MAX-PLANCK-INSTITUT
FÜR
INFORMATIK

Broadcasting through a noisy one–dimensional network

Luděk Kučera

MPI–I–93–106 February 1993

INFORMATIK

Im Stadtwald
W 6600 Saarbrücken
Germany

Broadcasting through a noisy one–dimensional network

Luděk Kučera

MPI–I–93–106 February 1993

Broadcasting through a noisy one-dimensional network

Luděk Kučera
Max-Planck Institut für Informatik
Saarbrücken, Germany *

February 4, 1993

Abstract: We study one bit broadcast in a one-dimensional network with nodes $\mathcal{N}_0, \ldots, \mathcal{N}_n$, in which each $\mathcal{N}_{i-1}$ sends information to $\mathcal{N}_i$. We suppose that the broadcasting is synchronous, and at each step each atomic transmission $\mathcal{N}_{i-1} \rightarrow \mathcal{N}_i$ could be temporarily incorrect with probability equal to a constant $0 < p < 1/2$. The probabilities of failure for different steps and different nodes are supposed to be independent.

For each constant c there is a "classical" algorithm with $O(n \log n)$ broadcast time and error probability $O(n^{-c})$.

The paper studies the possibility of a reliable broadcasting in $o(n \log n)$ time. We first show that one natural generalization of the classical algorithm, which was believed to behave well, has very bad properties (the probability of an error close to 1/2).

The second part of the paper presents the ultimate solution of the problem of the broadcast time in a one-dimensional nework with faults. Our algorithms have linear broadcast time, good (though not optimal) delay time, and they are extremely reliable. For example we can transmit a bit through a network of $N = 1000000$ of nodes with $p = 0.1$ in $8999774 < 9N$ steps with probability of error less than 10^{-436}.

Index terms: One-dimensional network, fault-tolerant broadcasting, linear time broadcasting.

1 Introduction

In the present paper we study how to broadcast one bit of information through a linearone-dimensional network with noise. The network has nodes $\mathcal{N}_0, \ldots, \mathcal{N}_n$, and a node $\mathcal{N}_{i-1}$ sends

*This research was partially supported by EC Cooperative Action IC-1000 (project ALTEC: *Algorithms for Future Technologies*)

(repeatedly) information to $\mathcal{N}_i$ for $i = 1, \ldots, n$. We suppose that any elementary node-to-node transmission is incorrect with probability $p < 0.5$, and probabilities of errors are independent for different nodes and different steps of the broadcast. We also suppose that p does not depend on the length n of the network, and the broadcast is synchronous and oblivious.

The task is to find a protocol that guarantees that the last node determines with large probability correctly the bit given originally to $\mathcal{N}_0$.

A very simple algorithm, which we call *classical*, solves the problem in the following way: choose a constant k; a node $\mathcal{N}_i$ waits until it gets $\lceil k \log_2 n \rceil$ bits from $\mathcal{N}_{i-1}$, and then it sends $\lceil k \log_2 n \rceil$ times the most frequent bit among those received from $\mathcal{N}_{i-1}$.

It follows immediately from known bounds to the tail of the binomial distribution that for any constant D there exists a constant $k = k(D, p)$ such that the probability that the majority bit is correct is $1 - O(n^{-(D+1)})$, and therefore the probability that the broadcast is globally correct is at least $1 - O(n^{-D})$.

The classical algorithm has broadcast time $N\lceil k \log_2 n \rceil \geq Nk \log_2 n$, and we will try to speed it up. We first study an algorithm that is its natural generalization: any node starts to broadcast immediately after it has received the first bit, and at any moment it sends the bit that is the most frequent among those received so far (breaking ties arbitrarily). Even though the latter algorithm, that will be called *temporary-majority*, seems quite similar to the former one, we will show that the probability that the broadcast is correct is $0.5(1 + o(1))$ if the time is bounded by any polynomial function. This means that the algorithm is very bad, as the probability 0.5 of the correct result would mean that the output does not depend on the input at all.

The second part of the paper introduces a class of time optimal algorithms that broadcast a bit in linear time with probability of success close to 1 as n grows to infinity. The algorithms has also very reliable - the probability of error can be made as small as $\exp(-\Omega(n^{1-\varepsilon}))$ for any constant $\varepsilon > 0$ while keeping linear time bound. We also give several particular examples; e.g. if $p = 0.1$, then one bit can be transmitted over $N = 1000000$ nodes in time $9N$ with probability of error less than 10^{-436} , while the classical algorithm needs e.g. time $19N$ for $N = 100$ and $P < 0.001$ and time $63N$ for $N > 1000000$ and $P < 10^{-9}$.

