On the Complexity of Radio Communication

(Extended Abstract)

Noga Alon * Amotz Bar-Noy † Nathan Linial ‡ David Peleg §

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. This paper is concerned with lower and upper bounds for the complexity of realizing various communication primitives for radio networks.

Our first result deals with the *broadcast* operation. We prove the existence of a family of radius-2 networks on n vertices for 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 earlier by Bar-Yehuda, Goldreich and Itai [BGI]. It is worth mentioning that this lower bound holds even under optimal centralized coordination, while the (randomized) algorithm of [BGI] is distributed.

We then look at the question of simulating two of the standard message-passing models on a radio network. Both 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.

Finally we give tight bounds for the length of schedules for computing census functions on a radio network.

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, CK, 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

*Tel-Aviv University, Sackler Faculty of Exact Sciences. Supported in part by an Allon Fellowship, by a Bat-Sheva de Rothschild grant and by a Bergmann Memorial grant.

†Stanford University. Supported in part by a Weizmann fellowship, and by contract ONR N0014-85-C-0731.

‡IBM Almaden Research Center, Stanford University and the Hebrew University of Jerusalem.

§Weizmann Institute and Stanford University. Supported in part by contract ONR N00014-85-C-0731 and by a grant of Stanford Center for Integrated Systems.

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. A major goal of this article is to point out more fundamental reasons for the difficulty of radio communication. In particular, we study the time required for some of the most basic functions usually performed on networks (such as broadcasting a message from a single source to all other stations), and give lower bounds which hold even if the processors are centrally coordinated and have complete knowledge of the graph.

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.

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. Much of this paper concerns simulations between the standard synchronous point-to-point message-passing model and the radio model. (This is one case of a more general program to study the relative power of various communication modes by mutual simulation.) 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.)

Now let us start discussing specific results. For the broadcast problem in radio networks, Bar-Yehuda et al. [BGI] present a randomized distributed algorithm requiring $O(D \log n + \log^2 n)$ rounds, where D is the diameter of the graph. This implies the existence of a schedule of the same length. (Their algorithm is described for undirected graphs, but can be readily extended to handle also the directed case.) The deterministic centralized (polynomial time) algorithm of [CW] constructs a schedule of length $O(D \log^2 n)$. In this paper we demonstrate the existence of a family of radius-2 networks on n vertices for which the number of rounds required by any broadcast schedule is $\Omega(\log^2 n)$, which is optimal in view of [BGI]. The existence of such a family is established by probabilistic arguments, including the FKG-Inequality.

It should be noted that both our lower bound and the $\Omega(n)$ deterministic lower bound of [BGI] fail if one assumes that messages may be broken down and split to bits. In such a case, the following algorithm is possible (we state it for radius 2, though the general case is just as simple). At time 0 the *sender* s passes its broadcast message M to all its neighbors. In the next $length(M)$ rounds, each neighbor of s transmits the message bit by bit as follows: At time i (for $1 \leq i \leq length(M)$), it transmits a 1 if the i-th bit of M is 1, and remains silent otherwise. Since transmissions are synchronous, the stations at distance 2 from s receive the message by interpreting a 1 message or noise as a 1 bit and silence as a 0 bit. It is easy to extend this scheme into an $O(length(M) + D)$ time broadcast algorithm for general graphs. Notice that this method is of little practical significance, because any external noise would fail it. In order to withstand noise and to achieve synchronization, actual radio networks enclose messages by sequences of control bits. Such a high overhead cost per bit sent in this method renders it completely useless, except if M is extremely short. It is also a sensible requirement in the design of radio networks that messages are not to be meddled with by intermediate stations and should be forwarded as a whole, possibly by a special purpose message controller that stations have. In this paper we adopt the assumption that messages are transmitted as they are. Alternatively, our lower bound holds also when messages are long enough to render this method useless.

We cannot rule out the intriguing possibility that an $O(D + \log^2 n)$ schedule always exists. This may seem a surprising possibility in view of the $\Omega(\log^2 n)$ time that may be required to pass the message to all vertices at distance 2 from the origin. Would it not take a similar amount of time to proceed to those of distance 3 etc...? Let V_i be all the vertices at distance i from the sender s. There is a possibility that through some clever pipelining the network may be engaged in passing the message from V_i to V_{i+1} at the same time that it deals with V_j and V_{j+1} for some $j > i + 1$. How efficiently this may be done we do not know. It is worthwhile to try and understand if this is indeed possible.

Our main results for the single-round simulation problem are as follows. 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.

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 $O(\Delta_{out}\Delta_{in}^2 \log^2 n)$ 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. 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.

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.

