Single Round Simulation on Radio Networks *

Noga Alon †
IBM Almaden Research Center,
San Jose, CA 95120, and
Sackler Faculty of Exact Sciences,
Tel-Aviv University,
Ramat Aviv, Tel Aviv, Israel.

Amotz Bar-Noy ‡
IBM Research Division,
T. J. Watson Research Center,
Yorktown Heights, NY 10598.

Nathan Linial §
Department of Computer Science,
the Hebrew University,
Jerusalem 91904, Israel.

David Peleg ¶
Department of Applied Mathematics,
the Weizmann Institute,
Rehovot 76100, Israel.

February 24, 2002

[*] An abstract of this paper appeared in *Proc. 21st Symp. on Theory of Computing,* (1989), pp. 274-285.

[†] Supported in part by a Bergmann Memorial grant.

[‡] This work was carried out while this author was visiting Stanford University. Supported in part by a Weizmann fellowship, by contract ONR N0014-85-C-0731 and by a grant of Stanford Center for Integrated Systems.

[§] This work was carried out while this author was visiting IBM Almaden Research Center and Stanford University.

[¶] This work was carried out while this author was visiting Stanford University. Supported in part by contract ONR N00014-85-C-0731 and by a grant of Stanford Center for Integrated Systems.

Abstract

A *radio network* is a synchronous network of processors that communicate by transmitting messages to their neighbors. A processor receives a message in a given step if and only if it is silent then and precisely one of its neighbors transmits. This stringent rule poses serious difficulties in performing even the simplest tasks. This is true even under the overly optimistic assumptions of centralized coordination and complete knowledge of the network topology.

We look at the question of simulating two of the standard message-passing models on a radio network. In the general message-passing model, a processor may send each of its outgoing neighbors a possibly different message in each round. In the uniform message-passing model, in each round a processor must send an identical message to all its outgoing neighbors. Both message-passing models can easily simulate the radio model with no overhead. In the other direction, we propose and study a primitive called the *single-round simulation (SRS)*, enabling the simulation of a single round of an algorithm designed for the standard message models. We give lower bounds for the length of SRS schedules for both models, and supply constructions or existence proofs for schedules of matching (or almost matching) lengths.

1 Introduction

Communication is a major aspect of every distributed system. Its role and cost are widely studied in many areas of Computer Science. Less attention is being paid, though, to the difficulties arising in communicating over various communication media; usually one adopts the most convenient medium of point-to-point non-interfering communication lines. A notable exception is multiple-access channels (such as Ethernet) which did receive rather extensive attention (cf. [HLR, GFL, GGMM]). The main difference is that two stations transmitting at the same time interfere with each other.

In the growing field of communications, other mechanisms are being proposed, studied and implemented. However, important aspects of these new communication modes and the relationships between them are left somewhat neglected, and there is much more to be said about these issues from algorithmic and computational standpoint.

In this paper we study packet radio networks [BGI, CK1, CK2, CK3, CW, GVF, K, KGBK, SC]. A *radio network* is a directed graph $G = (V, E)$ whose vertices are processors (or stations) that communicate among themselves in synchronous time-slots using radio transmissions. (In fact, the Ethernet is a special case of a radio network.) The properties of this medium are described by the following rules. In each step a processor can either *transmit* or *keep silent*. A processor x *receives* a message from a processor y in a given step if and only if x keeps silent and y is the only incoming neighbor of x (i.e., such that $(y, x) \in E$) to transmit in this step. If more than one incoming neighbor of x transmits, a *collision* occurs in which case x hears only *noise*, but no message. On hearing noise, x can only conclude that some of his neighbors tried to transmit at this round. Also, a processor cannot hear while transmitting. Directed edges reflect asymmetric situations, e.g., some stations may be more powerful transmitters than others.

It is intuitively clear that the possibility of collisions should make radio networks hard to coordinate and control, and that performing even the simplest tasks may pose serious difficulties. This effect is especially marked when the processors operate in a distributed fashion, and have no a-priory knowledge of the network's topology. This difficulty was demonstrated, for instance, by the $\Omega(n)$ lower bound given by [BGI] for the time required for deterministic distributed broadcast protocols in radio networks. Their lower bound takes full advantage of the assumptions of the distributed environment and processors' ignorance of the graph topology.

In order to study the inherent limitations of radio communication in this way, we need to neutralize the effects of (the absence of) knowledge of the network. This can be done by concentrating on *topology-bound schedules*, rather than *distributed algorithms.* A topology-bound schedule is a pre-fixed, oblivious algorithm designed for the particular network at hand. Such a schedule supplies every vertex with an individual list of instructions specifying what actions to take in each round of the run. (Note that the assumption of a central control being oblivious causes no loss in power.) To be more specific, a *schedule* S is a list $(T_1, \ldots, T_t)$ of *transmissions.* For each round i, $i = 1, 2, \ldots, t$, the set $T_i = \{(v_1, M_1), (v_2, M_2), \ldots\}$ specifies the processors that have to transmit in round i and the contents of their messages. Such a schedule may be generated by a *centralized* algorithm whose input contains the network's topology. Clearly, a schedule for a given problem on a given graph will generally outperform any distributed algorithm for the same problem (run on the same graph). Thus when studying worst scenarios, lower bounds on the number of rounds required for a schedule apply also to distributed algorithms. Conversely, the existence of a distributed algorithm induces a schedule of the right complexity for the problem on every graph (although the schedules derived in this way might not be optimal for some graphs).

In [ABLP] we prove the existence of a family of radius-2 networks on n vertices in which any broadcast schedule requires at least $\Omega(log^2 n)$ rounds of transmissions. This matches an upper bound of $O(log^2 n)$ rounds for networks of radius 2 proved in [BGI]. Thus, even with a complete knowledge of the topology the radio network causes many difficulties in achieving simple tasks such as broadcasting.

When trying to demonstrate that one mode of operation is slower or weaker than another, a natural approach is to study possible simulations between the two modes. This paper concerns simulations between the standard synchronous point-to-point message-passing model and the radio model. Clearly, the simulation of algorithms for radio networks on message-passing systems with the same underlying topology can be achieved in a straightforward manner with no overhead at all (in terms of number of rounds). However, in the other direction our results imply that the simulation of message-passing based algorithms in radio networks causes a considerable slowdown, typically depending on the vertex degrees in the network. This observation lends additional support to one's feeling about the relative difficulty in radio communication.

The notion of network simulation may have practical significance as well. Whenever a new type of communication mode is invented, new algorithms have to be developed for it for all standard network operations. Simulation procedures could help to convert algorithms designed for networks with the same topology but different means of communication to algorithms for the new communication mode. In particular, since designing algorithms for radio networks from scratch turns to be a hard task, the simulation of algorithms for standard message-passing systems may prove to be a plausible approach.

