Sampling regular graphs and a peer-to-peer network

(Extended Abstract)

Colin Cooper*, Martin Dyer†and Catherine Greenhill‡

Abstract

We consider a simple Markov chain for d-regular graphs on n vertices, and show that the mixing time of this Markov chain is bounded above by a polynomial in n and d. A related Markov chain for d-regular graphs on a varying number of vertices is introduced, for even degree d. We use this to model a certain peer-to-peer network structure. We prove that the related chain has mixing time which is bounded by a polynomial in N, the expected number of vertices, under reasonable assumptions about the arrival and departure process.

1 Introduction

In this paper we prove rapid mixing of a simple and natural Markov chain for generating random regular graphs. We use this to model a peer-to-peer network implementation of Bourassa and Holt [2], which is based on random walks. Bourassa and Holt conjectured that their network quickly becomes a random regular graph, and consequently has good properties with high probability. Their claim was supported by computer simulation. We extend our Markov chain analysis to give theoretical justification of their conjecture.

1.1 Sampling regular graphs The problem of sampling graphs with a given degree sequence (on a fixed number of vertices) has been well studied, especially the case of sampling d-regular graphs on n vertices, where $d = d(n)$ may grow with n. Expected polynomial-time algorithms for uniform generation were described by Bollobás [1], Frieze [5], McKay and Wormald [12] and Steger and Wormald [15] (for degrees $O(\sqrt{\log n})$, $o(n^{1/5})$, $o(n^{1/3})$ and $o(n^{1/11})$, respectively). Simpler algorithms can be used if we are content to sample *approximately* uniformly. Tinhofer [16] describes one such but does not bound how far away the resulting probability distribution is from uniform. Jerrum and Sinclair [9] give a polynomial time approximately uniform generator with running time polynomial in n and $\log(\varepsilon^{-1})$, for any regular degree sequence $d = d(n) \leq n/2$ (and by complementation, higher degrees can be handled). Their algorithm uses a Markov chain which samples perfect and near-perfect matchings of a related graph, and can be extended to certain non-regular degree sequences (see [8]). Moreover, the analysis of Jerrum, Sinclair and Vigoda [10] shows that the approach of [9] gives a polynomial time approximately uniform generator for *bipartite* graphs with any given degree sequence (see [10, Section 6]).

Kannan, Tetali and Vempala [11] study the mixing of a simple Markov chain on regular graphs which "switches" a pair of nonadjacent edges at each step. They give a proof of rapid mixing for the bipartite case using canonical paths, but this does not appear complete in every respect, and it is certainly unclear how it extends to generating nonbipartite graphs. We give a proof for the general version which provides, as a special case, the result given in [11] for sampling bipartite graphs. Our proof uses different canonical paths for this problem, defined by Goldberg and Jerrum [6]. We then describe a Markov chain for d-regular graphs on a varying number of vertices, with d a fixed constant. This Markov chain is a model of a peer-to-peer network proposed in [2]. Using a multicommodity flow argument we prove that this Markov chain mixes in time bounded above by a polynomial in the expected number of vertices of the graph, under some assumptions about the rate of arrival and departure of vertices.

For future reference we note that the number of d-regular graphs on n vertices is well known to be asymptotically equal to

$$\sqrt{2}\, e^{-(d^2-1)/4} \left(\frac{d^{d/2}}{e^{d/2}\, d!} \right)^n n^{dn/2}. \tag{1.1}$$

See for example [17, Corollary 4.2] or [7, Corollary 9.8].

In this extended abstract we must, unfortunately, omit detailed proofs. These are given in a draft of the full paper, available as [3].

*ccooper@dcs.kcl.ac.uk, Department of Computer Science, Kings College, London WC2R 2LS, UK.

†dyer@comp.leeds.ac.uk, School of Computing, University of Leeds, Leeds LS2 9JT, UK.

‡csg@unsw.edu.au, School of Mathematics, UNSW, Sydney 2052, Australia.

1.2 Peer-to-peer networks A *peer-to-peer* (P2P) network is a decentralised, dynamic network for sharing data and computing resources among the vertices (participants) of the network. The participants must follow the protocol of the network, and if they do, then the network should maintain some desirable properties. The properties which the designers of protocols would like to ensure include connectivity, low degree and small diameter of the network. In addition, the protocol should ideally be simple, local and robust.

The low diameter of a changing network is usually maintained by imposing some randomness on the new connections. Pandurangan, Raghavan and Upfal [13] proposed a protocol which ensures that the network is connected and has logarithmic diameter with high probability, and has always bounded degree. A simpler, fully decentralised protocol was proposed by Bourassa and Holt [2], based on random walks: if a vertex v in the network needs an address of a random vertex, then v initiates a random walk and gets an address of the vertex reached at some specified step of the walk. We describe the exact mechanism by which a vertex joins or leaves the network in section 3 below. Bourassa and Holt conjecture that their protocol produces a random regular graph, so the properties of such graphs (low diameter, good connectivity) should hold. In section 3 we will show that this conjecture is true under reasonable assumptions, to be discussed below, about the dynamic behaviour of the network.

Cooper and Radzik [4] also propose a decentralised protocol based on random walks, which is robust under adversarial deletion of vertices and edges and actively reconnects itself.

1.3 Markov chain definitions Let $\mathcal{M}$ be an ergodic, time-reversible Markov chain on the finite state space Ω with transition matrix P and stationary distribution π. The *total variation distance* between two probability distributions σ, π on Ω is given by

$$d_{\mathrm{TV}}(\sigma,\pi) = \tfrac{1}{2}\sum_{x\in\Omega} |\sigma(x)-\pi(x)|.$$

The *mixing time* $\tau(\varepsilon)$ is defined by

$$\tau(\varepsilon) = \max_{x\in X} \min\left\{T \mid d_{\mathrm{TV}}(P_x^t,\pi) \le \varepsilon \text{ for all } t \ge T\right\},$$

where P_x^t is the distribution of the state X_t of the Markov chain after t steps from the initial state $X_0 = x$.

