An Asymptotically Optimal Push-Pull Method for Multicasting over a Random Network

Vasuki Narasimha Swamy
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai, India

Rajesh Sundaresan
Department of ECE
Indian Institute of Science
Bangalore, India

Pramod Viswanath
Department of ECE
University of Illinois, Urbana-Champaign
Urbana, IL, USA

***Abstract*—We consider multicast flow problems where either all of the nodes or only a subset of the nodes may be in session. Traffic from each node in the session has to be sent to every other node in the session. If the session does not consist of all the nodes, the remaining nodes act as relays. The nodes are connected by undirected edges whose capacities are independent and identically distributed random variables. We study the asymptotics of the capacity region (with network coding) in the limit of a large number of nodes, and show that the normalized sum rate converges to a constant almost surely. We then provide a decentralized push-pull algorithm that asymptotically achieves this normalized sum rate.**

I. Introduction

In this paper, we investigate the capacity of allcast and multicast sessions over a random edge-capacitated graph.

Allcast: Consider a setting where there are n nodes, all of which are engaged in a conference over a wired network. Each node has data that needs to be made entirely available over the network to each of the other $n-1$ nodes in a simultaneous fashion. The data can be split and routed and coded in any way, so long as all nodes eventually get the information. The underlying complete undirected graph on n vertices is capacitated: each undirected edge e has capacity C_e sampled independently and identically from a distribution F. An allcast *flow* assignment is said to be feasible if the net flow over any edge (in any direction) respects the edge's capacity constraint. For each such flow assignment, let r_i be the bit-rate of traffic sent by node i to each of the other nodes. We address the question of the set of all achievable rate tuples $r_1, \cdots, r_n$ in the asymptotics of a large number of nodes n. As we shall soon see, this problem is closely related to packing of disjoint spanning trees in an edge-capacitated network with integer capacities. Minor extensions of some previous results readily yield that the achievable rate region is almost surely (a.s.)

$$\left\{ (r_1, r_2, \ldots) \ : \ \limsup_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} r_i \leq \frac{1}{2}\mathbb{E}[C] \right\} \quad (1)$$

where the expectation is of a random variable C having distribution F. The linear programming formulation of this problem is given in Section II, and the proof of (1) is given in Sections III (converse) and IV (achievability). It is known that network coding does not yield any gain in allcast settings [1], and thus we have an asymptotic characterization of allcast-capacity.

Multicast: We next address a more general setting with only a subset of k_n nodes in the multicast session, where $\lim_{n\to\infty} k_n/n = \alpha$ and $\alpha < 1$. Data from each of the k_n nodes has to reach every one of the other $k_n - 1$ nodes. The remaining $n - k_n$ nodes serve as relays. Again, in an edge-capacitated framework where each edge is independent and identically distributed (iid) with distribution F, we are interested in the set of all achievable rate tuples $r_1, \cdots, r_{k_n}$ in the asymptotics of a large number of nodes n. We demonstrate that the capacity region is almost surely

$$\left\{ (r_1, r_2, \ldots) \ : \ \limsup_{n\to\infty} \frac{1}{n}\sum_{i=1}^{k_n} r_i \leq \left(1 - \frac{\alpha}{2}\right)\mathbb{E}[C]. \right\} \quad (2)$$

The LP formulation of this problem is in Section II, proof of the converse is in Section III, and proof of achievability is in Section V. Our proof of achievability in this case is via a combination of "push" and "pull" that suggests a decentralized implementation.

Our achievability proof is based on flows (allowing for duplications) and thus do not employ network coding. While it has been observed empirically that the gain from network coding is marginal for random network topologies (see [1, p.1017]), our results of this paper imply that there is indeed no asymptotic gain from network coding with respect to the performance metric of normalized rate. Our simple "push-pull" scheme in the multicast case and the observation that asymptotically network coding does not yield a gain for almost every graph (with respect to a probability distribution on graphs) are the central contributions of this paper.

II. A Linear Programming Formulation

A. Allcast

Consider the allcast problem described in Section I. Right-away, we observe that the allcast capacity region in the undirected capacitated network does not depend on who the sender is ([1, Th. 4]). This is straightforward for flow-based schemes, but the same holds under coding as well; see [1, Th. 4]. We may therefore assume that there is only one sender (say node 1), and all other $n-1$ nodes are recipients that must receive all information sent by node 1. Thus, the rates are $(r_1, 0, 0, \ldots)$, and we characterize r_1.