We concentrate on *round-by-round* simulations where a separate *phase* of radio transmission rounds is dedicated to simulating each *single round* of the original algorithm. We propose and study a general primitive called *single-round simulation (SRS)*, serving as a building block in such simulations. The role of this primitive is to ensure that every message passed by the original algorithm during the simulated round will be transmitted (and received) during the simulating phase. The hardest round to simulate is one where a message is to be sent over each link of the network. The simulating phase should guarantee that for every edge (u, v) there is a step in which u transmits the message M_{uv} designated to v and v manages to receive it. (The resulting communication primitive, SRS, bears close resemblance to the "network testing" primitive of [EGMT], although their roles are different. The task of performing SRS on a radio network is in fact equivalent to the problem described as phase allocation in [CK3].)

Denote by Δ_{in} and by Δ_{out} the maximum indegree and outdegree of any vertex in the graph, respectively, and let $\Delta = \max\{\Delta_{in}, \Delta_{out}\}$. (Without loss of generality, $\Delta_{in}, \Delta_{out} \geq 2$.) We deal with two variants of the message-passing model. In the *general model*, a processor x may send each of its outgoing neighbors y a (possibly different) message M_{xy} in each round. A more restricted model is what we call the *uniform model*, where in each round a processor x may send a single message M_x to all its outgoing neighbors.

Clearly every algorithm that works in the uniform model works also in the general model. Conversely, a single round of an algorithm for the general model may require $\Omega(\Delta_{out})$ rounds in a simulating algorithm in the uniform model, since each processor needs to send its messages to its outgoing neighbors in Δ_{out} separate rounds. If messages of unbounded length are allowed then the two models are equivalent. To simulate a round of the general model each processor concatenates all its messages along with indicators of the destinations. This triviality is avoided by considering the individual message as indivisible and by charging a unit cost for their transmission. However we assume messages to have a fixed length.

Our main results for the single-round simulation problem are as follows. For the general model we first consider single-round simulation (SRS) schedules. We prove a matching lower and upper bound of $\Theta(\Delta_{in}\Delta_{out})$ rounds. The upper bound is achieved by a simple construction method for SRS schedules, based on coloring a certain related graph called the *interference graph*. Our lower bound is based on a family of graphs with $\Delta_{in} = \Delta_{out} = \Delta$ that forces every SRS schedule to last at least $\Omega(\Delta^2)$ rounds.

We then turn our attention to distributed algorithms for SRS in the general model. We present a randomized (Las-Vegas) distributed SRS algorithm for general graphs. With probability $1-p$ $(0 < p < 1)$, it requires $O(\Delta_{in}\Delta_{out} \log \frac{n}{p})$ rounds. Deterministic SRS algorithms are presented under the assumption that processors have distinct identities $\{1, \ldots, n\}$. In addition to the obvious $O(n\Delta_{out})$ round algorithm, we present an $O\left(\Delta_{out}\Delta_{in}^2 \frac{\log^2 n}{\log(\Delta_{in} \log n)}\right)$ round deterministic algorithm.

For the uniform model there is an obvious *global* lower bound of $\Omega(\Delta_{in})$ on the length of an SRS schedule for every graph. Further, there are graphs for which we can show a lower bound of $\Omega(\Delta_{in} \log \Delta)$ rounds. In these graphs Δ_{in} and Δ_{out} are considerably different. For the special case of an undirected network in which $\Delta = \Delta_{in} = \Delta_{out}$, we describe another lower bound of $\Omega(\Delta \log \Delta)$. A probabilistic argument using the Lovász Local Lemma establishes the existence of an SRS schedule of $O(\Delta_{in} \log \Delta)$ rounds for every graph. We also present a (centralized) algorithm for constructing a schedule of $O(\Delta_{in} \log n)$ rounds for every graph. The previously known solution required $O(\Delta^2)$ rounds [CK3].

Finally we consider distributed algorithms for SRS in the uniform model. We exhibit a randomized (Las-Vegas) distributed algorithm for general graphs, which with probability $1-p$ requires only $O(\Delta_{in} \log \frac{n}{p})$ rounds. We also present deterministic algorithms based on the assumption of distinct id's $\{1, \ldots, n\}$. These algorithms are faster by a factor of Δ_{out} than the corresponding algorithms for the general model.

Let us comment that our distributed algorithms are run on the radio network itself, and require no control message exchange (i.e., the only messages sent are those the processors wish to pass to their neighbors). In contrast, the distributed algorithm of [CK3] has to be run on a standard point to point message passing network built on the same set of vertices as the radio network and assumed to have the same topology; such a network is rarely available in practice.

Remark: Unless specified otherwise, all logarithms are to base 2. For easier readability we omit all floor and ceiling rounding throughout. Also, we assume without further notice that all our parameters are sufficiently large whenever needed.

2 The general model

Let us first give a precise description of the SRS primitive for the general message-passing model. We are given a directed graph $G = (V, E)$. For every edge $(x, y) \in E$, the processor x has a message M_{xy} destined for y. The single-round simulation problem calls for the delivery of all of these messages. We say that an edge (x, y) is *satisfied* by a transmission step T if T contains the instruction (x, M_{xy}) and neither y nor any other incoming neighbor z of y transmits in T (so y gets to receive M_{xy}). A schedule $S = (T_1, T_2, \ldots)$ satisfies an edge if at least one of its transmissions does. A schedule S is a *single-round simulation (SRS) schedule* if it satisfies all edges in E.

2.1 Schedules

For every directed graph $G = (V, E)$, define the simple undirected *interference graph* of G, $I(G) = (V_{I(G)}, E_{I(G)})$, as follows. The vertices of $I(G)$ are the edges of G ($V_{I(G)} = E$). There is an edge between the vertices (x, y) and (z, w) if at least one of the following two conditions holds:

1. The edges (x, y) and (z, w) are adjacent edges in G (i.e., $|\{x, y\} \cap \{z, w\}| \geq 1$ and $(x, y) \neq (z, w)$).
2. At least one of the two edges (z, y) and (x, w) exists in G.

Note that the neighbor set of (x, y) in $I(G)$ consists of at most $(2\Delta_{in} - 1) + (2\Delta_{out} - 1)$ edges (z, w) contributed by Condition 1 and at most $2(\Delta_{in} - 1)(\Delta_{out} - 1)$ contributed by Condition 2. Altogether we get the following lemma.

Lemma 2.1 *The maximum degree of $I(G)$ is at most $2\Delta_{in}\Delta_{out}$.* ∎

Denote the chromatic number of an undirected graph H by $\chi(H)$. The following lemma expresses the least length of an SRS schedule for G in terms of $I(G)$.

Lemma 2.2 *$\chi(I(G))$ rounds are necessary and sufficient for an SRS schedule for the general model.*

Proof: Let $r = \chi(I(G))$ and let

$$\varphi : V_{I(G)} \rightarrow \{1, \ldots, r\}$$

be a coloring of $I(G)$ with r colors. Construct the following SRS schedule S. If (x, y) is colored by φ with the color i, $1 \leq i \leq r$, then processor x transmits M_{xy} (i.e., the message designated to processor y) in round i of the schedule S.

Since for every $z \in V$ the vertices (x, y) and (x, z) are adjacent in $I(G)$ this rule is well-defined and in each round x has to transmit at most one message. Every directed edge of G is colored, so each processor transmits each of its messages exactly once. It remains to prove that all of the transmissions succeed.