One way to bound the mixing time of Markov chains is via *multicommodity flows* (see [14]). Let $\mathcal{G}$ be the graph underlying the Markov chain $\mathcal{M}$, so that xy is an edge of $\mathcal{G}$ if and only if $P(x,y) > 0$. A *flow* in $\mathcal{G}$ is a function $f : \mathcal{P} \to \mathbb{R}^+$ which satisfies

$$\sum_{p\in\mathcal{P}_{xy}} f(p) = \pi(x)\pi(y) \quad \text{for all } x,y\in\Omega,\ x\ne y,$$

where $\mathcal{P}_{xy}$ is the set of all simple directed paths from x to y in $\mathcal{G}$ and $\mathcal{P} = \cup_{x\ne y}\mathcal{P}_{xy}$. Extend f to a function on oriented edges by setting $f(e) = \sum_{p\ni e} f(p)$, so that $f(e)$ is the total flow routed through e. Let $\ell(f)$ be the length of the longest path with $f(p) > 0$, and let

$$\rho(f) = \max_e \frac{f(e)}{Q(e)}$$

where $Q(e) = \pi(x)P(x,y)$ if $e = xy$. Then Sinclair [14, Proposition 1 and Corollary 6′] proves that

$$\tau(\varepsilon) \le \rho(f)\ell(f)\left(\log(1/\pi^*) + \log(\varepsilon^{-1})\right). \tag{1.2}$$

2 A Markov chain on regular graphs of fixed size

Let $\Omega_{n,d}$ be the set of all d-regular graphs $G = (V,E)$ with $V = \{1,\dots,n\}$. In this section we analyse the Markov chain $\mathcal{M}$ described in Figure 2. This is almost

```
From G ∈ Ω_{n,d} do
  with probability 1/2 do nothing; otherwise
    choose two nonadjacent distinct edges ij, kℓ, u.a.r.,
    choose a perfect matching M of {i,j,k,ℓ} u.a.r.,
    if M ∩ E(G) = ∅ then
      delete the edges ij, kℓ and add the edges of M,
    otherwise do nothing;
    end if;
  end;
end;
```

Figure 1: The Markov chain on $\Omega_{n,d}$

the same as the Markov chain studied by [11] in the bipartite case. However, in order to ensure the validity of (1.2), we have added a "holding probability" of $\frac{1}{2}$ at every move. A move of this Markov chain is called a *switch*. Note that this chain is symmetric: the number of unordered pairs of nonadjacent edges in a d-regular graph on n vertices is always

$$a_{n,d} = \binom{nd/2}{2} - n\binom{d}{2},$$

and hence $P(X,Y) = P(Y,X) = 1/(6a_{n,d})$ if X and Y differ just by a switch. The proof that $\mathcal{M}$ is irreducible will follow from the canonical paths defined

below. Since $P(X,X) > 0$ for each X, the chain is aperiodic. Therefore $\mathcal{M}$ is ergodic and time-reversible with uniform stationary distribution. We write $\mathcal{G}$ for the underlying graph of $\mathcal{M}$, so $\mathcal{G} = (\Omega_{n,d}, \Gamma)$ where each edge $e \in \Gamma$ corresponds to a transition of $\mathcal{M}$.

Now given $G, G' \in \Omega_{n,d}$, let $H = G \triangle G'$ be the symmetric difference of G and G'. We will refer to the edge set E_B of $G \setminus G'$ as *blue* and the edge set E_R of $G' \setminus G$ as *red*. Thus $H = (V, E_H)$, where $E_H = E_B \cup E_R$. The first step in defining the flow is to decompose H in a canonical way. For bipartite graphs, the symmetric difference can be written as the union of edge-disjoint alternating cycles. However, we may not be able to achieve this in the nonbipartite case. For example,

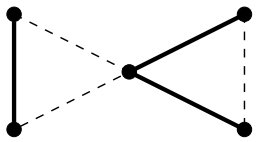

Figure 2: The bowtie graph.

if $d \geq 2$ then it is possible to find d-regular graphs G, G' with symmetric difference given by the "bowtie" graph of Figure 2. We must be able to deal with such symmetric differences.

2.1 Defining the flow We begin by decomposing H into a sequence $\mathcal{C}$ of *circuits*. A circuit $C = w_0 w_1 \cdots w_k$ is a string over the alphabet V such that $w_{i-1}w_i \in E_H$ $(i = 1, \ldots, k)$, $w_k w_0 \in E_H$, and all *edges* are distinct (note that we do not require the *vertices* to be distinct). Let $H = G \triangle G'$ be the above symmetric difference, with (equal) red and blue vertex degrees ρ_v $(v \in V)$. We choose a *pairing* of the red and blue edges around each vertex. Let $\Psi(G, G')$ be the set of all such pairings, so $|\Psi(G, G')| = \prod_{v \in V} \rho_v!$. For each pairing in $\Psi(G, G')$, we construct a canonical path from G to G' in $\mathcal{G}$. Each of these paths will carry $1/|\Psi(G, G')|$ of the total flow from G to G'.

Fix a pairing $\psi \in \Psi(G, G')$. Let the lexicographically least edge in E_H be $w_0 w_1$ $(w_0 < w_1)$. If $w_0 w_1$ is blue (resp. red), choose the red (resp. blue) edge $w_1 w_2$ paired with $w_0 w_1$ at w_1. Now choose the blue (resp. red) edge $w_2 w_3$ paired with $w_1 w_2$ at w_2. In general choose the red (resp. blue) edge $w_i w_{i+1}$ paired with the blue (resp. red) edge $w_{i-1} w_i$ at w_i. Note that the w_i are not necessarily distinct, but the edges are distinct. This terminates with a circuit $C_1 = w_0 w_1 \ldots w_{k-1} w_k$ when $w_k w_0$ is paired with $w_0 w_1$. Then, if $E_H = C_1$, $\mathcal{C} = (C_1)$. Otherwise take the lexicographically least edge not in C_1 and generate a new circuit C_2 by the above procedure. Continue generating circuits until $E_H = C_1 \cup C_2 \cup \cdots \cup C_s$. Then $\mathcal{C} = (C_1, C_2, \ldots, C_s)$ and the circuits $C_1, C_2, \ldots, C_s$ are edge-disjoint. Note that each circuit C_i is an *alternating* circuit in H: as we traverse the edges they are alternately red and blue. This implies that each circuit has even length. Note also that, once the pairing has been chosen, $\mathcal{C}$ is formed without regard to the colouring of H. This property will be used later.