The last problem we study is that of schedules for computing *census functions* on a radio network. The problem is defined as follows (taking addition as our example). Each vertex x of G has an input value v_x. The goal is to sum all the values to a designated vertex p in G.

We give a schedule of $O(D\Delta_{in})$ rounds for the census problem, where D denotes the diameter of G. We show that this result is tight by constructing a graph in which every census schedule requires at least $\Omega(D\Delta_{in})$ rounds.

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

2 A lower bound for broadcast

In a schedule $S = (T_1, T_2, \ldots)$ for broadcast, the transmissions T_i need only specify the transmitting processors, not the messages. In step i, every processor $v \in T_i$ is assumed to already hold a copy of the original message M, and is required to transmit it. The schedule S is a *broadcast schedule* for the sender s in G if after applying S, every processor in the network has a copy of M.

Let $G = (V, E)$ be a radius-2 graph, denote by V_1 (V_2) the set of all the vertices at distance one (two) from the sender s. For convenience, we assume $V_1 = N = \{1, 2, \ldots, n\}$. After the first round of any broadcasting schedule in G all the processors in V_1 have the message M. Therefore, the remaining rounds of any schedule only need to guarantee the arrival of M to all processors in V_2. The graphs considered here have no edges between vertices in V_2, and so existence of a $t-$round schedule can be cast in combinatorial terms as follows.

Let $\mathcal{H}$ and $\mathcal{F}$ be families of nonempty subsets of N. (Any $H \in \mathcal{H}$ is the set of neighbors of some vertex in V_2. Members of $\mathcal{F}$ are transmissions in the schedule.) We say that $F \in \mathcal{F}$ *hits* $H \in \mathcal{H}$ if $|F \cap H| = 1$. (This means that the vertex in V_2 corresponding to H got the message on the transmission corresponding to F.) Also $\mathcal{F}$ *hits* H if some $F \in \mathcal{F}$ does, and $\mathcal{F}$ hits $\mathcal{H}$ if it hits every $H \in \mathcal{H}$. Let $t(\mathcal{H})$ be the minimum cardinality of $\mathcal{F}$ which hits $\mathcal{H}$. (The shortest broadcast schedule.) Define $t(n) = max\{t(\mathcal{H})\}$ over all $\mathcal{H}$ of n subsets of N. The problem is to determine, or estimate $t(n)$.

We determine $t(n)$ up to a constant factor.

Theorem 2.1 There are two positive constants c_1, c_2 such that $c_1 \log^2 n \leq t(n) \leq c_2 \log^2 n$ for all $n \geq 2$.

The upper bound was established in [BGI] and we show the lower bound $t(n) = \Omega(\log^2 n)$. Theorem 2.1 implies the desired corollary

Corollary 2.1 There is a family of order n graphs with radius 2 for which any schedule for the broadcast problem requires $\Omega(\log^2 n)$ rounds. ∎

2.1 Proof outline of Theorem 2.1

The proof is based on a probabilistic argument. We exhibit the existence of a family $\mathcal{H}$ of subsets of N which cannot be hit by any $\mathcal{F}$ of size $(\log^2 n)/100$.

The lower bound of $\Omega(\log^2 n)$ changes only by a constant factor, as long as the cardinality of the family $\mathcal{H}$ is polynomial in n, and in fact, the constructed family, $\mathcal{H}$, is composed of $0.2 \log n$ subfamilies $\mathcal{H}_\ell$, each of cardinality n^7. For each ℓ, $0.4 \log n \leq \ell \leq 0.6 \log n$, let $\mathcal{H}_\ell$ be a random family of n^7 (not necessarily distinct) subsets H of N chosen as follows: for each $i \in N$, independently, $\Pr(i \in H = \frac{1}{2^\ell})$. It is shown that for any fixed family $\mathcal{F}$ of at most $\log^2 n/100$ sets there is only a small probability for $\mathcal{F}$ to hit $\mathcal{H}$. The sum of these probabilities over all such $\mathcal{F}$ is less than 1 so there is an $\mathcal{H}$ which is hit by no $\mathcal{F}$.

As observed in [BGI2] every $\mathcal{H}_\ell$ may be hit by an $\mathcal{F}$ of size $O(\log n)$. The proof essentially shows that each $\mathcal{H}_\ell$ requires an $\mathcal{F}_\ell$ of size $\Omega(\log n)$ in order to be hit and $\mathcal{F}_\ell$ "does not help" in hitting $\mathcal{H}_j$ for $j \neq \ell$.