Assume that in round i of the schedule S processor x transmits M_{xy}. Since for any $z \in V$ the vertices (y, z) and (x, y) are adjacent in $I(G)$ it follows that y does not transmit at round i. The second rule in constructing the edges of $I(G)$ guarantees that at round i none of the incoming neighbors of y transmits. The above two arguments assure that this transmission succeeds and y receives M_{xy} at round i.

For the other direction, assume that S is an SRS schedule of r rounds. Color the vertex (x, y) of $I(G)$ with the color i where i is the first round in S in which x succeeds in transmitting M_{xy}. The construction of $I(G)$ and arguments as before prove that this is a legal coloring. ▮

By Lemma 2.1 no degree in $I(G)$ exceeds $2\Delta_{in}\Delta_{out}$, hence a greedy coloring of $I(G)$ and Lemma 2.2 yields the following Theorem.

Theorem 2.1 *For every directed graph G there is a (polynomial time constructible) SRS schedule for the general model of $2\Delta_{in}\Delta_{out} + 1$ rounds.* ▮

If G contains a complete bidirected bipartite graph with Δ vertices in each side, then $I(G)$ contains a clique of size $2\Delta^2$, whence $\chi(I(G)) \geq 2\Delta^2$ and the next theorem follows from Lemma 2.2.

Theorem 2.2 *For every $\Delta \geq 2$ and $n \geq \Delta$ there exists a graph with n vertices and maximum indegree and outdegree Δ for which every SRS schedule requires $2\Delta^2$ rounds.* ▮

2.2 Distributed algorithms

We now turn our attention to the subject of finding a randomized distributed algorithm for the problem. We assume that each processor knows the identity of its neighbors in the network, as well as n, Δ_{in} and Δ_{out}, but does not know the entire topology. The following procedure will be used several times in the sequel.

Procedure `Random_Transmit`(M, r)**:** In each round i, $1 \leq i \leq r$, transmit M with probability $\frac{1}{\Delta_{in}}$, and keep silent with probability $1 - \frac{1}{\Delta_{in}}$.

In all cases, this procedure is applied at every processor x simultaneously using the same r values, with appropriate messages M_x. For an edge (x, y) denote by A_{xy} the event: "the processor y fails to receive M_x over the edge e_{xy} during all r rounds".

Lemma 2.3 *For every edge* (x, y)*,*

$$\Pr(A_{xy}) \leq \exp\left(-\frac{r}{2e\Delta_{in}}\right).$$

Proof: The probability that a single transmission step of the procedure succeeds on the edge (x, y) for y with indegree d is bounded below by

$$\frac{1}{\Delta_{in}}\left(1 - \frac{1}{\Delta_{in}}\right)^{d-1} \geq \frac{1}{\Delta_{in}}\left(1 - \frac{1}{\Delta_{in}}\right)^{\Delta_{in}} \geq \frac{1}{2e\Delta_{in}}.$$

Therefore, it is only with probability at most

$$\left(1 - \frac{1}{2e\Delta_{in}}\right)^{r} < \exp\left(-\frac{r}{2e\Delta_{in}}\right)$$

that the algorithm fails to transmit M_x on (x, y) in all r rounds. ∎

Our randomized algorithm for SRS consists of Δ_{out} phases. Let $y_1, \ldots, y_k$ be the outgoing neighbors of x in the network. In phase i $(1 \leq i \leq k)$, x applies the procedure `Random_Transmit`(M_{xy_i}, r) where

$$r = \left\lceil 2e\Delta_{in} \ln \frac{n\Delta_{out}}{q} \right\rceil$$

for some *safety parameter* $0 < q < 1$. As a result of Lemma 2.3 we get

Lemma 2.4 *The probabilistic algorithm succeeds in transmitting on all the edges with probability* $1 - q$*.*

Proof: Denote by P the probability that the algorithm fails on some edge.

$$P = \Pr(\bigcup_{(x,y)\in E} A_{xy}) \leq \sum_{(x,y)\in E} \Pr(A_{xy}).$$

By Lemma 2.3 and the choice of r

$$P \leq n\Delta_{out} \exp\left(-\frac{2e\Delta_{in} \ln \frac{n\Delta_{out}}{q}}{2e\Delta_{in}}\right) = q.$$

Thus the entire algorithm succeeds on all edges with probability at least $1-q$. ▮

Theorem 2.3 *For every $0 < q < 1$ and $1 \leq \Delta_{in}, \Delta_{out} \leq n$ the SRS problem for the general model has a randomized (Las-Vegas) distributed algorithm requiring $O(\Delta_{in}\Delta_{out} \log \frac{n}{q})$ rounds with success probability $1-q$ on any n-vertex graph G.* ▮

Finally we consider deterministic algorithms for SRS. We make the assumption that processors have distinct identities $\{1, \ldots, n\}$. Under this assumption there is an obvious $O(n\Delta_{out})$ round algorithm where each edge is associated with a distinct round. We now present an $O\left(\Delta_{out}\Delta_{in}^2 \frac{\log^2 n}{\log(\Delta_{in} \log n)}\right)$ round deterministic algorithm.

As a consequence of the prime number theorem (cf. [D]) we have

Fact 2.1 *Let P be the set of all primes $p \leq x$. Then $\prod_{p\in P} p = e^{(1+o(1))x}$.*

Our algorithm relies on the following lemma.

Lemma 2.5 *Let $1 \leq x_1, \ldots, x_s \leq n$ be s distinct integers. Then for every $1 \leq i \leq s$ there exists a prime $p \leq s \log_2 n$ such that $(x_i \bmod p) \neq (x_j \bmod p)$ for every $j \neq i$.*

Proof: Fix i and consider the set P of all primes $p \leq s \log n$. Suppose that for each $p \in P$ there is an index j such that $(x_i \bmod p) = (x_j \bmod p)$. Then, denoting $a_j = |x_j - x_i|$, we have that $\prod_{1\leq j\leq n, j\neq i} a_j$ is divisible by each $p \in P$. Hence

$$n^s \geq \prod_{1\leq j\leq n, j\neq i} a_j \geq \prod_{p\in P} p.$$

By Fact 2.1 we get that

$$n^s \geq e^{(1+o(1))s \log_2 n} > n^s,$$

a contradiction. ▮

The algorithm operates in Δ_{out} phases. In phase i, each vertex x transmits its message M_i destined to its ith neighbor y_i. Let $\{p_1, \ldots, p_t\}$ be the set of primes in the range

$[2..(\Delta_{in}+1)\log n]$. (Note that $t = (1+o(1))\frac{(\Delta_{in}+1)\log n}{\log((\Delta_{in}+1)\log n)}$.) Each phase consists of t subphases where subphase j proceeds for p_j rounds and vertex x transmits M_i at time $(x \bmod p_j)$.