The canonical path will involve swapping the red and blue edges of each circuit in the order of $\mathcal{C}$. We will show how to do this for a particular circuit $C = w_0 w_1 \ldots w_{k-1} w_k$. To this end we make the following definition. A *1-circuit* $S = v_0 v_1 v_2 \cdots v_t$ will be a substring of C such that

1. S is an alternating circuit in H,
2. $v_0 = w_0$, and w_0 appears only once in S.

We will show how to decompose a circuit into a sequence of 1-circuits. We *process* each 1-circuit in order, meaning we switch its red and blue edges. We show later how this is done. We make *temporary* changes to the pairing ψ at some vertices during the construction and processing of the 1-circuits. When this is done at vertex v, we say v is (temporarily) *rewired*. Note that rewiring does not change the symmetric difference or the pairing, just the decomposition which is produced.

Let $v = w_0$, and suppose $C^{(0)} = C$ is initially the unprocessed section of C. Then

$$C^{(0)} = v x_1 \cdots y_1 v x_2 \cdots y_2 \cdots v x_{\rho-1} \cdots y_{\rho-1} v x_\rho \cdots y_\rho,$$

where $\rho = \rho_v$. If $v x_1$ and $y_1 v$ have different colours, $S = v x_1 \cdots y_1$ is a 1-circuit. We process it to give $C^{(1)} = v x_2 \cdots y_2 \cdots v x_{\rho-1} y_{\rho-1} v \cdots v x_\rho \cdots y_\rho$ as the unprocessed section of C, which we process inductively. If $v x_\rho$ and $y_\rho v$ have different colours, $S = v x_\rho \cdots y_\rho$ is a 1-circuit. We process it to give $C^{(1)} = v x_1 \cdots y_1 v x_2 \cdots y_2 v \cdots v x_{\rho-1} \cdots y_{\rho-1}$ to be processed inductively. Otherwise, $v x_1$, $y_1 v$ have the same colour, and $v x_\rho$, $y_\rho v$ have the other colour. We process $S' = v x_\rho \cdots y_\rho v x_1 \cdots y_1$, to give $C^{(1)} = v x_2 \cdots y_2 v \cdots v x_{\rho-1} \cdots y_{\rho-1}$, to be processed inductively. Let us call S' a *2-circuit*. However, it is not a 1-circuit, and we must show how to process it. We remark that the bipartite case is somewhat easier, since this cannot occur.

We now show how to process a 2-circuit given that we can process a 1-circuit. Suppose that we have reached the graph Z on the canonical path from G to G' in $\mathcal{G}$. Let us assume that the first edge of S' is in Z, if necessary by reversing S', by which we mean following the pairings in the opposite direction. Let

$$S' = v w x \cdots u v y \cdots z,$$

and note that u, w, y, z are distinct neighbours of v. (See Figure 3.) Therefore x is distinct from at least one

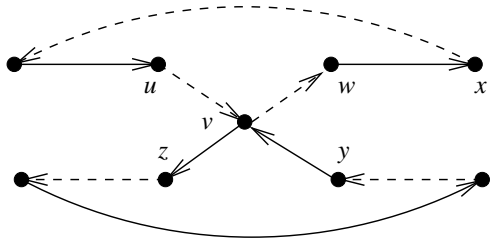

Figure 3: A 2-circuit

of y, z. It is also distinct from v, w since they lie on successive edges. If x is only distinct from z, we rewire v so that

$$S' = vwx \cdots uvz \cdots y,$$

so let us assume it is distinct from y. Consider xy. There are three cases.

Case (a): xy (or yx) is an edge of S'.

In this case, we can rewire either x or y to split the 2-circuit S' into two 1-circuits, in a canonical way. We order the 2-circuits and process them in order. The details are omitted here.

Case (b): xy is not an edge of S' and either (b1) xy and wx are both edges of Z, or (b2) xy, wx are both non-edges in Z.

By rewiring, we divert the path $yvwx$ through the edge yx. This gives the 1-circuit $vz \cdots yx \cdots u$, which we process. Finally we switch the 4-cycle $yvwx$. Note that the edge xy is restored to its original state by the end of this process (in case (b1) it is deleted and then reinstated, while in case (b2) it is added and then deleted). Call xy the *shortcut*.

Case (c): xy is not an edge of S' and exactly one of xy, wx is an edge of Z.

Similar to case (b). First switch the 4-cycle $yvwx$, and then use xy to give the 1-circuit $vz \cdots yx \cdots u$, which we then process. Finally we rewire to divert through the path $yvwx$ rather than the edge (or non-edge) yx.