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

The maximum rate is obtained by solving the following linear programming (LP) problem. Given the complete graph K_n on n vertices, let $C : E \to \mathbb{R}$ be the capacity function (a realization of iid random variables). Let $\mathcal{T}_n$ be the set of all spanning trees on the complete graph. The vertices are labeled, and so Cayley's formula tells that the number of such trees is n^{n-2}. Solve the LP (Tutte [2], Nash-Williams [3] Barahona [4], Li et al. [1]):

$$\begin{array}{lll} \text{Maximize} & \sum_{T \in \mathcal{T}_n} \lambda_T & \\ \text{subject to} \quad (1) & \sum_{T \in \mathcal{T}_n : T \ni e} \lambda_T \le C_e & \text{for all } e \\ \qquad\qquad\quad (2) & \lambda_T \ge 0 & \text{for all } T \in \mathcal{T}_n. \end{array} \tag{3}$$

Denote the maximum value of (3) as π_n. Then π_n is the maximum rate at which node 1 can allcast its information to all the other nodes. This has a simple and intuitive explanation.

- If one tags an infinitesimal information element originating at node 1 and follows the path of its spread to each of the $n-1$ recipients, one gets a directed graph rooted at the source node 1 and spanning all the n nodes.
- If the undirected version of this directed graph is not a tree, i.e., there is some cycle, some node in the cycle is receiving this information element from two other nodes. One of these links can be removed without affecting the multicast property. We can thus reduce the directed graph to a *spanning arborescence*, which is a directed graph with no incoming edges at the root node, exactly one incoming edge at every other node, and all vertices are covered.
- This spanning arborescence is in one-to-one correspondence with a tree, since the root is specified. So we may simply focus on the spanning tree associated with arborescence. Call this tree T, an element of $\mathcal{T}_n$.
- Collect all the information elements that are spread via this tree. Call its volume λ_T.

It is clear that each $\lambda_T \ge 0$ and constraint (1) above is the capacity constraint associated with each of the edges. Consequently, the value of the optimization problem in (3) is an upper bound on the optimal net flow from node 1. But it is immediate that any set of λ_T satisfying the two constraints provides a means to achieve a rate $\sum_T \lambda_T$, since λ_T units of information may be directed through the spanning arborescence associated with the tree T and root vertex 1. Thus the maximum rate of allcast flow from a single sender is π_n, the solution to the LP in (3).

When edge capacities are random, π_n is a random variable whose asymptotics we shall soon characterize.

B. Multicast

For the multicast problem, without loss of generality, let us index the session nodes as $\{1, 2, \ldots, k_n\}$. Denote by $\mathcal{T}_n(k_n)$ the set of all Steiner trees that span the vertices $1, 2, \ldots, k_n$. Obviously $\mathcal{T}_n(n) \equiv \mathcal{T}_n$. For multicast, as for allcast, the maximum simultaneously transmissible rate from one sender (node 1) to the $k_n - 1$ other recipients is the maximum value of the modified LP:

$$\begin{array}{lll} \text{Maximize} & \sum_{T \in \mathcal{T}_n(k_n)} \lambda_T & \\ \text{subject to} \quad (1) & \sum_{T \in \mathcal{T}_n(k_n) : T \ni e} \lambda_T \le C_e & \text{for all } e \\ \qquad\qquad\quad (2) & \lambda_T \ge 0 & \text{for all } T \in \mathcal{T}_n(k_n). \end{array} \tag{4}$$

Set $\alpha_n = k_n/n$, and denote the maximum value of (4) as $\pi_n(\alpha_n)$. The above LP is the same as that of (3) with $\mathcal{T}_n$ replaced by the less restrictive $\mathcal{T}_n(k_n)$.

III. An Upper Bound

Consider the following definitions.

