders. We do not see how to make the argument constructive. More precisely we prove

Theorem 3 *Given a bounded degree (α, β, γ)-expander graph there exists a parameter c that depends on α, β, γ, but not on n, such that any set of less than $cn/(\log n)^2$ disjoint pairs of vertices can be connected by edge disjoint paths.*

Proof: Let
$$K = \left\lfloor \frac{rn}{577\tau^2} \right\rfloor,$$
where
$$\tau = \lceil c_2 \log n \rceil$$
and c_2 is a constant discussed below. Let $(a_1, b_1), (a_2, b_2), \ldots, (a_K, b_K)$ be any set of K disjoint pairs of vertices. We claim that G contains edge disjoint paths joining a_i to b_i for $i = 1, 2, \ldots, K$.

Our proof follows the blueprint used in [6]. We start by splitting the original graph $G = (V, E)$ into two disjoint β'-expanders $G_R = (V, E_R)$ and $G_B = (V, E_B)$ exactly as was done in [6]. The salient facts here are: (a) $\beta' > 1$; (b) the construction succeeds with probability $1 - o(1)$, thus such a split always exists; and (c) the maximum degree in G_B is at most $3r/4$ and the minimum degree is at least $r/4$..

The disjoint paths are constructed in two stages. In the first stage we choose a random set $Z = \{\zeta_1, \zeta_2, \ldots, \zeta_{2K}\}$ of $2K$ distinct vertices. We connect the original endpoints to the vertices of Z in an arbitrary fashion via edge-disjoint paths in G_R, such that each Z-vertex is the endpoint of exactly one path. A simple flow argument proves constructively the existence of such edge-disjoint paths on any graph with edge expansion larger than one.

Let $\tilde{a}_i$ (resp. $\tilde{b}_i$) denote the vertex in Z that was connected to the original end-point a_i (resp. b_i) in the first stage. The core of the proof is to show via the Lovász Local Lemma the existence of edge-disjoint paths in G_B connecting $\tilde{a}_i$ to $\tilde{b}_i$, for $i = 1, ..., K$. To this end we choose for each i a random path of length 2τ from $\tilde{a}_i$ to $\tilde{b}_i$. However the direct application of the Lovász Local Lemma is precluded by the fact that we do not have any control over how the pairing $(\tilde{a}_i, \tilde{b}_i)$ was done, and thus, although the probability of paths sharing an edge is small, the dependency graph is complete. To avoid this calamity, further randomization is necessary, as follows: assume that we rename the vertices of Z at random. Then the pairing induced by the flow becomes a random pairing.

More formally, let ρ be a random permutation of $[2K]$. Let

$$z_i = \zeta_{\rho(i)}, \qquad \text{for } 1 \leq i \leq 2K.$$

The flow algorithm defines a pairing $f : Z \to Z$, that is, a function f such that for all $\zeta \in Z$ we have $f(\zeta) \neq \zeta$ and and $f^2(\zeta) = \zeta$. This pairing is defined by $f(\tilde{a}_i) = \tilde{b}_i$ and $f(\tilde{b}_i) = \tilde{a}_i$. In turn, it induces a pairing ϕ on $[2K]$ via

$$f(\zeta_i) = \zeta_j \leftrightarrow f(z_{\rho^{-1}(i)}) = z_{\rho^{-1}(j)} \leftrightarrow \phi(\rho^{-1}(i)) = \rho^{-1}(j).$$

It is easy to verify that after f is fixed, if ρ is chosen uniformly at random among the permutations of $[2K]$ then ϕ is uniform over the set Φ of all possible pairings of $[2K]$.

Now suppose that for $1 \leq i \leq K$ we choose x_i with distribution

$$\mathbf{Pr}(x_i = v) = \pi_B(v) = \frac{d_B(v)}{2|E_B|}$$

(the steady state distribution of a random walk on G_B) and then choose W_i' and W_i'' randomly from all trajectories of length τ which go from $\tilde{a}_i$ to x_i and $\tilde{b}_i$ to x_i respectively. The distribution used for choosing W_i' is that of a random walk of length τ starting at $\tilde{a}_i$ conditional on ending at x_i. Let W_j denote the walk which starts at z_j (obtained by catenating a pair W_i', W_i'') for $1 \leq j \leq 2K$ and define the event

$$\mathcal{E}_{i,j} = \{W_i \cap W_j \neq \emptyset \text{ and } \phi(i) \neq j\}.$$

Our proof reduces to showing that the event

$$\mathcal{E} = \bigcap_{i<j} \bar{\mathcal{E}}_{i,j}$$

has positive probability.