We now form a path $G = Z_0, Z_1, \ldots, Z_M = G'$ from G to G' in the underlying graph of the Markov chain (that is, to get from Z_a to Z_{a+1} we perform a switch), by processing each circuit C_i in turn. Suppose that we have processed the circuits $C_1, \ldots, C_{r-1}$, to get the (partial) path $G = Z_0, Z_1, \ldots, Z_R$. To process C_r, we use the decomposition into 1-circuits described above and process these in order. It remains to describe how we process a 1-circuit. If necessary, we reverse S to ensure that the first edge is blue. Let $S = x_0 x_1 \ldots x_{2k-1}$ be any 1-circuit of C_r, so x_0 is the first vertex of C_r and $x_{2t}x_{2t+1} \in Z_R$ $(0 \le t \le k-1)$ are the blue edges. There will be p phases in the processing of 1-circuit S, where

$$p = |\{t : 1 \le t \le k-1, x_0 x_{2t+1} \notin Z_R\}|.$$

For the first phase, find the lowest t such that $x_0 x_{2t+1} \notin Z_R$. Form Z_{R+1} from Z_R by deleting the edges x_0x_{2t-1}, $x_{2t}x_{2t+1}$ and replacing them by the edges x_0x_{2t+1}, $x_{2t-1}x_{2t}$. Note that either $t = k-1$ or x_{2t+1} now has no Z_{R+1}-neighbours among x_{2t}, x_{2t+2}. Similarly, either $t = 1$ or x_{2t-1} has both x_{2t}, x_{2t-2} as Z_{R+1}-neighbours. If $t > 1$ then there are $t-1$ further steps in phase 1. For $j := t-1, t-2, \ldots, 1$ do: form $Z_{R+t-j+1}$ from Z_{R+t-j} by deleting the edges x_0x_{2j-1}, $x_{2j}x_{2j+1}$ and replacing them with the edges $x_0x_{2j+1}, x_{2j-1}x_{2j}$. Any chord $x_0x_{2\ell-1}$ which was in Z_R but was deleted when $j = \ell$, will be reinstated when $j = \ell - 1$. The only chord which is introduced during this phase which was not present in Z_R is the chord x_0x_{2t+1}.

This completes phase 1, and extends the path by $Z_{R+1}, \ldots, Z_{R+t}$. The edges of Z_{R+t} on C_{j_r} match those of G' on the segment $x_0x_1 \cdots x_{2t}x_{2t+1}$. Also, a chord x_0x_{2j+1} is present in Z_{R+t} if and only if it is present in Z_R, for $1 \le j \le t-1$. If $t = k-1$ then there is only one phase, so the processing of W is complete. Otherwise, there are further phases and x_{2t+1} still has no Z_{R+t}-neighbours in x_{2t}, x_{2t+2}. However, since the phase is complete, no vertex is joined to both its predecessor and successor around S in Z_{R+t}.

The subsequent phases are similar. For phase 2, look for the first index $q > t$ such that $x_0x_{2q+1} \notin Z_{R+t}$. Perform a step by switching x_0x_{2q-1}, $x_{2q}x_{2q+1}$ for edges x_0x_{2q+1}, $x_{2q-1}x_{2q}$. Then work backwards to fix up all segments of the 1-circuit between $j = q-1$ and $j = t+1$ (if any). There are $q - t$ steps in phase 2. Continue until all phases are complete. After processing this 1-circuit, the edges of S in the final graph Z_T agree with G', and the only odd chords present are exactly those which were present at the start. Moreover, we have not added or deleted any other edges. Note that reinstating the chords correctly is an important feature of the method, since 1-circuits can intersect arbitrarily. If we did not do this, we could not guarantee to process the decomposition "circuit by circuit" in the manner we have described.

This completes the description of how to form the canonical path from G to G', by processing each circuit in the canonical decomposition (in order) as described above. Each switch performed is a proper move of the Markov chain since x_0 is distinct from all other vertices in the 1-circuit, and two consecutive edges in the circuit span three distinct vertices.

2.2 Analysing the flow Fix a pairing $\psi \in \Psi(G, G')$ and let Z be any graph on the corresponding canonical path from G to G'. Identify each graph with its symmetric $n \times n$ adjacency matrix. Define a symmetric $n \times n$ matrix Q by $Q + Z = G + G'$. Entries of Q belong to $\{-1, 0, 1, 2\}$. Alternatively, we can think of Q as K_n with edge labels given by the entries in Q. An edge in Q is called *bad* if its label is -1 or 2. Note that Q is independent of ψ. We call Q an *encoding* for Z (with respect to G, G'). Note that an edge receives label -1 if it is absent in both G and G' but it is present in Z, while an edge receives label 2 if it is present in both G and G' but it is not present in Z. Thus, edges in the symmetric difference $G \triangle G'$ never receive bad labels.

Let Z' be the next graph after Z in the canonical path from G to G'.

LEMMA 2.1. *Given (Z, Z'), Q and ψ, we can uniquely recover G and G'.*

The proof of the following is similar to one in [11] (but is omitted).

LEMMA 2.2. *There are at most four bad edges in any encoding Q.*

We can extend the notion of a switch to encodings. Let $Q(ab)$ denote the label of edge ab in the encoding Q. By definition, the sum of edge-labels around any vertex v in an encoding Q adds up to d. If x, y, z, w are vertices with $Q(xy) > -1$, $Q(zw) > -1$, $Q(xz) < 2$ and $Q(yw) < 2$ then we may perform a switch by decreasing $Q(xy)$ and $Q(zw)$ by one and increasing $Q(xz)$ and $Q(yw)$ by 1, to give a new encoding Q'. (This is the analogue of deleting edges xy, zw and inserting edges xz, yw.)

The proof of the next lemma is adapted from Goldberg and Jerrum [6] and is omitted.

LEMMA 2.3. *One can obtain a graph from a valid encoding, using at most 3 switches.*

LEMMA 2.4. *The number of encodings for a transition (Z, Z') is at most $2d^8n^3|\Omega_{n,d}|$.*

For each pair (G, G') of distinct graphs in $\Omega_{n,d}$, let $\mathcal{P}_{G,G'}$ be the set of $|\Psi(G, G')|$ canonical paths which we have defined from G to G', one for each pairing $\psi \in \Psi(G, G')$. Let $\mathcal{P} = \cup_{G \neq G'} \mathcal{P}_{G,G'}$. Define

$$f(\gamma) = |\Omega_{n,d}|^{-2} \, |\Psi(G, G')|^{-1}$$

for each path $\gamma \in \mathcal{P}_{G,G'}$. Then

$$\sum_{\gamma \in \mathcal{P}_{G,G'}} f(\gamma) = |\Omega_{n,d}|^{-2} = \pi(G)\pi(G')$$

where π is the stationary probability of the Markov chain, which is uniform on $\Omega_{n,d}$. Thus $f : \mathcal{P} \to \mathbb{R}^+$ is a flow. We want to apply (1.2). Now for any transition $e = (Z, Z')$ we have

$$1/Q(e) = |\Omega_{n,d}|/P(Z, Z') = 6a_{n,d} \, |\Omega_{n,d}| \leq d^2 \, n^2 |\Omega_{n,d}|.$$