Lemma 2.6 *The above algorithm is correct.*

Proof: Let (x, y) be an edge in G where $y = y_i$ is the ith neighbor of x and let $z_1, \ldots, z_k$ be the other incoming neighbors of y. The set of vertices $x, y, z_1, \ldots, z_k$ is of cardinality at most $\Delta_{in} + 1$. By Lemma 2.5 there is a prime $p_j \leq (\Delta_{in}+1)\log n$ such that $(x \bmod p_j)$ is different from $(y \bmod p_j)$ as well as from $(z_\ell \bmod p_j)$ for $1 \leq \ell \leq k$. Now in round p_j of subphase j of phase i, x successfully transmits M_i to y. ▮

There are Δ_{out} phases each consists of at most $O\left(\frac{\Delta_{in}\log n}{\log(\Delta_{in}\log n)}\right)$ subphases. Each subphase consists of at most $(\Delta_{in}+1)\log n$ rounds. Therefore the overall complexity of this algorithm is $O\left(\Delta_{out}\Delta_{in}^2\frac{\log^2 n}{\log(\Delta_{in}\log n)}\right)$

Theorem 2.4 *For every $1 \leq \Delta_{in}, \Delta_{out} \leq n$ the SRS problem for the general model has deterministic distributed algorithms requiring $O(n\Delta_{out})$ or $O\left(\Delta_{out}\Delta_{in}^2\frac{\log^2 n}{\log(\Delta_{in}\log n)}\right)$ rounds on any n-vertex graph G with distinct processor identities $\{1, \ldots, n\}$.* ▮

3 The uniform model

We now consider the SRS primitive for the uniform model. That is, in the single round which we simulate, each processor x sends an identical message M_x to all of its outgoing neighbors.

3.1 Schedules

We first consider the existence of efficient schedules for the problem. The next obvious lower bound follows from the fact that each processor has to hear from Δ_{in} different processors.

Theorem 3.1 *For every graph G, any SRS schedule for the uniform model requires $\Omega(\Delta_{in})$ rounds.* ▮

Note that this lower bound is global in the sense that it holds for every graph. In contrast, the $\Omega(\Delta^2)$ lower bound of Theorem 2.2 for the simulation of the general model is true only for some particular graphs.

We can also prove a tight (but non-global) lower bound.

Theorem 3.2 *For infinitely many values of n there exist n-vertex graphs for which every SRS schedule requires $\Omega(\Delta_{in} \log \Delta)$ rounds.*

Proof: For $s \geq 2$, construct a directed bipartite graph $G = (V, U, E)$ with $V = \{1, \ldots, s\}$, $U = \{\{i, j\} | 1 \leq i < j \leq s\}$ and all directed edges $(i, \{i, j\})$ for every $1 \leq i \neq j \leq s$. In this graph $\Delta_{in} = 2$ and $\Delta_{out} = \Delta = s - 1$. Associate the edge $(i, \{i, j\})$ in G with the edge (i, j) in the complete graph on s vertices, K_s.

We now claim that for this graph, every SRS schedule needs $\Omega(\Delta_{in} \log \Delta) = \Omega(\log s)$ rounds. Notice that the set of edges in G, satisfied by a transmission round $T \subseteq V$, is associated with the cut $T \times (V - T)$ in K_s. This is true as vertices which transmit cannot receive any message and vice versa. The claim follows as the edges of K_s cannot be covered by fewer than $\lceil \log s \rceil$ cuts (cf. [Bo]). ▮

We remark that the above argument uses graphs in which Δ_{in} and Δ_{out} are considerably different. For the interesting special case of an undirected network (in which $\Delta = \Delta_{in} = \Delta_{out}$) the $\Omega(\Delta \log \Delta)$ lower bound still holds, although its proof becomes quite involved and is of independent interest. We have

Theorem 3.3 *There exists a constant $\epsilon > 0$, such that for every $\Delta > 1$ and for each sufficiently large n with $n\Delta$ even, there exist n-vertex Δ-regular undirected graphs for which every SRS schedule requires at least $\epsilon \Delta \log \Delta$ rounds.*

The proof of this theorem is deferred to the Appendix.

We now prove that for every graph there exists a simulation schedule of $r = O(\Delta_{in} \log \Delta)$ rounds. In order to prove the existence of the desired schedule it suffices to show that on any given graph, applying the procedure `Random_Transmit`(M_x, r) at every vertex for r rounds, where r is as above, succeeds with positive probability.

The proof is based on the Lovász Local Lemma ([EL], cf. [S]). Let $A_1, \ldots, A_n$ be events in a probability space. A graph H on the vertices $\{1, \ldots, n\}$ (the indices for the A_i) is called a *dependency graph* for $A_1, \ldots, A_n$ if for all i the event A_i is mutually independent of all A_j with $(i, j) \notin H$.

Lemma 3.1 (EL) *Assume that for all i, $\Pr(A_i) \leq p$ and let d be the maximum degree of* *vertices in H. If $4dp \leq 1$ then*

$$\Pr\left(\bigcap_{i=1}^{n} \bar{A}_i\right) > 0.$$

▮

For every directed edge (x, y) in the network G the event A_{xy} is defined as in the previous section: "the processor y fails to receive M_x over the edge e_{xy} during all r rounds".

Lemma 3.2 *There is a dependency graph H for these events with maximum degree $d \leq 2\Delta^3$.*

Proof: For all (x, y) the event A_{xy} is independent of all A_{vw} where (v, w) is an edge at distance at least four from the edge (x, y). (Distance is measured in the underlying graph of G and two incident edges are at distance one.) The lemma follows as there are at most $2\Delta^3$ directed edges at distance one or two from (x, y). ▮

Lemma 3.3 *Applying Procedure* `Random_Transmit`(M_x, r) *at every vertex for*

$$r = \left\lceil 2e\Delta_{in} \ln(9\Delta^3) \right\rceil$$

rounds succeeds in transmitting on all the edges with positive probability.

Proof: Consider the dependency graph of the events defined above. By Lemmas 3.2 and 2.3 and by the choice of r we have that

$$4dp \leq 8\Delta^3 exp\left(-\frac{r}{2e\Delta_{in}}\right) < 1.$$

Hence by Lemma 3.1

$$\Pr\,(\bigcap_{i=1}^{n} \bar{A}_i) > 0. \qquad ▮$$

In other words, Lemma 3.3 states that with a positive probability all the edges are satisfied by some transmission and hence we get the following theorem.

Theorem 3.4 *For every directed graph G there exists an SRS schedule for the uniform model of $O(\Delta_{in} \log \Delta)$ rounds.* ▮

As is the case with other instances where the Lovász Local Lemma is used we are not able to constructively find a schedule of $O(\Delta_{in} \log \Delta)$ rounds. Rather we describe a construction of a schedule with $O(\Delta_{in} \log n)$ rounds. Note that since every schedule for the general model is also a schedule for the uniform model we already have a constructive schedule of $O(\Delta_{in}\Delta_{out})$ rounds for the uniform case.

The schedule is constructed by a doubly iterative process. On the highest level, the schedule is constructed sequentially round by round. For each round i select a set of transmitters T_i by an internal iterative process. Suppose that T_j is already constructed for $1 \le j \le i$. Denote by S_i the set of edges *satisfied* in one of the first i rounds and by $F_i = E - S_i$ the set of *failed* edges (x, y), i.e., such that M_x still needs to be received by y. Initially $F_0 = E$ and $S_0 = \emptyset$. The construction process continues until a round i when $S_i = E$.