Now define $\phi_0 \in \Phi$ to be the pairing $\{(1, K+1), (2, K+2), \dots, (K, 2K)\}$. Let $\mathbf{Pr}_0$ denote probabilities conditional on $\phi = \phi_0$. We will prove that

$$\mathbf{Pr}_0(\mathcal{E}) > 0. \tag{26}$$

Since $\mathbf{Pr}(\phi = \phi_0) > 0$ this suffices to complete the proof. Now if $\phi = \phi_0$ then

$$\mathcal{E} = \bigcap_{|j-i| \neq K} \bar{\mathcal{A}}_{i,j}$$

where

$$\mathcal{A}_{i,j} = \{W_i \cap W_j \neq \emptyset\}.$$

We will apply the local lemma to the events $\mathcal{A}_{i,j}$, conditional on $\phi = \phi_0$.

For each pair (i, j) with $|j - i| \neq K$ we let

$$S_{i,j} = \{\mathcal{A}_{i',j'} : i', j' \notin \{i, j, i+K, j+K\}\}.$$

Fix i, j and let $\mathcal{B}$ denote any event dependent only on the outcome of events in $S_{i,j}$. Then

$$\begin{aligned} \mathbf{Pr}_0(\mathcal{A}_{i,j} \mid \mathcal{B}) &= \sum_{x,y \in V} \mathbf{Pr}_0(W_i \cap W_j \neq \emptyset, z_i = x, z_j = y \mid \mathcal{B}) \\ &= \sum_{x,y \in V} \mathbf{Pr}_0(W_i \cap W_j \neq \emptyset, z_i = x, z_j = y) \qquad (27) \\ &= \mathbf{Pr}_0(\mathcal{A}_{i,j}). \qquad (28) \end{aligned}$$

Equation (27) follows from the fact that if we fix z_i and z_j then the occurrence of $\mathcal{A}_{i,j}$ does not depend on the paths $\{W_k : k \neq i, j\}$. Thus conditional on $\phi = \phi_0$, $\mathcal{A}_{i,j}$ is independent of the events in $S_{i,j}$. Thus the dependency graph has maximum degree d where

$$d \leq 4K. \tag{29}$$

This justifies the complexity of the previous analysis. If no care is taken, the dependence graph will be complete.

Still keeping i, j fixed, let $\Phi_{i,j} = \{\phi \in \Phi : \phi(i) \neq j\}$ and choose an arbitrary $\phi' \in \Phi_{i,j}$. Now

$$\begin{aligned} \mathbf{Pr}_0(\mathcal{A}_{i,j}) &= \sum_{x,y \in V} \mathbf{Pr}_0(\mathcal{A}_{i,j}, z_i = x, z_j = y) \\ &= \sum_{x,y \in V} \mathbf{Pr}(\mathcal{A}_{i,j} \mid z_i = x, z_j = y, \phi = \phi_0)\, \mathbf{Pr}(z_i = x, z_j = y \mid \phi = \phi_0) \\ &= \sum_{x,y \in V} \mathbf{Pr}(\mathcal{A}_{i,j} \mid z_i = x, z_j = y, \phi = \phi')\, \mathbf{Pr}(z_i = x, z_j = y \mid \phi = \phi') \qquad (30) \\ &= \mathbf{Pr}(\mathcal{A}_{i,j} \mid \phi = \phi'). \qquad (31) \end{aligned}$$

To justify (30) observe that

- $\mathbf{Pr}(\mathcal{A}_{i,j} \mid z_i = x, z_j = y, \phi = \phi_0) = \mathbf{Pr}(\mathcal{A}_{i,j} \mid z_i = x, z_j = y, \phi = \phi')$ since given $z_i = x$ and $z_j = y$ as long as $\phi(i) \neq j$, that is z_i is not paired with z_j, we can decide $\mathcal{A}_{i,j}$ without further reference to ϕ.

- $\mathbf{Pr}(z_i = x, z_j = y \mid \phi = \phi_0) = \mathbf{Pr}(z_i = x, z_j = y \mid \phi = \phi')$ since, conditioning on Z and on the pairing induced by the flow phase on Z, if $\{x, y\} \not\subseteq Z$ or x is paired with y, then both sides are 0; otherwise, the LHS is proportional to the number of permutations that induce ϕ_0 and make $z_i = x$ and $z_j = y$. This is clearly the same as the number of permutations that induce ϕ' and make $z_i = x$ and $z_j = y$.

It follows from (31) that

$$\begin{aligned}\mathbf{Pr}_0(\mathcal{A}_{i,j}) &= \frac{1}{|\Phi_{i,j}|} \sum_{\phi' \in \Phi_{i,j}} \mathbf{Pr}(\mathcal{A}_{i,j} \mid \phi = \phi') \\ &= \frac{|\Phi|}{|\Phi_{i,j}|} \sum_{\phi' \in \Phi_{i,j}} \mathbf{Pr}(\mathcal{A}_{i,j}, \phi = \phi') = \mathbf{Pr}(\mathcal{A}_{i,j} \mid \phi(i) \neq j)\end{aligned}$$

and thus

$$\mathbf{Pr}_0(\mathcal{A}_{i,j}) \leq \frac{1}{\mathbf{Pr}(\phi(i) \neq j)} \sum_{x,y \in V} \mathbf{Pr}(\mathcal{A}_{i,j} \mid z_i = x, z_j = y) \, \mathbf{Pr}(z_i = x, z_j = y) \tag{32}$$

Since ϕ is a random pairing,

$$\mathbf{Pr}(\phi(i) \neq j) = \frac{2K-2}{2K-1}. \tag{33}$$

The sum in (32) is the probability of the following event $\mathcal{M}$: Choose x, y uniformly at random (without replacement). Choose x', y' with probability π_B. Do random walks W, W' of length τ from x to x' and y to y'. The event is $\{W \cap W' \neq \emptyset\}$.