Also $\ell(f) \leq dn/2$, since each transition along a canonical path replaces an edge of G by an edge of G'. Since π is uniform we have

$$\log 1/\pi^* = \log |\Omega_{n,d}| \leq dn \log(dn)$$

(see for example [17]). Plugging all this into (1.2) gives

$$\tau(\varepsilon) \leq \tfrac{1}{2} d^4 n^4 \, |\Omega_{n,d}| \, \log(dn\varepsilon^{-1}) \, \left(\max_e f(e) \right).$$

It remains to find an upper bound on $f(e)$ for each transition $e = (Z, Z')$.

From Lemma 2.1 we know that we can recover G and G' given Q, ψ and e. From Q and Z, we can recover the uncoloured H, and we will colour its edges *green* (in Z) and *yellow* (not in Z). Vertex v has green degree ρ_v in H, and equal yellow degree. Moreover, we know that the original red-blue colouring of H is largely consistent with the yellow-green colouring in the sense that the original red-blue pairing ψ will also be a yellow-green pairing. This follows from the way whole circuits are switched, and is *exactly* true for pairs which do not lie in the current circuit C_r. However it is *not* exactly true for C_r. Here there may be some green-green or yellow-yellow pairs in ψ. Let us call these *bad pairs*, and a vertex at which a bad pair occurs a *bad vertex*. It is not difficult to prove that a polynomial bound for the mixing time cannot be achieved unless there are only $O(\log n)$ bad vertices.

However, the care we have taken allows us to limit b to a small constant. We consider two types of bad vertex, *type 1* at which two bad pairs are allowed, one green-green and one yellow-yellow, and *type 2* at which only one bad pair is allowed. Thus a type 2 bad vertex has two more edges of one colour than the other. We know that there can be at most two bad pairs at w_0, the start vertex of C_r, separating the processed from the unprocessed sections. Thus there can be at most one bad vertex of type 1. This results from our order of processing the 1-circuits and 2-circuits. In addition, there may be at most three bad vertices of type 2 arising from the processing of a 1-circuit. Moreover, these vertices lie at the end of potentially bad edges, so we have already accounted for their choices in Lemma 2.4. Each type 1 vertex gives a factor less than d^2, and each type 2 vertex a factor less than d, so the number of consistent red-blue pairs is at most $d^5|\Psi(G, G')|$.

Let $\Xi(e)$ denote the pairs (G, G') which route through $e \in \Gamma$, and let $\Psi(H)$ be the set of red-blue pairings for $H = G \triangle G'$. Given e and an encoding Q, let $\Psi'(Q)$ be the set of red-blue colourings consistent with the yellow-green colouring of H. Denote by $\mathcal{Q}$ the set of all valid encodings. Combining the above, we can bound the total flow through e by

$$\begin{aligned} |\Omega_{n,d}|^2 f(e) &= \sum_{\Xi,\Psi} 1/|\Psi(H)| \\ &\leq \sum_{Q,\Psi'} 1/|\Psi(H)| \\ &\leq \sum_{Q} d^5 |\Psi(H)|/|\Psi(H)| \\ &= d^5 |\mathcal{Q}| \\ &\leq 2\, d^{13}\, n^3\, |\Omega_{n,d}|. \end{aligned}$$

Hence it follows from (2.2) that

$$\tau(\epsilon) \leq d^{17}\, n^7\, \log(dn\varepsilon^{-1}).$$

This bound is polynomial time, but probably very far from the truth. One might guess that $O(n \log n)$ steps would suffice, for constant d, but this seems beyond the reach of known proof techniques.

3 A chain on regular graphs with random size

We consider a model of the SWAN process [2] for peer-to-peer networking, which we view as a Markov chain on $\Omega = \cup_{n \geq d+1} \Omega_{n,d}$, for some even constant d. For simplicity, we will examine the case $d = 4$ in detail and discuss later how this generalises.

The process described in [2] is a way of maintaining a self-organised network which is a random 4-regular graph. At any time, a new client may arrive and request to join the network. This happens by taking two non-adjacent edges of the network and "clothespinning" them, replacing them by four new edges which join the new client to each of the four endpoints of the two chosen edges. Similarly, clients may choose to leave at any time, at which point this clothespinning procedure must be (randomly) reversed. This is described in more detail below.

We will assume that the inter-arrival times are independently exponentially distributed. While there are n clients in the system, the expected inter-arrival time is ν_n, and that the residual service times for clients currently in the system are identically and independently exponentially distributed with mean μ_n. We will further assume that $n\nu_n/\mu_n$ is a strictly increasing function of n. Thus we are assuming that a larger system does not *too strongly* encourage arrivals or discourage departures compared with a smaller system. Then, at any event (either an arrival or a departure), the probability that it is an arrival (or *insertion*) is

$$p_n = \frac{1/\nu_n}{1/\nu_n + n/\mu_n} = \frac{\mu_n}{\mu_n + n\nu_n},$$

and $q_n = 1 - p_n$ is the probability that it is a departure (or *deletion*). Note that, by our assumption, p_n is a strictly decreasing function of n. Also, assuming $\nu_0 > 0$, we have $p_0 = 1$. Let us suppose $p(n) = p_n$ is a twice-differentiable function of n, and N is such that $p(N) = \frac{1}{2}$. Note that if no such N exists, the network will grow indefinitely, and there is no equilibrium behaviour. We will write $p(n) = f(n/N)$, and assume that $f''(x)$ is uniformly bounded by an absolute constant on $x \geq 0$. Then we have

$$f(0) = 1, \quad f(1) = 1/2, \quad f'(1) = -\alpha < 0,$$

for some positive constant α.