It is easy to check that for a set A of a elements in N and a random set B of b elements in N, the probability of A hitting B is $(1+o(1))xe^{-x}$ where $x = ab/n$. Now, the cardinalities of the sets in $\mathcal{H}_\ell$ are almost surely very close to $n/2^\ell$, so, for a fixed F and $H \in \mathcal{H}_\ell$ the following three cases may occur: If $|F| << 2^\ell$ then, with high probability, $F \cap H = \emptyset$. If $|F| >> 2^\ell$, then with high probability, $|F \cap H| \geq 2$. If $|F|$ is close to 2^ℓ, then with a constant probability F hits H. Consequently, a set $F \in \mathcal{F}$ which is of the "right" size for some $\mathcal{H}_\ell$ is either too small, or too large for other $\mathcal{H}_j$.

If we associate each $F \in \mathcal{F}$ with the appropriate $\mathcal{H}_\ell$, then there is an $\mathcal{H}_\ell$ with less than $\log n/20$ associated F's (since $|\mathcal{F}| < \log^2 n/100$ and $\mathcal{H}$ consists of

$0.2 \log n$ subfamilies $\mathcal{H}_\ell$). A simple argument can already be made here to yield some lower bound, but not quite the correct one, due to two difficulties. First, the function xe^{-x} mentioned above does not decay sufficiently fast as we move from its maximum at $x = 1$. Consequently, an F of size 2^ℓ may "help" also in hitting $\mathcal{H}_j$ for j "close" to ℓ. Second, estimating the probability of $\mathcal{F}$ hitting $\mathcal{H}$ by summing over all $F \in \mathcal{F}$ and $H \in \mathcal{H}$, gives bounds which are too crude and we need some independence. These difficulties are overcome by applying the FKG-Inequality and by the following refinement of the pigeonhole argument (described formally in Lemma 2.1).

Let $\mathcal{F}$ be a fixed family of $t \leq (\log^2 n/100)$ subsets of N, Lemma 2.1 shows that there are an index ℓ ($0.4 \log n \leq \ell \leq 0.6 \log n$) and a subfamily $\mathcal{G}$ of $\mathcal{F}$ satisfying the following conditions:

1. $| \cup_{A \in \mathcal{G}} A|$ is "small".
2. For each $B \in \mathcal{F} \setminus \mathcal{G}$, $B^r = B \setminus (\cup_{A \in \mathcal{G}} A)$ is "large".

This index ℓ indicates which subfamily $\mathcal{H}_\ell$ is not hit by $\mathcal{F}$.

Let H be a member in $\mathcal{H}_\ell$. The claim is that the probability that $\mathcal{F}$ does not hit H is more than $1/n^5$. By (1.) the union of all sets in $\mathcal{G}$ is sufficiently small, thus, the probability that $H \cap A = \emptyset$ for *all* $A \in \mathcal{G}$ is at least $1/n^2$ (Lemma 2.2). The probability that $|H \cap B^r| \geq 2$ for a fixed $B \in \mathcal{F} \setminus \mathcal{G}$ is at least $1 - O(2^\ell/|B^r|)$. The dependencies between the events $|B_i^r \cap H| \geq 2$ and $|B_j^r \cap H| \geq 2$ only help by the FKG inequality ([Bo Thm. 19.5]), and so the probability that $|H \cap B^r| \geq 2$ for all $B \notin \mathcal{G}$ is at least $\prod(1 - O(2^\ell/|B^r|))$. Condition (2.) implies that this is larger than $1/n^3$ (Lemma 2.3). Since $\cup_{\mathcal{G}} A$ and the B^r are disjoint these two events are independent and the $1/n^5$ bound is proved.

The size of $\mathcal{H}_\ell$ (n^7) implies that the probability that for all $H \in \mathcal{H}_\ell$ there is an $F \in \mathcal{F}$ with $|F \cap H| = 1$ is at most e^{-n^2}. As there are less than $2^{n \log^2 n} << e^{n^2}$ such possible families $\mathcal{F}$, the lower bound follows.

2.2 A combinatorial lemma

We assume that n is large enough whenever needed, also the numerical constants can be certainly improved.

Lemma 2.1 Suppose $t \leq \frac{\log^2 n}{100}$ and let $\mathcal{F}$ be a family of t subsets of $N = \{1, 2, \ldots, n\}$. Then, there is an index ℓ and a subfamily $\mathcal{G}$ of $\mathcal{F}$ such that the following four conditions hold:

(i) $0.4 \log n \leq \ell \leq 0.6 \log n$.

(ii) $| \cup_{A \in \mathcal{G}} A| \leq 2^\ell \log n$.

(iii) For each $B \in \mathcal{F} \setminus \mathcal{G}$ define $B^r = B \setminus (\cup_{A \in \mathcal{G}} A)$ then $|B^r| \geq 2^\ell$ for all $B \in \mathcal{F} \setminus \mathcal{G}$.