- Let χ_n and $\chi_n(k_n)$ denote the *maximum throughput achievable* in the allcast and multicast settings with the added possibility of network coding at each node. (The dependence of these quantities on the edge capacities is understood and suppressed).
- Let η_n denote the *strength* of the allcast network defined as follows. Let $\mathcal{P}$ denote the set of all partitions of the vertex set $\{1, 2, \ldots, n\}$. Consider a partition $\wp \in \mathcal{P}$. Let $\partial\wp$ denote the set of intercomponent edges. Define
$$\eta_n := \min_{\wp \in \mathcal{P}} \frac{\sum_{e \in \partial\wp} C_e}{|\wp| - 1} \tag{5}$$
where $|\wp|$ denotes the number of subsets in the partition.
- Let $\eta_n(k_n)$ denote the strength of the multicast network with k_n nodes in the session. This is defined as follows. Let $\mathcal{P}(k_n)$ denote the set of all partitions of the vertex set $\{1, 2, \ldots, n\}$ such that each component of a partition contains at least one of the session nodes $\{1, 2, \ldots, k_n\}$. Define
$$\eta_n(k_n) := \min_{\wp \in \mathcal{P}(k_n)} \frac{\sum_{e \in \partial\wp} C_e}{|\wp| - 1}. \tag{6}$$

Li et al. [1] showed the following result.

Theorem 1: (Li et al. [1, Th. 2 and Th. 3])
(a) For any allcast session, $\pi_n = \chi_n = \eta_n$.
(b) For any multicast session, $\pi_n(k_n) \le \chi_n(k_n) \le \eta_n(k_n)$. □

We can easily find upper bounds on η_n and $\eta_n(k_n)$ as in the following theorem.

Theorem 2: Let $\{C_{i,j}\}_{1 \le i < j \le n}$ denote the undirected edge capacities. We then have the following upper bounds:

$$\eta_n \le \frac{1}{n-1} \sum_{1 \le i < j \le n} C_{i,j} \tag{7}$$

$$\eta_n(k_n) \le \frac{1}{k_n} \left(\sum_{i \le k_n} \sum_{j > k_n} C_{i,j} + \sum_{1 \le i < j \le k_n} C_{i,j} \right). \tag{8}$$

As a consequence, with $\lim_{n\to\infty} k_n/n = \alpha$, the inequalities

$$\limsup_{n\to\infty} \frac{\eta_n}{n} \leq \frac{1}{2}\mathbb{E}[C] \tag{9}$$

$$\limsup_{n\to\infty} \frac{\eta_n(k_n)}{n} \leq \left(1-\frac{\alpha}{2}\right)\mathbb{E}[C] \tag{10}$$

hold almost surely. □

Proof: Consider the partition $\wp = \{\{1\},\{2\},\cdots,\{n\}\}$. There are n subsets in the partition, and $\partial\wp$ is the set of all edges. Apply now the definition (5) of η_n and we immediately get (7) as the upper bound for the allcast case.

For the multicast case, consider the partition

$$\wp = \{\{1\},\{2\},\cdots,\{k_n\},\{k_n+1,\ldots,n\}\}.$$

There are k_n+1 subsets in the partition. The set of edges in $\partial\wp$ are

$$\{(i,j): 1\leq i\leq k_n, j\geq k_n\} \cup \{(i,j): 1\leq i<j\leq k_n\}.$$

Apply now the definition (6) of $\eta_n(k_n)$ and we immediately get (8) as the upper bound for the multicast case.

Note that $|\partial\wp| = n(n-1)/2$ for allcast, and

$$|\partial\wp| = k_n(n-k_n) + \frac{k_n(k_n-1)}{2} = k_n\left(n-\frac{k_n+1}{2}\right) \tag{11}$$

for multicast.

Using $|\partial\wp| = n(n-1)/2$ for allcast in (7), we obtain

$$\frac{\eta_n}{n} \leq \frac{1}{2}\frac{1}{|\partial\wp|}\sum_{e\in\partial\wp} C_e.$$

The sum on the right-hand side is composed of independent and identically distributed random variables. Consequently, the right-hand side converges almost surely to $\frac{1}{2}\mathbb{E}[C]$ by the strong law of large numbers, and we obtain (9).