We now prove that

$$\mathbf{Pr}(\mathcal{M}) \leq \frac{(36 + o(1))\tau^2}{rn}. \tag{34}$$

Let $P^{(\tau)}_{v,\cdot}$ be the distribution of a random walk on G_B starting from v after τ steps. Consider the event $\mathcal{M}'$ which differs from $\mathcal{M}$ only in that x and y are chosen independently with distribution π_B, and x' and y' are chosen with the distribution $P^{(\tau)}_{x,\cdot}$ and $P^{(\tau)}_{y,\cdot}$ respectively. (We can now have $x = y$, but this has probability $O(1/n)$ and we will deal with it later.) Let $W = e_1, e_2, \ldots, e_\tau$ and $W' = f_1, f_2, \ldots, f_\tau$ as edge sequences. Then for $1 \leq \ell, m \leq \tau$ we have

$$\mathbf{Pr}(f_\ell = e_m) = \frac{1}{|E(G_B)|} \leq \frac{4}{rn}.$$

since the edges of each random walk have the uniform distribution over $E(G_B)$ and the start points of these walks are chosen independently. Thus,

$$\mathbf{Pr}(\mathcal{M}') \leq \sum_{\ell=1}^{\tau} \sum_{m=1}^{\tau} \mathbf{Pr}(f_\ell = e_m) \leq \frac{4\tau^2}{rn}.$$

Let now $\mathcal{M}''$ be the same as $\mathcal{M}'$ except that we choose x' and y' according to π_B. We take c_2 such that the separation $s(\tau)$ is less than $1/n$, and therefore

$$\mathbf{Pr}(\mathcal{M}'') \leq \mathbf{Pr}(\mathcal{M}') + \frac{1}{n}.$$

Finally, given the bounds on the degrees occurring in G_B, when choosing a pair x, y according to π_B we find that no pair occurs with probability more than 9 times any other pair. Thus

$$\mathbf{Pr}(\mathcal{M}) \leq 9\,\mathbf{Pr}(\mathcal{M}'' \mid x \neq y) \leq 9\,\mathbf{Pr}(\mathcal{M}'')/\,\mathbf{Pr}(x \neq y)$$

and (34) follows.

We see from (32), (33) and (34) that

$$\mathbf{Pr}_0(\mathcal{A}_{i,j}) \leq \frac{(36 + o(1))\tau^2}{rn}.$$

Using this and (29) in the local lemma yields the theorem. □

References

[1] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[2] N. Alon and F. R. K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72:15–19, 1989.

[3] N. Alon and J. Spencer The probabilistic method

[4] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive on-line routing. In *34th Annual Symposium on Foundations of Computer Science*, pages 32–40, Palo Alto, California, 3–5 Nov. 1993. IEEE.

[5] B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line admission control and circuit routing for high performance computing and communication. In *35th Annual Symposium on Foundations of Computer Science*, pages 412–423, Santa Fe, New Mexico, 20–22 Nov. 1994. IEEE.

[6] A. Z. Broder, A. M. Frieze, and E. Upfal. Existence and construction of edge-disjoint paths on expander graphs. *SIAM J. Comput.*, 23(5):976–989, Oct. 1994.

[7] P. Erdös and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal et al., editors, *Infinite and Finite Sets*, volume 11 of *Colloq. Math. Soc. J. Bolyai*, pages 609–627. 1975.

[8] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue in random regular graphs. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 587–598, 1989.

[9] A. Kamath, O. Palmon, and S. Plotkin. Routing and admission control in general topology networks with Poisson arrivals. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 269–278, Atlanta, Georgia, 28–30 Jan. 1996.

[10] J. Kleinberg and R. Rubinfeld. Short paths in expander graphs. In *37th Annual Symposium on Foundations of Computer Science*, Burlington, Vermont, 14–16 Oct. 1996. IEEE.

[11] A. Lubotsky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[12] F.P. Kelley. Blocking probabilities in large circuit-switching networks. *Adv. Appl. Prob.*, 18:573–505, 1986.

[13] F.T. Leighton and S. Rao. Circuit switching: a multi-commodity flow based approach. *Proc. of 9th International Parallel Processing Symposium*, 1995.

[14] A. Lubotsky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[15] N. Robertson and P. D. Seymour. Graph minors-XIII: The disjoint paths problem. *JCT series B*, 63:65-110, 1995.

[16] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, July 1989.