We now describe the internal iterative procedure for constructing the transmission set T_{i+1} of round $i + 1$. Let $F = F_i$ and $S = S_i$. For every processor x and for every set $W \subseteq V$ denote by $f_W(x)$ (respectively, $s_W(x)$) the number of edges (x, z) for $z \in W$ belonging to F (respectively, S). Note that the total number of edges pointing to W is $\sum_{x \in V}(f_W(x) + s_W(x)) \le \Delta_{in}|W|$.

Throughout the construction the set V of processors is partitioned into four groups:

(1) T – The transmitters

(2) H – Processors with exactly one incoming neighbor in T. These processors will hear a message (and contribute an edge to S) if the processors in T transmit.

(3) C – Processors having at least two incoming neighbors in T, they hear no message if the processors in T transmit. Also, processors having one incoming edge from T which already belongs to S.

(4) R – The rest of the processors.

Initially $T, H, C = \emptyset$ and $R = V$.

Call a processor x *useful* if it satisfies one of the following conditions:

1. $x \in H$ and $f_R(x) > 2(f_H(x) + s_H(x) + 1)$,
2. $x \in C \cup R$ and $f_R(x) > 2(f_H(x) + s_H(x))$.

Note that a useful processor can never belong to T because processors in T have no outgoing neighbors in R. Intuitively, a useful processor is a processor whose addition to T will increase the number of satisfied edges while maintaining some invariants needed for the analysis of the algorithm.

In each step we select a useful processor x from $H \cup C \cup R$, transfer it to T, and change the sets H, C and R accordingly. Repeat this selection process as long as such a vertex can be found. Once no processor is useful, let $T_{i+1} = T$ and start constructing the next round.

Lemma 3.4 *The invariants*

$$|T| \leq |H| \quad \textit{and} \quad |C| \leq |H|$$

are maintained by the construction procedure described above.

Proof: The proof is by induction on the number of iterations in the construction of T. The base case is trivial since $|T|, |C|, |H| = 0$. Assume that the claim holds after j selection steps and that x is chosen in step $j + 1$ to be transferred to T. Let H', C' and T' be the new sets of processors and denote by a the number of new processors that were added to H (from R) and by b the number of processors that were removed from H (mostly to C, except for $x \in H$ which is moved to T).

The definition of a useful processor implies that $a > 2b$. Therefore,

$$|H'| = |H| + a - b \geq |H| + b + 1.$$

By the inductive hypothesis and the above inequality,

$$|H'| \geq |H| + 1 \geq |T| + 1 = |T'|,$$

and

$$|H'| > |H| + b \geq |C| + b \geq |C'|.$$

▮

Lemma 3.5 *The construction of* T *proceeds as long as* $|H| < \frac{|F|}{5\Delta_{in}}$.

Proof: Assume to the contrary that $|H| < \frac{|F|}{5\Delta_{in}}$ and yet no processor $x \notin T$ is useful, i.e., every $x \in R \cup C$ satisfies

$$f_R(x) \leq 2(f_H(x) + s_H(x))$$

and every $x \in H$ satisfies

$$f_R(x) \leq 2(f_H(x) + s_H(x) + 1).$$

Summing these inequalities over all the processors in $R \cup C \cup H$ implies

$$\sum_{x \in V-T} f_R(x) \leq 2 \sum_{x \in V-T} (f_H(x) + s_H(x)) + 2|H|.$$

The edges in F are classified according to the sets into which they point. This implies that the cardinality of F is

$$\sum_{x \in V-T} f_R(x) + \sum_{x \in V} f_T(x) + \sum_{x \in V} f_H(x) + \sum_{x \in V} f_C(x) \leq \sum_{x \in V-T} f_R(x) + \Delta_{in}|H \cup C \cup T|.$$

(The first summation does not include processors in T since there are no F edges from T to R.) This inequality and Lemma 3.4 imply that

$$\sum_{x \in V-T} f_R(x) \geq |F| - \Delta_{in}(|H| + |C| + |T|) \geq |F| - 3\Delta_{in}|H|.$$

Since there is exactly one incoming edge from T into each vertex of H, and this edge is in F, $|H| = \sum_{x \in T} f_H(x)$. Using this fact and bounding by Δ_{in} we get

$$2 \sum_{x \in V-T} (f_H(x) + s_H(x)) + 2|H| \leq 2 \sum_{x \in V} (f_H(x) + s_H(x)) \leq 2\Delta_{in}|H|.$$

Combining the inequalities we get $|F| - 3\Delta_{in}|H| \leq 2\Delta_{in}|H|$ and hence,

$$|H| \geq \frac{|F|}{5\Delta_{in}},$$

contradicting the assumption. ▮

We remark that an alternative procedure for the internal selection step can be derived by using the method of conditional probabilities (cf. [S; Lecture 4]). This is done by observing that if we choose every vertex, independently, with probability $1/\Delta_{in}$ to transmit, then the expected number of edges satisfied is about a fraction $1/(e\Delta_{in})$. Since this estimate concerns only expected values, it is possible to de-randomize this algorithm and find an appropriate set of transmitters deterministically. This is done by considering the vertices one by one, and deciding for each whether to choose it or not, so as to maximize the conditional expectation of the number of edges satisfied given the specific choices made so far and assuming that every vertex to be considered later will be chosen with probability $1/\Delta_{in}$ independently.

Theorem 3.5 *There is a (polynomial time constructible) SRS schedule for the uniform model of only $r = O(\Delta_{in} \log n)$ rounds.*

Proof: Let F_i denote the set of unsatisfied edges at the end of round i. Lemma 3.5 implies that in round i the transmission succeeded on at least $\frac{|F_{i-1}|}{5\Delta_{in}}$ edges. Hence for every $i \geq 1$,

$$|F_i| \leq \left(1 - \frac{1}{5\Delta_{in}}\right) |F_{i-1}|,$$

so

$$|F_i| \leq \left(1 - \frac{1}{5\Delta_{in}}\right)^i |E| \leq \left(1 - \frac{1}{5\Delta_{in}}\right)^i \Delta_{in} n.$$

Therefore after at most

$$r = \lceil 5\Delta_{in} \ln(\Delta_{in} n) \rceil = O(\Delta_{in} \log n)$$

rounds $|F_r|$ is bounded by a constant. Now, $O(1)$ additional rounds, one per each unsatisfied edge, all edges are satisfied. ▮

3.2 Distributed algorithms

We now consider a distributed randomized algorithm for SRS. We may apply Procedure `Random_Transmit`(M_x, r) (for a sufficiently large number of rounds) as a randomized algorithm for the problem, with any desired success probability.