For the multicast case, use (11) in (8) to obtain

$$\frac{\eta_n(k_n)}{n} \leq \left(1-\frac{(k_n+1)}{2n}\right)\frac{1}{|\partial\wp|}\sum_{e\in\partial\wp} C_e.$$

Again by an application of the strong law of large numbers, the conclusion (10) follows. ■

Observe that, by Theorem 1, the upper bounds in Theorem 2 apply for capacity with the possibility of network coding. Let us now turn to achievability of this rate with no network coding.

IV. Allcast: Achievability

Achievability follows directly from prior results.

Theorem 3: For the allcast problem, we have

$$\lim_{n\to\infty} \frac{\pi_n}{n} = \frac{1}{2}\mathbb{E}[C] \text{ a.s.}$$

□

Proof: The converse was already shown in (9). So it suffices to show achievability.

Furthermore, it suffices to prove achievability for graphs whose edge capacities are independent Bernoulli random variables with parameter p, i.e., $\liminf_{n\to\infty} \pi_n/n \geq p/2$ almost surely. By following standard techniques of truncation, scaling, and quantization, see for example [5], one can show that $\liminf_{n\to\infty} \pi_n/n \geq (1/2)\mathbb{E}[C]$ for any generic distribution with an expectation.

Graphs whose edge capacities are independent Bernoulli random variables with parameter p are the Erdős-Rényi random graphs denoted $G(n,p)$. For such graphs, even for p as low as $(28\log n/n)^{1/3}$, Catlin et al. [6, Sec. 3] proved the stronger result that

$$\pi_n = \left\lfloor \frac{\sum_{1\leq i<j\leq n} C_{i,j}}{n-1} \right\rfloor \text{ a.s.}$$

By the strong law of large numbers, it is then obvious that $\lim_n \pi_n/n = p/2$ almost surely. ■

V. Multicast: Achievability

While one could in principle proceed as in Catlin et al. [6] to prove achievability, we wish to provide a more constructive proof of achievability for cases when $\alpha < 1$. We shall use Theorem (3) in the proof. Our constructive procedure does not yet handle the boundary case when $\alpha = 1$.

Theorem 4: For the multicast problem with k_n in the session, let $\lim_{n\to\infty} k_n/n = \alpha < 1$. We then have

$$\lim_{n\to\infty} \frac{\pi_n(k_n)}{n} = \left(1-\frac{\alpha}{2}\right)\mathbb{E}[C] \text{ a.s.}$$

□

Proof: As in the proof of Theorem 3, converse was already shown in (10). So showing achievability suffices, and further this can be shown on Erdős-Rényi random graphs with parameter p.

Next observe that the subset of session nodes alone form a complete graph with k_n vertices for which Theorem 3 is applicable. Using the scheme suggested by that theorem, we have

$$\pi_n^{(1)} \geq \frac{\sum_{1\leq i<j\leq k_n} C_{i,j}}{k_n-1} \tag{12}$$

is achievable for simultaneous multicast, almost surely, without using any of the relay nodes.

Removing these direct links between the session nodes, we end up with a graph in Figure 1, where the session nodes are now only connected to the $m_n = n-k_n$ relay nodes. The edge to each relay node from each session node has Bernoulli(p) capacity. Further the relay nodes have inter-relay edge capacities that are independent Bernoulli(p) random variables. We now claim that a rate $\pi_n^{(2)}$ can be simultaneously multicast to the k_n session nodes (solely with the help of the relay nodes), and the rate almost surely satisfies

$$\liminf_n \frac{\pi_n^{(2)}}{m_n} \geq p. \tag{13}$$

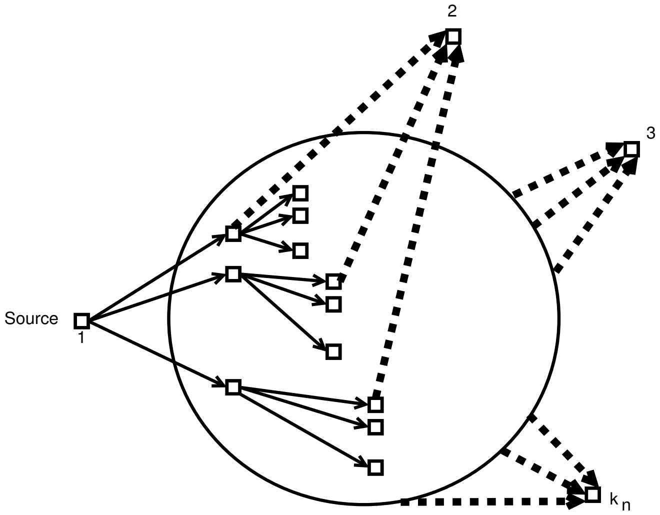

Figure 1. Graph with all links between session nodes removed. Source pushes bits to first hop relays which then push to secondary relays (solid lines). The session nodes pull the bits from either first hop relays or secondary relays (dashed lines).