An example which fits this model is constant Poisson arrivals with rate $1/\nu$, and constant departure rate $1/\mu$. If there are n clients in the system, then $p_n = \mu/(\mu + n\nu)$. Then $N = \mu/\nu$, and $p_n = \frac{1}{1+n/N} = f(n/N)$, with $f(x) = 1/(1+x)$, which satisfies our conditions. This model for arrivals and departures was used by Pandurangan, Raghavan and Upfal [13] in an analysis of a different architecture for peer-to-peer networking.

For ease of notation, let us write Ω_n instead of $\Omega_{n,4}$. We now describe how this processed can be viewed as a Markov chain on the state space $\Omega = \cup_{n \geq 5} \Omega_n$. Since Ω_n is the set of all 4-regular graphs with vertex set $\{1, \ldots, n\}$, our "conceptual" chain must do some relabelling to make sure that the vertex set of the graph always has this form. Of course, the process does not do this relabelling in practice. The Markov chain algorithm is shown in Figure 3. Note that we have added a holding probability $\frac{1}{2}$ to ensure the applicability of (1.2).

Think of the sets Ω_n as "levels", with Ω_{n+1} being the level above Ω_n. We want the stationary distribution to be uniform on labelled 4-regular graphs when conditioned on being in a certain level. So that the process has a chance of mixing, we must show that the process does not grow too far away from having the mean number of vertices in time polynomial in N.

Let π be the stationary distribution of the Markov chain and let $\sigma_n = \pi(\Omega_n)$. If π is to be uniform on the levels we must have

$$\pi(X) = \frac{\sigma_n}{|\Omega_n|} \text{ for all } X \in \Omega_n.$$

Suppose that $Y \in \Omega_{n+1}$ and that the Markov chain can move from X to Y in one step. Then

$$P(X, Y) = \frac{p_n}{a_n\,(n+1)}, \quad P(Y, X) = \frac{q_{n+1}}{3\,(n+1)},$$

From $G \in \Omega_{n,4}$ do
 with probability $\frac{1}{2}$ do nothing; otherwise
  with probability p_n do
   choose two distinct non-adjacent edges u.a.r.,
   add vertex $n+1$ on these edges,
   choose $i \in \{1, \ldots, n+1\}$ u.a.r.,
   swap the labels of vertex i and vertex $n+1$,
  else (with probability q_n) do
   choose $i \in \{1, \ldots, n+1\}$ u.a.r.,
   let a, b, c, d be the neighbours of vertex i,
   choose a perfect matching M of $\{a, b, c, d\}$, u.a.r,
   if $M \cap E(G) = \emptyset$ then
    swap the labels of vertex i and vertex $n+1$,
    delete vertex $n+1$ and add M to the edge set,
   else do nothing;
   end if;
  end;
 end;

Figure 4: The Markov chain on Ω

where $a_n = a_{n,4} = \binom{2n}{2} - n\binom{4}{2} = 2n^2 - 7n$ is the number of unordered pairs of non-adjacent edges in a 4-regular graph on n vertices. Detailed balance requires that $\pi(X)P(X,Y) = \pi(Y)P(Y,X)$, or in other words,

$$\frac{\sigma_n\, p_n}{|\Omega_n|\, a_n} = \frac{\sigma_{n+1}\, q_{n+1}}{3\,|\Omega_{n+1}|}.$$

From (1.1) we find that

$$\frac{|\Omega_{n+1}|}{|\Omega_n|} \sim \frac{a_n}{3}\Big(1 + O\Big(\frac{1}{n}\Big)\Big).$$

Therefore we set

$$\sigma_{n+1} = \frac{\sigma_n\, p_n}{q_{n+1}}\Big(1 + O\Big(\frac{1}{n}\Big)\Big) \tag{3.3}$$

so detailed balance guarantees that the Markov chain has stationary distribution π.

The proof of the following uses Taylor's expansion and properties of f (but is omitted).

LEMMA 3.1. *The equilibrium distribution of the size n of the system is approximately normal with mean N and variance $N/4\alpha$.*

It will be convenient to consider only levels n where $|N - n| \le 2N^{3/4}$. Define

$$\Omega^* = \bigcup_{|N-n| \le 2N^{3/4}} \Omega_n$$

and call this set the *centre* of the state space, and elements of Ω^* are called *central* states. The next result says that states are exponentially unlikely to be outside the centre of the state space, at least after a "burn-in" period.

LEMMA 3.2. *The following statements fail with probability exponentially small in N. If started from empty, the system reaches size $N \pm 2N^{3/4}$ in time $O(N^{5/4})$. Thereafter the system size does not leave this range in any time polynomial in N.*

We will analyse the mixing time of our Markov chain *conditioned on* its trajectory remaining within the central states. By Lemma 3.2, the difference between the conditional and unconditional probabilities is an exponentially small relative error, which we ignore. Hence we will write σ_n, π for the conditional probabilities as well, by a slight abuse of notation. The analysis of the Markov chain proceeds by defining a flow between all pairs of states in Ω^*. This flow builds upon the flow we defined on each level $\Omega_{n,d}$ in section 2. Note that we can simulate a switch move within level n, using an insertion immediately followed by a deletion. The transition which inserts vertex $n+1$ on edges ab, cd is used to simulate a switch move. When we delete we can replace the original edges ab, cd by either ac, bd or ad, bc. Simulating switch moves in this way increases the load on each transition by a factor of at most 2. In fact we will do all our "horizontal" moves, which simulate switch moves, using levels Ω_N and Ω_{N+1}. We write N rather than $\lfloor N \rfloor$ since we assume that N is large and hence the error is small.