Theorem 3.6 *For every $0 < q < 1$ and $1 \leq \Delta_{in}, \Delta_{out} \leq n$ the SRS problem for the uniform model has a randomized (Las-Vegas) distributed algorithm requiring $O(\Delta_{in} \log \frac{n}{q})$ rounds with success probability $1 - q$ on any n-vertex graph G.*

Proof: From Lemma 2.3 and the fact that $|E| \leq \Delta n$ we get that the probability to fail on at least one of the edges is

$$\Pr\left(\bigcup_{(x,y)\in E} A_{xy}\right) \leq \sum_{(x,y)\in E} \Pr(A_{xy}) \leq \Delta n e^{-\frac{r}{2e\Delta_{in}}}.$$

A simple calculation shows that for

$$r = \left\lceil 2e\Delta_{in} \ln \frac{\Delta n}{q} \right\rceil$$

this probability is less than q. ▮

Deterministic algorithms for SRS can again be designed under the assumption that processors have distinct identities $\{1, \ldots, n\}$. The algorithms presented for the general model can be modified to run faster in the uniform model, saving a factor of Δ_{out} as each processor has only one message to transmit.

Theorem 3.7 *For every $1 \leq \Delta_{in}, \Delta_{out} \leq n$ the SRS problem for the uniform model has deterministic distributed algorithms requiring $O(n)$ or $O\left(\Delta_{in}^2 \frac{\log^2 n}{\log(\Delta_{in} \log n)}\right)$ rounds on any n-vertex graph G with distinct processor identities $\{1, \ldots, n\}$.* ▮

Acknowledgments

We thank Shay Kutten and Alex Schäffer for working with us in early stages of this research, and Eli Upfal and Avi Wigderson for helpful discussions.

References

[ABLP] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A lower Bound for Radio Broadcast, *J. of Computer and System Sciences*, to appear.

[BGI] R. Bar-Yehuda, O. Goldreich and A. Itai, On the Time-Complexity of Broadcast in Radio Networks: An Exponential Gap between Determinism and Randomization, *Proc. 5th ACM Symp. on Principles of Distributed Computing*, (1986), pp. 98–107.

[Bo] B. Bollobás, *Combinatorics*, Cambridge University Press, Cambridge, 1986.

[CK1] I. Chlamtac and S. Kutten, On Broadcasting in Radio Networks – Problem Analysis and Protocol Design, *IEEE Trans. on Commun.* **33**, (1985), 1240–1246.

[CK2] I. Chlamtac and S. Kutten, Tree Based Broadcasting in Multihop Radio Networks, *IEEE Trans. on Computers* **36**, (1987), pp. 1209–1223.

[CK3] I. Chlamtac and S. Kutten, A Spatial Reuse TDMA/FDMA for Mobile Multi-hop Radio Networks, *Proc. IEEE INFOCOM*, 1985, Washington DC, pp. 389–394.

[CW] I. Chlamtac and O. Weinstein, The Wave Expansion Approach to Broadcasting in Multihop Radio Networks, *Proc. INFOCOM*, 1987, pp. 874–881.

[D] H. Davenport, *Multiplicative Number Theory*, 2nd ed. revised by H. L. Montgomery, Springer Verlag, Berlin, 1980.

[EL] P. Erdös and L. Lovász, Problems and Results on 3-chromatic Hypergraphs and Some Related Questions, in *Infinite and Finite Sets*, North-Holland, 1975.

[EGMT] S. Even, O. Goldreich, S. Moran and P. Tong, On the NP-Completeness of Certain Network Testing Problems, *Networks* **14**, (1984), pp. 1–24.

[HLR] J. Hastad, T. Leighton and B. Rogoff, Analysis of Backoff Protocols for Multiple Access Channels, *19th ACM Symp. on Theory of Computing*, 1987, pp. 241–253.

[GFL] A. G. Greenberg, P. Flajolet and R. E. Ladner, Estimating the Multiplicities of Conflicts to Speed their Resolutions in Multiple Access Channels, *J. ACM* **34**, (1987), pp. 289–325.

[GGMM] J. Goodman, A. G. Greenberg, N. Madras and P. March, On the Stability of Ethernet, *Proc. 17th Symp. on Theory of Computing,* (1985), pp. 379–387.

[GVF] I. Gitman, R. M. Van Slyke and H. Frank, Routing in Packet-Switching Broadcast Radio Networks, *IEEE Trans. on Commun.*, (1976), pp. 926–930.

[KGBK] R. E. Kahn, S. A. Gronemeyer, J. Burchfiel and R. C. Kunzelman, Advances in Packet Radio Technology, *Proc. IEEE* **66**, (1978).

[K] R. C. Kunzelman, Overview of the ARPA Packet Radio Experimental Network, *Proc. COMPCON*, 1978, pp. 157–160.

[SC] N. Shacham and E. J. Craighill, Dynamic Routing for Real-Time Transport in Packet Radio Networks, *Proc. INFOCOM*, 1982.

[S] J. Spencer, *Ten Lectures on the Probabilistic Method*, *SIAM*, Philadelphia, 1987.

A Proof of the $\Omega(\Delta \log \Delta)$ lower bound

In this appendix we prove Theorem 3.3, providing the lower bound of $\Omega(\Delta \log \Delta)$ on the length of an SRS schedule for an undirected graph in the uniform model. Here, $\Delta = \Delta_{in} = \Delta_{out}$.

We first state the problem in a different way (see also [ABLP] for a similar formulation). Let G be a bipartite graph with vertex classes A and B. If X is a subset of A and (a, b) is an edge of G with $a \in A$ and $b \in B$ we say that X *hits* (a, b) if $N_G(b) \cap X = \{a\}$ where $N_G(b)$ is the set of all neighbors of b in A. We say that a family of subsets $\mathcal{F} = \{X_1, \ldots, X_\ell\}$ of A hits (a, b) if there is an i, $1 \leq i \leq \ell$, such that X_i hits (a, b). $\mathcal{F}$ *hits* G if it hits all the edges of G. Let $h(G)$ denote the minimum cardinality of a family $\mathcal{F}$ that hits G.

By an application of the local lemma it was shown in Theorem 3.4 that for every graph G with maximum degree Δ, $h(G) \leq O(\Delta \log \Delta)$. We show here that there is an $\epsilon > 0$ such that for every $\Delta \geq 1$ there is a bipartite graph G with maximum degree Δ such that $h(G) \geq \epsilon \Delta \log \Delta$. Therefore, the maximum possible value of $h(G)$, as G ranges over all graphs with maximum degree Δ is $\Theta(\Delta \log \Delta)$.

Our proof is probabilistic. Let A and B be two disjoint sets, $|A| = |B| = n$. Let $\epsilon > 0$ be a small real number (e.g., $\epsilon = 10^{-3}$) and put $p = 1/n^\epsilon$. Let G be the random bipartite graph on the vertex classes A and B in which for each $a \in A$, $b \in B$ independently (a, b) is chosen to be an edge with probability p. To complete the proof we show that almost surely (i.e., with probability that tends to 1 as n tends to ∞), the maximum degree of G is at most $\frac{3}{2}n^{1-\epsilon}$ and $h(G) > \epsilon n^{1-\epsilon} \log n$.

For convenience we split the proof into several lemmas. Whenever it is needed, we assume that n is sufficiently large. Throughout the proof all logarithms are in the natural basis e. We omit all the integral part signs whenever it is clear those can be omitted.