Supposing the claim is true, we can put (12) and (13) together, and get an achievable rate $\pi_n = \pi_n^{(1)} + \pi_n^{(2)}$ that satisfies the following almost surely:

$$\begin{aligned}
&\liminf_n \frac{\pi_n}{n} \\
&= \liminf_n \left(\frac{\pi_n^{(1)}}{n} + \frac{\pi_n^{(2)}}{n} \right) \\
&\geq \liminf_n \left(\frac{\pi_n^{(1)}}{n} \right) + \liminf_n \left(\frac{\pi_n^{(2)}}{n} \right) \\
&\geq \liminf_n \left(\frac{k_n}{2n} \frac{\pi_n^{(1)}}{k_n/2} \right) + \liminf_n \left(\frac{m_n}{n} \frac{\pi_n^{(2)}}{m_n} \right) \\
&\geq \left(\lim_n \frac{k_n}{2n} \right) \cdot p + \left(\lim_n \frac{m_n}{n} \right) \cdot p \\
&\quad \text{(using (12) and (13))} \\
&= \left(\frac{\alpha}{2} \right) p + (1-\alpha) p \\
&= \left(1 - \frac{\alpha}{2} \right) p
\end{aligned}$$

which establishes the theorem.

We now proceed to verify the claim in (13). Fix $\varepsilon > 0$ sufficiently small. Data is "pushed" as follows.

Push step 1: The source sends a total of $m_n p(1-\varepsilon)$ bits to relays connected to it. (Rounding to integer can easily be handled and is ignored for expository purposes). Each relay carries a unique bit. If there are more neighbors than $(m_n p(1-\varepsilon))$, the extra relays are not supplied a bit. Let us call all those nodes that receive a bit directly from the source as *first hop relays*. The case when there are fewer first hop relays than $m_n p(1-\varepsilon)$ is an event having negligible probability. Indeed, let A_1 be the event that the degree of the source node is less than $m_n p(1-\varepsilon)$. By Bernstein's inequality, $\Pr\{A_1^{(n)}\} \leq e^{-m_n c_1}$ for some $c_1 > 0$.

Push step 2: Each first hop relay node relays its bit to each neighbor that is not already a first hop relay. The resulting bit carrying relays are called *secondary* relays. Note that there may be secondary relays that are in contact with the source, but did not receive any bit on account of the source's degree exceeding $m_n p(1-\varepsilon)$.

The bit-map of available bits with relays is as follows. Let us index the bits as $1, 2, \ldots, m_n p(1-\varepsilon)$, and relays as $1, 2, \ldots, m_n$. Then row i indicates what relay i contains.

In the table, $X_{i,j} = 1$ if relay i (with $m_n p(1-\varepsilon) < i \leq m_n$) is connected to first hop relay j. Clearly, the presence or absence of this edge is independent of all other events, and so $X_{i,j}$ is a Bernoulli(p) random variable.

	Bit 1	Bit 2	$\cdots$	Bit $m_n p(1-\varepsilon)$
First hop relay 1	1	0	$\cdots$	0
First hop relay 2	0	1	$\cdots$	0
$\vdots$			$\ddots$	
First hop relay $m_n p(1-\varepsilon)$	0	0	$\cdots$	1
Relay $1 + m_n p(1-\varepsilon)$				
$\vdots$			$((X_{i,j}))$	
Relay m_n				

Data is "pulled" from the relays by the session nodes as follows. Consider a session node i.

Pull step 1: If a session node i is connected to a first hop relay, it pulls the corresponding bit. Let $A_2^{(n)}(i)$ be the event there are less than $m_n p^2(1-\varepsilon)^2$ first hop relays that our session node is in contact with. Again by Bernstein's inequality, $\Pr\{A_2^{(n)}(i)\} \leq e^{-m_n c_2}$ for some $c_2 > 0$.