3.1 Defining the flow Let $X \in \Omega_n$, $Y \in \Omega_m$ be elements of Ω^*. We define a family $\mathcal{P}_{X,Y}$ of canonical paths from X to Y implicitly, as follows: first define a family of paths taking flow from X to a set of graphs $S_X \subseteq \Omega_N$, then define a family of paths taking flow from Y to another set of graphs $S_Y \in \Omega_N$, and finally using the "horizontal moves" to define a flow from t_X to t_Y, for all $(t_X, t_Y) \in S_X \times S_Y$. Once we have defined the set of paths $\mathcal{P}_{X,Y}$ from X to Y, we will set $f(\gamma) = \pi(X)\,\pi(Y)/|\mathcal{P}_{X,Y}|$ for all $\gamma \in \mathcal{P}_{X,Y}$. This ensures that f is a flow. It remains to show how to define the canonical paths from X to some set S_X of graphs in level N, for each $X \in \Omega^*$.

Suppose first that $X \in \Omega_n$ where $n < N$. There are a_n neighbours of X in Ω_{n+1} which are obtained by a transition of the Markov chain involving a null relabelling: that is, the vertex label i chosen is $i = n+1$, so the "relabelling" has no effect. The set of canonical paths from X up to level N is the set of all paths $X = Z_0, Z_1, \ldots, Z_{N-n}$ where Z_1 is one of these a_n

neighbours of X in Ω_{n+1}, and $Z_1, \ldots, Z_{N-n}$ is one of the canonical paths from Z_1 up to level N. This inductively defines a family of paths from X up to level N.

Next suppose that $X \in \Omega_n$ where $n > N$. We wish to make a similar inductive definition of a family of paths from X down to level N. Let d_X be the number of neighbours of X in Ω_{n-1} which are obtained by a transition of the Markov chain involving a null relabelling (again, this means that the label i chosen for the relabelling is equal to $n+1$, so that the "relabelling" has no effect). Then $1 \le d_X \le 3$ and most states in Ω_n have exactly 3 such neighbours in Ω_{n-1}. Note $d_X \ge 1$ since we know $n+1$ is deletable because it was inserted in Ω_{n-1}. So the set of canonical paths from X down to level N is the set of all paths $X = Z_0, Z_1, \ldots, Z_{n-N}$ where Z_1 is one of these d_X neighbours of X in Ω_{n-1} and $Z_1, \ldots, Z_{n-N}$ is one of the canonical paths from Z_1 down to level N.

3.2 Analysing the flow We wish to apply (1.2). Note that $\ell(f) \le 4N + 4N^{2/3}$ since each path from a central state to another central state involves at most $4N^{2/3}$ vertical transitions and at most $4N$ horizontal transitions. If π^* is the smallest stationary distribution, conditioned over all central states, then it follows that $\log(1/\pi^*) = O(N \log N)$ It remains to find

$$\rho(f) = \max_e \frac{f(e)}{Q(e)}.$$

Now if $e = (Z, Z')$ is a transition from Ω_n to Ω_{n+1} then

$$Q(e) = \frac{\sigma_n \, p_n}{|\Omega_n| \, a_n \, (n+1)}.$$

The amount of flow which originates at X is $\pi(X) = \sigma_n/|\Omega_n|$, the stationary probability of X. We distribute the flow evenly between the a_n neighbours of X in Ω_{n+1}. Hence a neighbour $Y \in \Omega_{n+1}$ of X receives $\sigma_n/(a_n|\Omega_n|)$ flow from X, and the total amount of flow it receives from its neighbours in Ω_n is at most

$$\frac{3\sigma_n}{a_n|\Omega_n|} = \frac{\sigma_{n+1}q_{n+1}}{p_n|\Omega_{n+1}|} < \frac{\sigma_{n+1}}{|\Omega_{n+1}|}.$$

This follows from detailed balance and the fact that $p_n + p_{n+1} = f(n/N) + f((n+1)/N) \ge 1$ for $n < N$, since f is decreasing and $f(1) = 1/2$. Thus the total flow through X originating in all central states below level n is at most $2N^{3/4}\sigma_n/|\Omega_n|$. The loading on any edge e from Ω_n to Ω_{n+1} is therefore at most

$$\frac{f(e)}{Q(e)} \le \frac{1}{a_n} \cdot \frac{2N^{3/4}\sigma_n}{|\Omega_n|} \cdot \frac{|\Omega_n| \, a_n \, (n+1)}{\sigma_n \, p_n} < 5N^{7/4}, \tag{3.4}$$

since $n \sim N$ and $p_n \sim \frac{1}{2}$ for central states. Moreover, the total flow arriving at a state X in level N is at most $2N^{3/4}\sigma_N/|\Omega_N|$.

Similarly consider states $X \in \Omega_m$ where $N < m < N + 2N^{3/4}$. We route flow $\sigma_m/|\Omega_m|$ out of X and send it down to level N. First we distribute it equally among the neighbours of X in level $m-1$. Let d_X be the number of neighbours that X has in level $m-1$. Then $1 \le d_X \le 3$ and most states in Ω_m have exactly 3 neighbours in Ω_{m-1}. The total amount of flow received by a vertex $W \in \Omega_{m-1}$ from level m is

$$\begin{aligned} & \frac{\sigma_m}{|\Omega_m|} \sum_{X \in \Omega_m \cap N(W)} \frac{1}{d_X} \\ \sim \; & \frac{a_{m-1}\sigma_m}{3|\Omega_m|} \\ = \; & \frac{\sigma_{m-1}p_{m-1}}{q_m|\Omega_{m-1}|} \\ < \; & \frac{\sigma_{m-1}}{|\Omega_{m-1}|}. \end{aligned}$$

To see that $\sum_{X \in \Omega_m \cap N(W)} 1/d_X \sim a_{m-1}/3$, note that $|\Omega_m \cap N(W)| = a_{m-1}$ for each $W \in \Omega_{m-1}$ and at most $18m$ pairs of non-adjacent edges can be at distance 1 apart (thereby blocking a transition) in any $X \in \Omega_m$. Since $a_{m-1} = 2(m-1)^2 - 7(m-1) \sim 2m^2$ we obtain

$$\begin{aligned} \sum_{X \in \Omega_m \cap N(W)} \frac{1}{d_X} &= \frac{a_{m-1}}{3}(1 + O(1/m)) \\ &= \frac{a_{m-1}}{3}(1 + O(1/N)). \end{aligned}$$