Lemma A.1 *The probability that the maximum degree of G exceeds $\frac{3}{2}n^{1-\epsilon}$ is at most $2ne^{-\frac{n^{1-\epsilon}}{16}}$ (and in particular tends to 0 as n tends to infinity).*

Proof: The degree of every vertex of G is a binomial random variable with parameters n and $p = 1/n^\epsilon$. By the standard estimates for binomial distributions (see, e.g., [S]), the probability that this degree exceeds its expectation by y is at most $e^{-\frac{y^2}{2(np)} + \frac{y^3}{2(np)^2}}$. The result now follows by taking $y = \frac{np}{2} = \frac{1}{2}n^{1-\epsilon}$ and by observing that G has $2n$ vertices. ▮

Lemma A.2 *Almost surely G satisfies the following property: for every $A' \subseteq A$ and $B' \subseteq B$ with $|A'| = |B'| = 10n^\epsilon \log n$, the induced subgraph of G on $A' \cup B'$ has more than $10n^\epsilon \log n$ edges.*

Proof: The probability that there are A' and B' that violate the above claim is at most

$$\binom{n}{10n^\epsilon \log n}\binom{n}{10n^\epsilon \log n}\binom{100n^{2\epsilon}\log^2 n}{10n^\epsilon \log n} \cdot (1-p)^{100n^{2\epsilon}\log^2 n - 10n^\epsilon \log n}$$

$$\leq n^{30n^\epsilon \log n} e^{-90n^\epsilon \log^2 n} \quad \longrightarrow_{n\to\infty} \; 0. \quad \blacksquare$$

Corollary A.1 *Almost surely there is no set $X \subseteq A$ of cardinality $|X| \geq 10n^\epsilon \log n$ that hits at least $10n^\epsilon \log n$ edges of G.*

Proof: If there is such an X, put $Y = \{b \in B\colon X$ hits at least one (and hence exactly one) edge incident with $b\}$. Then $|Y| \geq 10n^\epsilon \log n$. Let $A' \subseteq X$ be an arbitrary subset of cardinality $10n^\epsilon \log n$ of X and let $B' \subseteq Y$ be an arbitrary subset of cardinality $10n^\epsilon \log n$ of Y. Since $|N_G(b) \cap X| = 1$ for all $b \in B'$, $|N_G(b) \cap A'| \leq 1$ for all $b \in B'$ and hence the induced subgraph of G on $A' \cup B'$ has at most $|B'| = 10n^\epsilon \log n$ edges. Thus Lemma A.2 implies Corollary A.1. $\blacksquare$

Our objective is to prove that almost surely $h(G) \geq \epsilon n^{1-\epsilon} \log n$. Suppose this is false and let $\mathcal{F} = \{X_1, \ldots, X_\ell\}$ be a family of subsets of A that hits G, where $\ell \leq \epsilon n^{1-\epsilon} \log n$. By the last corollary, the subfamily of $\mathcal{F}$ consisting of all sets X_i whose cardinality is at least $10n^\epsilon \log n$ almost surely hits at most $\ell \cdot 10n^\epsilon \log n < n \log^2 n$ edges of G. Therefore, in order to complete the proof it suffices to prove the following:

Proposition A.1 *Almost surely, for every family $\mathcal{H} = \{X_1, \ldots, X_\ell\}$ of subsets of A, where $\ell \leq \epsilon n^{1-\epsilon} \log n$ and $|X_i| \leq 10n^\epsilon \log n$ for all i, there are more than $n \log^2 n$ edges of G that are not hit by $\mathcal{H}$.*

In order to prove this proposition, which is the main part of the argument, we prove several additional lemmas.

Lemma A.3 *Let $\mathcal{H} = \{X_1, \ldots, X_\ell\}$ be a family of subsets of $A = \{1, 2, \ldots, n\}$, where $\ell \leq \epsilon n^{1-\epsilon} \log n$ and $|X_i| \leq 10n^\epsilon \log n$ for all i, $1 \leq i \leq \ell$. Then there is a subset $A' \subseteq A$, $|A'| = n/2$ such that each $a \in A'$ belongs to at most $\log^2 n$ subsets in $\mathcal{H}$ and for no $1 \leq i \leq \ell$, $|X_i \cap A'| = 1$.*

Proof: The average number of subsets in $\mathcal{H}$ containing an element of A is at most

$$\frac{\ell \cdot 10n^{\epsilon} \log n}{n} \leq 10\epsilon \log^2 n \leq \frac{1}{3} \log^2 n.$$

Thus there is an $A'' \subset A$ containing at least two thirds of the points of A each member of which belongs to at most $\log^2 n$ subsets in $\mathcal{H}$. Omitting repeatedly (while modifying A'') the points a with $\{a\} = X_i \cap A''$ for some $1 \leq i \leq \ell$ we complete the proof. ▮

Lemma A.4 *For any real p, $0 < p < 1$ and any integer k, $k \geq 2$,*

$$[1 - (1-p)^{k-1}]^k \geq e^{-2/p}.$$

Proof: Since

$$(1-p)^{k-1} \leq e^{-p(k-1)} = \frac{1}{e^{p(k-1)}} \leq \frac{1}{p(k-1)+1}$$

we have

$$1 - (1-p)^{k-1} \geq 1 - \frac{1}{p(k-1)+1} = \frac{1}{1 + \frac{1}{p(k-1)}}.$$

Therefore,

$$[1 - (1-p)^{k-1}]^k \geq \frac{1}{(1 + \frac{1}{p(k-1)})^k} \geq \frac{1}{e^{k/p(k-1)}} = \left[\frac{1}{e^{k/(k-1)}}\right]^{1/p} \geq e^{-2/p}. \quad ▮$$

Lemma A.5 *Let A' be a set, $|A'| = n/2$. Let $\mathcal{H}'$ be a family of at most $\epsilon n^{1-\epsilon} \log n$ subsets of A', each subset having at least two elements. For each $a \in A'$, define a weight*

$$w(a) = \prod_{F,\ a \in F \in \mathcal{H}'} [1 - (1-p)^{|F|-1}], \quad \textit{where } p = 1/n^{\epsilon}.$$

Then

$$\prod_{a \in A'} w(a) \geq \left(\frac{1}{e^2}\right)^{\epsilon n \log n} = \frac{1}{n^{2\epsilon n}}.$$

Proof: We have

$$\begin{aligned}
\textstyle\prod_{a \in A'} w(a) &= \textstyle\prod_{a \in A'} \prod_{F,\ a \in F \in \mathcal{H}'} (1 - (1-p)^{|F|-1}) = \prod_{F \in \mathcal{H}'} \prod_{a \in F} (1 - (1-p)^{|F|-1}) \\
&= \textstyle\prod_{F \in \mathcal{H}'} (1 - (1-p)^{|F|-1})^{|F|} \\
&\geq \textstyle\prod_{F \in \mathcal{H}'} (e^{-2/p}) \qquad \text{(by Lemma A.4)} \\
&\geq e^{-2n^{\epsilon} \cdot \epsilon n^{1-\epsilon} \log n} = e^{-2\epsilon n \log n} = n^{\frac{1}{2\epsilon n}}. \qquad ▮
\end{aligned}$$