(iv) For each $k \geq 0$ let f_k denote the number of sets $B \in \mathcal{F} \setminus \mathcal{G}$ such that $2^{\ell+k} \leq |B^r| < 2^{\ell+k+1}$ then

$$\sum_{k \geq 0} \frac{f_k}{2^k} \leq \log n. \tag{1}$$

Proof: Define a permutation $A_1, A_2, \ldots, A_t$ of the members of $\mathcal{F}$ as follows. Let A_1 be a set of minimum cardinality in $\mathcal{F}$. Assuming $A_1, \ldots, A_i$ have already been chosen, ($1 \leq i \leq t$), let A_{i+1} be a set in $\mathcal{F} \setminus \{A_1, \ldots, A_i\}$ such that $|A_{i+1} \setminus (\cup_{j=1}^i A_j)|$ is minimum. Define, also, $x_i = |A_i \setminus (\cup_{j<i} A_j)|$ for $1 \leq i \leq t$. For each ℓ, $0.4 \log n \leq \ell \leq 0.6 \log n$, let $j = j(\ell)$ be the smallest j such that $x_j \geq 2^\ell$. (If there is no such j, put $j(\ell) = t + 1$). Notice that by the definition of the permutation $A_1, A_2, \ldots, A_t$, for every ℓ and for every $j' \geq j(\ell)$

$$|A_{j'} \setminus \{A_1 \cup \ldots \cup A_{j(\ell)-1}\}| \geq 2^\ell. \tag{2}$$

For each ℓ put

$$d_\ell = |\{i : 1 \leq i \leq t, 2^\ell \leq x_i < 2^{\ell+1}\}|,$$

and

$$d'_\ell = d_{\ell-1} + \frac{d_{\ell-2}}{2} + \frac{d_{\ell-3}}{4} + \frac{d_{\ell-4}}{8} + \cdots.$$

Clearly

$$\sum \{d'_\ell : 0.4 \log n \leq \ell \leq 0.6 \log n\} \leq$$

$$\leq 2 \sum_{\ell \geq 0} d_\ell \leq 2t \leq \frac{\log^2 n}{50}. \tag{3}$$

Call an index ℓ *good* if $0.4 \log n \leq \ell \leq 0.6 \log n$ and $d'_\ell \leq \log n$. By (3) the average value of d'_ℓ over all $0.4 \log n \leq \ell \leq 0.6 \log n$ is at most $(\log n)/10$ and hence at least 90% of the indices ℓ, $0.4 \log n \leq \ell \leq 0.6 \log n$ are good. Notice that if ℓ is good and $j = j(\ell)$ then for $\mathcal{G} = \{A_1, \ldots, A_{j-1}\}$ we have

$$| \bigcup_{A \in \mathcal{G}} A| = x_1 + \cdots + x_{j-1} \leq \sum_{p<\ell} 2^{p+1} d_p =$$

$$= 2^\ell \left(d_{\ell-1} + \frac{d_{\ell-2}}{2} + \cdots \right) = 2^\ell d'_\ell \leq 2^\ell \log n.$$

Hence $\mathcal{G}$ and ℓ satisfy conditions **(i)** and **(ii)**. Moreover, by (2), condition **(iii)** holds as well. To complete the proof we show that for at least one (and in fact for many) good ℓ condition **(iv)** holds too. For each good ℓ and each set A_k, with $k \geq j(\ell)$, define $s(\ell, k) = r$ if

$$2^{\ell+r} \leq |A_k \setminus \bigcup_{i=1}^{j(\ell)-1} A_i| < 2^{\ell+r+1}.$$

Notice that if $\ell' > \ell$ are both good then $j(\ell') \geq j(\ell)$ and hence if $k \geq j(\ell')$ then

$$|A_k \setminus \bigcup_{i=1}^{j(\ell')-1} A_i| \leq |A_k \setminus \bigcup_{i=1}^{j(\ell)-1} A_i|.$$

Consequently, in this case $s(\ell', k)$ is strictly smaller than $s(\ell, k)$. Therefore, for every fixed k,

$$\sum \left\{ \frac{1}{2^{s(\ell,k)}} : \ell \text{ is good}, j(\ell) \leq k \right\} \leq 2. \qquad (4)$$

For each good ℓ define $y_\ell = \sum_{k \geq j(\ell)} \frac{1}{2^{s(\ell,k)}}$. By (4)

$$\sum \{y_\ell : \ell \text{ is good}\} \leq$$

$$\leq \sum_{k=1}^{t} \left\{ \frac{1}{2^{s(\ell,k)}} : \ell \text{ is good}, j(\ell) \leq k \right\} \leq 2t \leq \frac{\log^2 n}{50}.$$