It follows that the total amount of flow into a central state at level m is at most

$$\begin{aligned} & 2N^{3/4} \frac{\sigma_m}{|\Omega_m|} \left(1 + O(1/N)\right)^{2N^{3/4}} \\ = \; & 2N^{3/4} \frac{\sigma_m}{|\Omega_m|} \left(1 + O(1/N^{1/4})\right). \end{aligned}$$

We can now show, similarly to (3.4), that the maximum load on any edge e between central levels m and $m-1$ is again at most $5N^{7/4}$. Now, for all pairs of central states X, Y, we route the the flow from X up (or down) equally to the states $S_X \subseteq \Omega_N$ obtained using the above path construction. Next we route it through Ω_N equally to the corresponding set S_Y. Then we route it equally along all the paths from S_Y up (or down) to Y, in the opposite direction to that in which we defined them. It now only remains to observe that we can achieve this for all central pairs of states by routing a total flow $2N^{3/4}\sigma_N/|\Omega_N|$ from each state $X \in \Omega_N$ equally to all states $Y \in \Omega_N$. Using the paths defined in section 2, the maximum loading on any edge will increased by a factor at most $2N^{3/4}\sigma_N < N^{3/4}$ from its previous value, since the flows are simply scaled up by this factor. Hence $\rho(f)$ is increased by a factor at most $N^{3/4}$, and so the mixing time $\tau \le N^{3/4}\tau_N$, where τ_N is the mixing time

from section 2. We must, of course add the $O(N^{5/4})$ "burn-in" time resulting from Lemma 3.2 to the overall mixing time, but this is comparatively negligible.

3.3 Higher degrees We could do a similar analysis for the corresponding process with larger even degree given in [2]. For $2r$-regular graph, we clothespin r non-adjacent edges. The number of sets of r non-adjacent edges in a $2r$-regular graph on n vertices satisfies

$$\binom{nr}{r} - n\binom{2r}{2}\binom{nr-2}{r-2} \le a_{n,2r} \le \binom{nr}{r},$$

so $a_{n,2r} \sim (nr)^r/r!$. When releasing a vertex there are $(2r)!/(2^r\, r!)$ ways to wire up the neighbours. But (1.1) shows that

$$\frac{|\Omega_{n+1,2r}|}{|\Omega_{n,2r}|} \sim \frac{(2nr)^r}{(2r)!} \sim \frac{a_{n,r}\, r!\, 2^r}{(2r)!}.$$

Hence detailed balance holds with σ_n satisfying (3.3), and we can proceed as before. We can still simulate the moves of the "switch" Markov chain from section 2. Hence we can define the flow as before, and bound the mixing time similarly.

Acknowledgement

We are grateful to Fred Holt for bringing [2] to our attention. We also thank Leslie Goldberg and Mark Jerrum for making their unpublished notes available to us. These provided some of the ideas used in the proof of the main result of section 2.

References

[1] B. Bollobás, *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European Journal of Combinatorics **1** (1980), 311–316.

[2] V. Bourassa and F. Holt, *SWAN: Small-world wide area networks*, in Proc. International Conference on Advances in Infrastructure (SSGRR 2003w), L'Aquila, Italy, 2003, paper #64. Available at `http://www.panthesis.com/content/swan_white_paper.pdf`

[3] C. Cooper, M. Dyer and C. Greenhill, *Sampling regular graphs and a peer-to-peer network*, pre-print of full paper, 2004. Available from `http://www.maths.unsw.edu.au/~csg/papers/cdg-preprint.pdf`

[4] C. Cooper, R. Klasing and T. Radzik, *Self maintaining networks*, pre-print, 2004.

[5] A. Frieze, *On random regular graphs with non-constant degree*, Research report 88-2, Department of Mathematics, Carnegie-Mellon University, 1988.

[6] L. Goldberg and M. Jerrum, Private communication, 2004.

[7] S. Janson, T. Łuczak and A. Ruciński, *Random Graphs*, Wiley and Sons, New York, 2000.

[8] M. Jerrum, A. Sinclair and B. McKay, *When is a graphical sequence stable?*, in *Random Graphs, Vol. 2 (Poznań, 1989)*, Wiley, New York, 1992, pp. 101–115.

[9] M. Jerrum and A. Sinclair, *Fast uniform generation of regular graphs*, Theoretical Computer Science **73** (1990), 91–100.

[10] M. Jerrum, A. Sinclair and E. Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries*, in *Proceedings of the 33rd annual ACM Symposium on Theory of Computing*, ACM Press (2001), pp. 712–721.

[11] R. Kannan, P. Tetali and S. Vempala, *Simple Markov-chain algorithms for generating bipartite graphs and tournaments*, Random Structures and Algorithms **14** (1999), 293–308.

[12] B. McKay and N. Wormald, *Uniform generation of random regular graphs of moderate degree*, Journal of Algorithms **11** (1990), 52–67.

[13] G. Pandurangan, P. Raghavan and E. Upfal, *Building low-diameter peer-to-peer networks*, IEEE Journal on Selected Areas in Communications **26** (2003), 995–1002.

[14] A. Sinclair, *Improved bounds for mixing rates of Markov chains and multicommodity flow*, Combinatorics, Probability and Computing **1** (1992), 351–370.

[15] A. Steger and N. Wormald, *Generating random regular graphs quickly*, Combinatorics, Probability and Computing **8** (1999), 377–396.

[16] G. Tinhofer, *On the generation of random graphs with given properties and known distribution*, Applied Computer Science Berichte zur Praktische Informatik **13** (1979), 265–297.

[17] N. Wormald, *Models of random regular graphs*, in *Surveys in Combinatorics, 1999* (J. Lamb and D. Preece eds.), London Math. Soc. Lecture Notes Series 267, CUP, 1999, pp. 239–298.