Lemma A.6 *Let $A'' \subset A'$ be two sets, $|A''| = n/4$, $|A'| = n/2$ and let $\mathcal{H}'$ be a family of at most $\epsilon n^{1-\epsilon} \log n$ subsets of A', each of cardinality at most $10n^{\epsilon} \log n$. Suppose, further, that no member of A' belongs to more than $\log^2 n$ subsets in $\mathcal{H}'$. Then there is a subset J of A'' satisfying*

(i) $|J| \geq \frac{n}{400 n^{2\epsilon} \log^6 n}$

(ii) *For every two distinct j, $j' \in J$ and for every (not necessarily distinct) F, $F' \in \mathcal{H}'$ with $j \in F$ and $j' \in F'$ we have $|F \cap F'| = \emptyset$. (Thus, in particular, there is no $F \in \mathcal{H}'$ containing both j and j').*

Proof: We construct J greedily. Choose an arbitrary element of A'' as j_1 and omit from A' the set J_1 of all the elements of A' contained in one of the edges containing j_1, and all the elements of A' contained in an edge containing one of the elements of J_1. Since by the assumptions we get $|J_1| \leq \log^2 n \cdot 10 n^\epsilon \log n$, at most $(10 n^\epsilon \log^3 n) \log^2 n \cdot 10 n^\epsilon \log n = 100 n^{2\epsilon} \log^6 n$ elements are omitted in this manner. We now pick j_2 among the elements left in A'', omit all the elements of A' of distance at most 2 (in the hypergraph $\mathcal{H}'$) from j_2 and continue in the same way. The result clearly follows. ▮

Returning now to our random bipartite graph G on the classes of vertices A and B, where $|A| = |B| = n$, let us fix a family $\mathcal{H} = \{X_1, \ldots, X_\ell\}$ of subsets of A, where $\ell \leq \epsilon n^{1-\epsilon} \log n$ and $|X_i| \leq 10 n^{\epsilon \log n}$ for all $1 \leq i \leq \ell$. In order to prove Proposition A.1 we show that the probability that for this fixed $\mathcal{H}$, $\mathcal{H}$ hits all but at most $n \log^2 n$ edges of G is extremely small. By Lemma A.3 there is an $A' \subseteq A$, $|A'| = n/2$ satisfying the conclusions of Lemma A.3. Define $\mathcal{H}' = \{X_i \cap A' : \ X_i \cap A' \neq \emptyset, \ 1 \leq i \leq \ell\}$. For each $a \in A$ define $w(a)$ as in Lemma A.5. By the assertion of this lemma

$$\prod_{a \in A'} w(a) \geq \frac{1}{n^{2\epsilon n}}.$$

Since $w(a) \leq 1$ for all a this implies that there is a set $A'' \subset A'$, $|A''| \geq \frac{n}{4}$ such that for each $a \in A''$, $w(a) \geq \frac{1}{n^{8\epsilon}}$. Applying Lemma A.6 for $A'' \subset A'$ and $\mathcal{H}'$ we get a set J satisfying the conclusions of this lemma. Consider, now, a fixed vertex $b \in B$ and a fixed element $j \in J$. Let $E_1(b,j)$ be the event that (b,j) is an edge of G. Let $E_2(b,j)$ be the event that b has a neighbor in every set $F \setminus \{y\}$ for all sets $F \in \mathcal{H}'$ that satisfy $j \in F$. Notice that the two events E_1 and E_2 are independent and that if they both occur then $\mathcal{H}$ fails to hit the edge (b,j). Note also that $\Pr(E_1(b,j)) = p = 1/n^\epsilon$ and that by the FKG-Inequality (see, e.g., [Bo])

$$\begin{aligned} \Pr(E_2(b,j)) &\geq \textstyle\prod_{F \in \mathcal{H}', \ j \in F} (b \quad \text{has a neighbor in } F - \{j\}) \\ &= \textstyle\prod_{F \in \mathcal{H}', \ j \in F} (1 - (1-p)^{|F|-1}) = w(j) \geq \frac{1}{n^{8\epsilon}}, \end{aligned}$$

where the last inequality holds since the weight $w(j)$ of each element of A'' (and in particular of each element of J) is at least $\frac{1}{n^{8\epsilon}}$. Therefore, if we denote by $E_3(b,j)$ the event that both

$E_1(b,j)$ and $E_2(b,j)$ occur (and hence $\mathcal{H}$ fails to hit (b,j)), then

$$\Pr(E_3(b,j)) \geq \frac{1}{n^{\epsilon}} \cdot \frac{1}{n^{8\epsilon}} = \frac{1}{n^{9\epsilon}}.$$

Moreover, since J satisfies the assertion of Lemma A.6 part (ii), we conclude that all the events $E_1(b,j)$ and $E_2(b,j)$, as j ranges over J and b ranges over B are mutually independent. We have thus proved that for each fixed family $\mathcal{H}$ satisfying the assumptions of Proposition A.1, when G is taken randomly as before, the random variable Y which counts the number of edges of G which are not hit by $\mathcal{H}$ satisfies the following: for each number s, the probability that Y is at least s is at least the probability that a binomial random variable with parameter $N = |B| \cdot |J| \geq \frac{n^2}{400n^{2\epsilon}\log^{6n}}$ and $p = \frac{1}{n^{9\epsilon}}$ is at least s. By applying the standard estimates for Binomial distributions ([S]) we obtain the following:

Lemma A.7 *For every fixed $\mathcal{H}$ satisfying the assumptions of Proposition A.1, the probability that $\mathcal{H}$ hits all but at most $\frac{n^2}{10000n^{11\epsilon}\log^6 n}$ edges of G (and certainly the probability that it hits all but $n\log^2 n$ of these edges) is smaller than $e^{-\frac{n^2}{10000n^{11\epsilon}\log^{6n}}}$.* ▮

Since there are less than

$$\left(\sum_{i=0}^{10n^{\epsilon}\log n} \binom{n}{i}\right)^{\epsilon n^{1-\epsilon}\log n} \leq n^{10\epsilon n\log^2 n} < e^{n\log^3 n}$$

possible choices for the family $\mathcal{H}$ this implies the assertion of Proposition A.1 and completes the proof of our main result, namely:

Theorem A.1 *There exists an $\epsilon > 0$ such that for all sufficiently large n there is a graph G with maximum degree $\Delta < \frac{3}{2}n^{1-\epsilon}$ such that $h(G) \geq \epsilon n^{1-\epsilon}\log n \geq \epsilon\Delta\log\Delta$.* ▮

Remark 1: Clearly if H is a subgraph of G then $h(G) \geq h(H)$. This, together with Theorem A.1 and an adjustment of ϵ enables one to prove:

Theorem A.2 *There exists an $\epsilon > 0$ such that for every $\Delta \geq 1$ there is a Δ-regular (bipartite) graph G satisfying $h(G) \geq \epsilon\Delta\log\Delta$.*

Remark 2: The parameters in the proof above, (i.e., the size of A, that of B, the probability p and the two epsilons in the expression $\epsilon n^{1-\epsilon}\log n$ (which can differ from each other)), can be adjusted to get lower bounds for $h(G)$ for graphs G with maximum indegree d_1 and maximum outdegree d_2. The interested reader can make the required computation.