Since there are at least $0.9 \cdot 0.2 \log n > \frac{1}{10} \log n$ good indices ℓ, there is at least one such ℓ with $y_\ell \leq \frac{\log n}{5} < \log n$. Define $j = j(\ell)$ and $\mathcal{G} = \{A_1, A_2, \ldots, A_{j-1}\}$. Clearly these $\mathcal{G}$ and ℓ satisfy conditions **(i)**,**(ii)** and **(iii)**. Moreover, if f_k denotes the number of sets $B \in \mathcal{F} \setminus \mathcal{G}$ such that $2^{\ell+k} \leq |B^r| < 2^{\ell+k+1}$ then

$$\sum_{k \geq 0} \frac{f_k}{2^k} = y_\ell < \log n,$$

i.e., condition **(iv)** holds too. This completes the proof of the lemma. ∎

2.3 The proof of Theorem 2.1

Now we show our probabilistic construction. For each ℓ, $0.4 \log n \leq \ell \leq 0.6 \log n$, let $\mathcal{H}_\ell = \{H_{\ell_1}, \ldots, H_{\ell_{n^7}}\}$ be a random family of n^7 (not necessarily distinct) subsets of $N = \{1, 2, \ldots, n\}$ chosen as follows, for each $i \in N$ and $1 \leq j \leq n^7$, independently, $\Pr(i \in H_{\ell_j}) = \frac{1}{2^\ell}$. Put

$$\mathcal{H} = \bigcup \{\mathcal{H}_\ell : 0.4 \log n \leq \ell \leq 0.6 \log n\}.$$

We show that with positive probability $t(\mathcal{H}) > (\log^2 n/100)$. Since $\mathcal{H}$ has less than $n^7 \log n$ sets all of which can be considered as subsets of an $\lfloor n^7 \log n \rfloor$-element set this shows that $t(n^7 \log n) = \Omega(\log^2 n)$ and hence $t(m) = \Omega(\log^2 m)$, completing the proof of Theorem 2.1. It thus remains to show that with positive probability

$$t(\mathcal{H}) > \frac{\log^2 n}{100}.$$

Let $\mathcal{F}$ be a fixed family of $t \leq (\log^2 n/100)$ subsets of N. By Lemma 2.1 there are an index ℓ and a subfamily $\mathcal{G}$ of $\mathcal{F}$ satisfying the conclusions **(i)-(iv)** of the lemma. Consider the subfamily $\mathcal{H}_\ell$ of $\mathcal{H}$ and let $H = H_{\ell_j}$ be one of the subsets in that subfamily. We claim that the probability that $|H \cap F| \neq 1$ for each $F \in \mathcal{F}$ is more than $1/n^5$ (for all sufficiently large n). To prove this claim we need the following two lemmas.

Lemma 2.2 The probability that $H \cap A = \emptyset$ for all $A \in \mathcal{G}$ is at least $1/n^2$.

Proof: This probability is precisely

$$\left(1 - \frac{1}{2^\ell}\right)^{|\cup_{A \in \mathcal{G}} A|} \geq \left(1 - \frac{1}{2^\ell}\right)^{2^\ell \log n} = \frac{1}{n^{1+o(1)}} > \frac{1}{n^2}.$$

∎

Lemma 2.3 The probability that $|H \cap B^r| \geq 2$ for all $B \in \mathcal{F} \setminus \mathcal{G}$ is at least $1/n^3$.

Proof: We first note that by the well known FKG inequality (see, e.g., [Bo Thm. 19.5]) the above probability is at least the product of the probabilities that $|H \cap B^r| \geq 2$, as B ranges over all sets in $\mathcal{F} \setminus \mathcal{G}$. Fix a set B in $\mathcal{F} \setminus \mathcal{G}$ and let $k \geq 0$ be an integer so that $2^{\ell+k} \leq |B^r| < 2^{\ell+k+1}$. Put $y = |B^r|$. Clearly if n is large enough:

$$\Pr(|H \cap B^r| \geq 2) =$$

$$= 1 - \left(1 - \frac{1}{2^\ell}\right)^y - y\frac{1}{2^\ell}\left(1 - \frac{1}{2^\ell}\right)^{y-1} \geq$$

$$\geq 1 - \left(1 - \frac{1}{2^\ell}\right)^{2^{\ell+k}} - 2^k \left(1 - \frac{1}{2^\ell}\right)^{2^{\ell+k}-1} \geq$$