Pull step 2: The session node now has to pull the remaining bits from the secondary relays. The number of bits that remain to be pulled is

$$b_n := m_n p(1-\varepsilon) - m_n p^2(1-\varepsilon)^2 = m_n p(1-\varepsilon)(1-p(1-\varepsilon)).$$

The number of relays that are not first hop relays is at least

$$m_n - m_n p(1-\varepsilon) = m_n(1-p(1-\varepsilon)).$$

Let $A_3^{(n)}(i)$ be the event that the session node is connected to fewer than $p(1-\varepsilon)$ fraction of these nodes, i.e., to fewer than $m_n(1-p(1-\varepsilon)) \cdot p(1-\varepsilon) = b_n$ such relays. By Bernstein's inequality again, $\Pr\{A_3^{(n)}(i)\} \leq e^{-m_n c_3}$ for some $c_3 > 0$.

Pull step 3: Assume now that $A_n^{(1)} \cup A_n^{(2)}(i) \cup A_n^{(3)}(i)$ does not occur. Then the matrix rows corresponding to the secondary relays in contact with the session node and the columns corresponding to the bits not yet pulled constitutes a $b_n \times b_n$ square submatrix whose entries are conditionally iid Bernoulli(p) random variables. We may view this as a bipartite graph with (non-first-hop) relays on the one side and not-yet-pulled bit indices on the other side. There are b_n vertices on each side. The edges of this bipartite graph are $X_{k,l}$ which are independent Bernoulli(p) random variables. For a selected relay k connected to session node i, if $X_{k,l} = 1$, then the relay is a secondary relay and has bit l. It may contain other bits of interest to session node i, but can send at most 1 bit to node i. If we have a complete matching (where all b_n selected relays and all unpulled bits are matched), then the session node can pull these remaining bits from the selected relays without violating the capacity constraint of one bit per edge.

The probability that such a matching does not exist, say $M_n(i)$, can be upper bounded by $\gamma(b_n)$, where $k(n)\gamma(b_n)$ is summable, using [7, Lem. 7.12, p.174]. (See Appendix).

In the final analysis, the event that some session node is unable to pull all the bits is

$$A_1^{(n)} \cup_{i=2}^{k_n} \left(A_n^{(2)}(i) \cup A_n^{(3)}(i) \cup M_n(i) \right).$$

Its probability is upper bounded by

$$e^{-m_n c_1} + k_n(e^{-m_n c_2} + e^{-m_n c_2} + \gamma(b_n)).$$

Using $\alpha < 1$, summing the above over n, using the fact that $k_n\gamma(n)$ is summable, and the Borel-Cantelli lemma, we obtain that almost surely all the session nodes will be able to pull $m_n p(1-\varepsilon)$ bits. By considering rational ε and staying out of the union of all the associated null sets, it follows that $\liminf_n \pi_n^{(2)}/m_n \geq p$ (a.s.). This concludes the proof. ■

VI. Summary

Our main contributions are the following.

- For multicast sessions on random graphs, specifically Erdős-Rényi random graphs, the maximum asymptotic rate can be achieved via flows. This was already known for the allcast case $k_n = n$. We studied the multicast case when $k_n/n \to \alpha < 1$.
- We proposed a push-pull scheme for data distribution in a multicast session with k_n nodes, $k_n/n \to \alpha < 1$. Our scheme is decentralized and easily implementable.

Appendix A
The Existence of a Bipartite Matching

The following lemma, taken from Bollobás, is key to showing that matchings exist almost surely and one can pull the b_n bits from secondary relays. We present the result for a bipartite graph with n vertices.

Lemma 5: ([7, Lem. 7.12, p. 174]). Let G be a bipartite graph with vertex classes V_1 and V_2, $|V_1| = |V_2| = n$. Suppose G has no isolated vertices and it does not have a complete matching. Then there is a set $A \subset V_i \quad (i = 1, 2)$ such that:

(i) $\Gamma(A) = \{y : (x,y) \in E(G) \text{ for some } x \in A\}$ has $|A|-1$ elements,
(ii) the subgraph of G spanned by $A \cup \Gamma(A)$ is connected and
(iii) $2 \leq |A| \leq (n+1)/2$.