$$\geq 1 - e^{-2^k} - (1 + o(1))2^k e^{-2^k} =$$

$$= 1 - (1 + o(1))\frac{2^k + 1}{e^{2^k}} \geq 1 - \frac{0.9}{2^k}.$$

Consequently, by Lemma 2.1 **(iv)** and by the FKG inequality mentioned above we conclude that if f_k is the number of sets $B \in \mathcal{F} \setminus \mathcal{G}$ such that $2^{\ell+k} \leq |B^r| < 2^{\ell+k+1}$ then the probability that H contains at least two elements from each B^r is at least

$$\prod_{k \geq 0} \left(1 - \frac{0.9}{2^k}\right)^{f_k} > \prod_{k \geq 0} (e^{-\frac{3}{2^k}})^{f_k} \geq e^{-3 \log n} = \frac{1}{n^3}.$$

This completes the proof of the lemma. ∎

The event considered in Lemma 2.2 and that considered in Lemma 2.3 are clearly independent (since $\cup_{\mathcal{G}} A$ and $\cup_{B \in \mathcal{F} \setminus \mathcal{G}} B^r$ are disjoint) and hence

$$\Pr(\forall_{F \in \mathcal{F}} |H \cap F| \neq 1) \geq$$

$$\geq \Pr(\forall_{A \in \mathcal{G}} |H \cap A| = 0 \bigwedge \forall_{B \in \mathcal{F} \setminus \mathcal{G}} |H \cap B^r| \geq 2) =$$

$$= \Pr(\forall_{A \in \mathcal{G}} |H \cap A| = 0) \cdot \Pr(\forall_{B \in \mathcal{F} \setminus \mathcal{G}} |H \cap B^r| \geq 2)$$

$$\geq \frac{1}{n^2} \cdot \frac{1}{n^3} = \frac{1}{n^5}.$$

We have thus proved that for each fixed family of at most $(\log^2 n)/100$ sets $\mathcal{F}$ there is some index ℓ such that for each of the random sets H in $\mathcal{H}_\ell$, the probability that $|H \cap F| = 1$ for some $F \in \mathcal{F}$ is at most $1 - \frac{1}{n^5}$. As the members of $\mathcal{H}_\ell$ are independent this implies that the probability that for all $H \in \mathcal{H}_\ell$ there is an $F \in \mathcal{F}$ with $|F \cap H| = 1$ is at most

$$\left(1 - \frac{1}{n^5}\right)^{n^7} \leq e^{-n^2}.$$

Therefore, for each fixed family $\mathcal{F}$ of at most $(\log^2 n)/100$ subsets of N, the probability that for all $H \in \mathcal{H}$ there is an $F \in \mathcal{F}$ with $|F \cap H| = 1$ is at most e^{-n^2}. As there are less than $2^{n \log^2 n} << e^{n^2}$ such possible families $\mathcal{F}$ this implies that for most families $\mathcal{H}$ constructed as above

$$t(\mathcal{H}) \geq \frac{\log^2 n}{100}.$$

(It is also easy to check that most of these families contains no empty sets). This completes the proof of Theorem 2.1 ∎

3 Simulation of the message-passing model

In this section we give lower and upper bounds on the number of rounds needed to simulate a single round in a message-passing network by a radio network in a synchronous environment by means of single-round simulation (SRS).

3.1 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.

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.

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

Lemma 3.1 $\chi(I(G))$ rounds are necessary and sufficient for an SRS schedule for the general model. ∎

Since no degree in $I(G)$ exceeds $2\Delta_{in}\Delta_{out}$, greedy coloring of $I(G)$ yields the following corollary.

Corollary 3.1 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 bidirected clique of order Δ, then $I(G)$ contains a $\Delta(\Delta - 1)$ clique, whence $\chi(I(G)) \geq \Delta^2 - \Delta$ and the next corollary follows from Lemma 3.1.

Corollary 3.2 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 $\Omega(\Delta^2)$ rounds. ∎

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 $A(M, r)$: 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 3.2 For every edge (x, y),

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

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

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

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 $(1 \le i \le k)$, x applies the procedure $A(M_{xy_i}, r)$ 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 the previous lemma we get

Lemma 3.3 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.

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

By the previous lemma and the choice of r

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

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

Theorem 3.1 For every $0 < q < 1$ and $1 \le \Delta_{in}, \Delta_{out} \le n$ 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. We now present an $O(\Delta_{out}\Delta_{in}^2 \log^2 n)$ round deterministic algorithm. This algorithm relies on the following easy fact.

Lemma 3.4 Let $1 \le x_1, \ldots, x_s \le n$ be s distinct integers. Then for every $1 \le i \le s$ there exists a prime $p \le s \log n$ such that $x_i \ne x_j (\mathrm{mod} p)$ for every $j \ne i$. ▮

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 $\{p_1, \ldots, p_s\}$ be the set of primes in the range $[2..(\Delta_{in} + 1) \log n]$. Each phase consists of s subphases where subphase j proceeds for p_j rounds and vertex x transmits M_i at time $x(\mathrm{mod} p_j)$.

Correctness is based on the fact that by the above lemma, for every two adjacent vertices x and y there is a prime p in the appropriate range such that $x(\bmod p)$ is different from $y(\bmod p)$ as well as from $z(\bmod p)$ for every neighbor z of y, $z \ne x$. The overall complexity of this algorithm is $O(\Delta_{out}\Delta_{in}^2 \log^2 n)$ rounds.

Theorem 3.2 For every $1 \le \Delta_{in}, \Delta_{out} \le n$ the SRS problem for the general model has deterministic distributed algorithms requiring $O(n\Delta_{out})$ or $O(\Delta_{out}\Delta_{in}^2 \log^2 n)$ rounds on any n-vertex graph G with distinct processor identities $\{1, \ldots, n\}$. ▮

3.2 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. We first consider the existence of efficient schedules for the problem. An obvious lower bound is:

Lemma 3.5 For every graph G, any SRS schedule for the uniform model requires $\Omega(\Delta_{in})$ rounds.

Proof: The claim follows from the fact that each processor has to hear from Δ_{in} different processors. ▮

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 Corollary 3.2 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.

Lemma 3.6 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 $r \geq 2$, construct a directed bipartite graph $G = (V, U, E)$ with $V = \{1, \ldots, r\}$, $U = \{\{i, j\} | 1 \leq i < j \leq n\}$ and all directed edges $(i, \{i, j\})$ for every $1 \leq i \neq j \leq r$. In this graph $\Delta_{in} = 2$ and $\Delta_{out} = \Delta = r - 1$. associate the edge $(i, \{i, j\})$ in G with the edge (i, j) in the complete graph on r vertices, K_r.

We now claim that for this graph, every SRS schedule needs $\Omega(\Delta_{in} \log \Delta) = \Omega(\log r)$ 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_r. The claim follows as the edges of K_r cannot be covered by fewer than $\lceil \log r \rceil$ 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 we do not have any lower bound higher than $\Omega(\Delta)$, yet our best upper bound is still $O(\Delta \log \Delta)$, leaving an intriguing gap. This can be stated in pure combinatorial terms as follows:

Problem: For a graph G let $\tau = \tau(G)$ be the least number of subsets $S_1, \ldots, S_\tau \subseteq V(G)$ so that if x and y are adjacent vertices, then there is an $1 \leq i \leq \tau$ for which $y \notin S_i$ and $\Gamma(y) \cap S_i = \{x\}$. Let τ_d be the maximum of $\tau(G)$ over all d-regular G. Our results show that $d \leq \tau_d \leq d \log d$. What is the true behavior of τ_d?

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 $A(M_x, r)$ 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 3.7 [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 subsection.

Lemma 3.8 There is a dependency graph H for these events with maximum degree $d \leq 2\Delta^2$.

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 three 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 $2\Delta^2$ directed edges at distance one or two from (x, y). ∎

Lemma 3.9 Applying the procedure $A(M_x, r)$ at every vertex for

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

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 3.8 and by the choice of r we have that

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

Hence by Lemma 3.7,

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

∎

For every directed graph $G = (V, E)$ define the undirected graph $\hat{G} = (V, \hat{E})$ as follows. The set $\hat{E}$ contains all the edges of E. In addition, $\hat{E}$ includes the edge (x, y), if there exists a vertex z such that (x, z) and (y, z) are in E. The following easy lemma bounds the least length of an SRS schedule for G in terms of $\hat{G}$.

Lemma 3.10 $\chi(\hat{G})$ rounds are sufficient for an SRS schedule for the uniform model. ∎

A greedy coloring of $\hat{G}$ yields the following corollary.

Corollary 3.3 For every directed graph G there is a (polynomial time constructible) SRS schedule for the uniform model of $\Delta_{out}\Delta_{in}$ 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.

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 edges (x, y) 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.