Let $n_1 = \lfloor (n+1)/2 \rfloor$. We follow Bollobás's arguments on [7, p. 174]. Let F_a denote the event that there is a set $A \subset V_i \quad (i = 1 \text{ or } 2)$, $|A| = a$, satisfying (i)-(iii) of Lemma 5. The subgraph spanned by $A \cup \Gamma(A)$ is connected, and so must have at least $2a-2$ edges. Further the vertices of A must not be connected to any vertex in $V_{3-i} - \Gamma(A)$ The probability that this happens for $A \subset V_1$ is at most

$$\binom{a(a-1)}{2a-2} p^{2a-2}(1-p)^{a(n-a+1)}.$$

There are $\binom{n}{a}$ choices for $A \subset V_1$ with $|A| = a$, $\binom{n}{a-1}$ choices for $\Gamma(A)$, and an extra factor of 2 to account for $A \subset V_i$, $i = 1, 2$. Using these and $\binom{n}{k} \leq (en/k)^k$, we get

$$\begin{aligned} P\left(\bigcup_{a=2}^{n_1} F_a\right) &\leq \sum_{a=2}^{n_1} P(F_a) \\ &= 2\sum_{a=2}^{n_1} \binom{n}{a}\binom{n}{a-1}\binom{a(a-1)}{2a-2} \\ &\qquad \times p^{2a-2}(1-p)^{a(n-a+1)} \\ &\leq 2\sum_{a=2}^{n_1} \left(\frac{en}{a}\right)^a \left(\frac{en}{a-1}\right)^{a-1} \left(\frac{ea}{2}\right)^{2a-2} \\ &\qquad \times p^{2a-2}(1-p)^{a(n-a+1)} \\ &\leq \text{const.} \sum_{a=2}^{n_1} e^{3a} n^{2a-1} p^{2a-2}(1-p)^{a(n-a+1)} \\ &=: \gamma_1(n), \end{aligned}$$

where the last equation is the defining equation for $\gamma_1(n)$.

The probability that there is some isolated node is upper bounded by $\gamma_2(n) := n(1-p)^{n-1}$. Thus the probability that the bipartite graph of n vertices does not have a complete matching is at most $\gamma(n) = \gamma_1(n) + \gamma_2(n)$.

When we have b_n vertices on each side, the above bound turns out to be $\gamma(b_n)$. Straightforward computations yield that $\sum_n k_n\gamma(b_n)$ is summable, details of which we are omitted.

Acknowledgements

This work was done as part of a summer internship by V. N. Swamy in her junior undergraduate year at the Indian Institute of Science, and where P. Viswanath was on sabbatical leave at the time. This work is supported by the Department of Science and Technology, Government of India, by the University Grants Commission, India, and by the US National Science Foundation under grant CCF-1017430. The authors would like to thank Prof. Navin Kashyap for bringing references [4] and [6] to their attention.

References

[1] Z. Li, B. Li, and L. C. Lau, "A constant bound on throughput improvement of multicast network coding in undirected networks," vol. 55, no. 3, pp. 1016–1026, Mar. 2009.

[2] W. T. Tutte, "On the problem of decomposing a graph into n connected factors," *J. London Math. Soc.*, vol. 36, pp. 221–230, 1961.

[3] C. S. J. A. Nash-Williams, "Edge-disjoint spanning trees of finite graphs," *J. London Math. Soc.*, vol. 36, pp. 445–450, 1961.

[4] F. Barahona, "Packing spanning trees," *Mathematics of Operations Research*, vol. 20, no. 1, pp. 104–115, Feb. 21995.

[5] M. Khandwawala and R. Sundaresan, "Optimal multicommodity flow through the complete graph with random edge-capacities," *Journal of Applied Probability*, vol. 47, no. 1, pp. 201–215, Mar. 2010.

[6] P. A. Catlin, Z. Hong Chen, and E. M. Palmer, "On the edge arboricity of a random graph," *ARS Combinatorica*, vol. 35-A, pp. 129–134, 1995.

[7] B. Bollobás, *Random Graphs*, 2nd ed. Cambridge, UK: Cambridge University Press, 2001, vol. Cambridge studies in advanced mathematics, no. 73.