(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 3.11 The invariants

$$|T| \le |H| \text{ and } |C| \le |H|$$

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,

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

By the inductive hypothesis and the above inequality,

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

and

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

▮

Lemma 3.12 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

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

and every $x \in H$ satisfies

$$f_R(x) \le 2(f_H(x) + s_H(x) + 1)$$

. Summing these inequalities over all the processors in $R \cup C \cup H$ implies $A \le B$ where

$$A = \sum_{x \in V-T} f_R(x)$$

and B is equal to

$$2 \sum_{x \in R \cup C} (f_H(x) + s_H(x)) + 2 \sum_{x \in H} (f_H(x) + s_H(x) + 1).$$

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

$$\sum_{x \in V-T} f_R(x) + \sum_{x \in V} f_T(x) + \sum_{x \in V} f_H(x) + \sum_{x \in V} f_C(x)$$

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

(The first summation does not include processors in T since there are no F edges from T to R.) This inequality and Lemma 3.11 imply that

$$A \geq |F| - \Delta_{in}(|H| + |C| + |T|) \geq |F| - 3\Delta_{in}|H|.$$

On the other hand

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

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

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

Combining the last two inequalities with the fact that $A \leq B$ we get

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

contradicting the assumption. ∎

Lemma 3.13 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 beginning of round i. Lemma 3.12 implies that in round i the transmission succeeded on at least $\frac{|F_i|}{5\Delta_{in}}$ 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

$$\frac{\log \Delta_{in} n}{-\log\left(1 - \frac{1}{5\Delta_{in}}\right)} = O(\Delta_{in} \log n)$$

rounds all edges are satisfied. ∎

We end by considering distributed algorithms for SRS. We may apply procedure $A(M_x, r)$ (for a sufficiently large number of rounds) as a randomized algorithm for the problem, with any desired success probability.

Theorem 3.3 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 3.2 and the fact that $|E| \leq \Delta n$ we get that the probability to fail on at least one of the edges is

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

A simple calculation shows that for

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

this probability is less than $\frac{1}{q}$. ∎

Deterministic algorithms for SRS can again be designed under the assumption that processors have distinct identities $\{1, \ldots, n\}$. Both algorithms presented for the general model can be modified to run faster in the uniform model, saving the factor of Δ_{out}.

Theorem 3.4 For every $1 \leq \Delta_{in}, \Delta_{out} \leq n$ the SRS problem for the general model has deterministic distributed algorithms requiring $O(n)$ or $O(\Delta_{in}^2 \log^2 n)$ rounds on any n-vertex graph G with distinct processor identities $\{1, \ldots, n\}$. ∎

4 The census problem

The *census problem* is defined as follows. Each vertex x of G has an input value v_x taken from a commutative semigroup $SG = (S, +)$. The goal is to sum all the values to a designated vertex p, $p \in G$ (i.e., at the end of the process p has the value $\sum_{x\in G} v_x$). To simplify matters only the case of $SG = (Z, +)$, the integers, is considered, though all the results carry to the general case.

Let $G = (V, E)$ be a directed graph, denote its diameter by D (i.e., the maximum distance between any two vertices in G). We now describe a schedule of $O(D\Delta_{in})$ rounds for the census problem, and show that this result is tight by constructing a graph in which every census schedule requires $\Omega(D\Delta_{in})$ rounds.

For every vertex $x \in G$ denote by $T(G, x)$ the following directed version of a BFS tree. For a vertex y let $d(y, x)$ be the length of the shortest path from y to x. The set $V_i = \{z | d(z, x) = i\}$ is linearly ordered as follows: the index of $z \in V_i$ in this ordering is determined according to the smallest index j of $y \in V_{i-1}$ such that $(z, y) \in E$. This y is called z's parent and the vertices with the same parent are ordered arbitrarily.

Lemma 4.1 For every directed graph $G = (V, E)$ and for every processor $p \in V$, that is reachable from all the other processors in G, there is a schedule for the census problem requiring $D\Delta_{in}$ rounds.

Proof: Construct a mapping

$$\varphi : V_i \to \{1, \ldots, \Delta_{in}\}$$

such that all vertices $z \in V_i$ with the same φ value can communicate their value to their parents without creating a conflict. The value of φ is determined by the order of vertices in V_i. Let y be the parent of z and let A be the set of all $\varphi(u)$ where u precedes z in V_i and $(u, y) \in E$. The set A has cardinality at most $\Delta_{in} - 1$ and $\varphi(z)$ is the smallest integer not in this set. It is easily verified that transmission in this order is possible whence we conclude. ▮

The lower bound is demonstrated using a Δ_{in}-ary tree of depth D.

Lemma 4.2 For every D and Δ_{in} there exists a graph $G = (V, E)$ with diameter D and maximum indegree Δ_{in} and a vertex $p \in V$, such that every schedule for the census problem requires $\Omega(D\Delta_{in})$ rounds. ▮

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

[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.

[BGI2] R. Bar-Yehuda, O. Goldreich and A. Itai, Private Communication.

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

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

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

[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).

[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*, (1987).