2 Temporary majority algorithm

In this paragraph it will be convenient to describe the broadcasting by means of three boolean matrices $\mathbf{E} = (\mathbf{E}_{i,j})$, $\mathbf{R} = (\mathbf{R}_{i,j})$, and $\mathbf{S} = (\mathbf{S}_{i,j})$, where $1 \leq i \leq n$ for $\mathbf{E}$, $0 \leq i \leq n$ for $\mathbf{R}$, $\mathbf{S}$, and $1 \leq j \leq m$, where $m = \lfloor n^d \rfloor$ for some constant $d > 1$. The matrix $\mathbf{E}$ is called an *error matrix*, and it describes occurences of elementary errors: $E_{i,j} = 1$ means that the bit received by the node $\mathcal{N}_i$ in the $i + j - 1$-th step was incorrect (note that $\mathcal{N}_i$ receives the first bit in the step i). If $i > 0$ then

$\mathbf{R}_{i,j}$ will denote the bit received by the node $\mathcal{N}_i$ in the $i+j-1$-th step ($\mathbf{R}_{0,j}$ will be the input bit for each j), and $\mathbf{S}_{i,j}$ will be the majority bit among $\mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,j}$. We suppose that the matrix E is generated by setting each $E_{i,j}$ to 1 with probability p independently for different entries of the matrix. Once $\mathbf{E}$ together with the input bit B_0 are given, the value of $\mathbf{R}$ is determined by

$$\mathbf{R}_{0,j} = B_0 \text{ for } j = 1, \ldots, m$$

$\mathbf{S}_{i,j}$ is the most frequent bit among $\mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,j}$ (if both 0 and 1 occurs $j/2$ times, $\mathbf{S}_{i,j}$ is chosen arbitrarily),

$$\mathbf{R}_{i,j} = \mathbf{S}_{i-1,j} \oplus \mathbf{E}_{i,j}.$$

We will suppose without loss of generality that the input bit is equal to 0.

We first show the reason why the immediate-majority algorithm is bad. Suppose that $\mathbf{E}_{i,1} = \mathbf{E}_{i,2} = 1$ for some i, and all other entries of $\mathbf{E}$ are 0, which means that the broadcast is practically error-free. Since $\mathbf{R}_{i,1} = \mathbf{R}_{i,2}$ are incorrect, i.e. equal to 1, $\mathbf{S}_{i,j} = 1$ for $j = 1, 2, 3$. Since there are no other elementary errors, $\mathbf{R}_{i+1,j} = 1$ for $j = 1, 2, 3$, which implies that $\mathbf{S}_{i+1,j}$ for $j = 1, \ldots, 5$, and in general $\mathbf{S}_{i+k,j} = 1$ for $j = 1, \ldots, 2^{k+1} + 1$, and therefore a small error amplifies from one node to another. A part of the matrix $\mathbf{E}$ in which a propagation of an error takes place will be called switch, and we will say that a switch is activated, if the number of errors that occur within a switch (and might break a propagation of the original error) is relatively small.

We will often use one technical lemma

Lemma 2.1 *Let $e_1, \ldots, e_k$ be independent 0,1-valued random variables, $1 \leq R \leq S \leq T \leq k$ be numbers, μ be a positive constant. Define* $\mathbf{Prob}(e_i = 1) = \pi_i$.. *There exists $C = C(\mu) > 0$ such that, with probability al least $1 - C^{-(S-R+1)}$,*

$$\text{if } \pi_1, \ldots, \pi_k \leq p, \text{ then } e_R + \cdots + e_j \leq (1+\mu)p(j-R+1) \text{ for each } j = S, \ldots, T,$$

$$\text{if } \pi_1, \ldots, \pi_k \leq 1-p, \text{ then } e_R + \cdots + e_j \leq (j-R+1) - (1-\mu)p(j-R+1) \text{ for each } j = S, \ldots, T,$$

Proof: It follows from the Chernoff bound, see e.g. [1], that $\pi_i \leq p$ implies

$$\mathbf{Prob}(e_R + \cdots + e_j \leq (1+\mu)p(j-R+1)) = \exp(-\frac{\mu^2}{2}p(j-R+1))$$

for arbitrary $R \leq j \leq T$, and therefore the probability that the inequality is true for all $j = S, \ldots, T$ is at most

$$\exp(-\frac{\mu^2}{2}p(S-R+1)) + \exp(-\frac{\mu^2}{2}p(S-R+2)) + \cdots + \exp(-\frac{\mu^2}{2}p(T-R+1)) =$$

$$= \exp(-\frac{\mu^2}{2}p(S-R+1))\left[1+q+q^2+\cdots\right],$$

where $q = \exp(-\mu^2 p/2)$, the proof of the second part of the lemma is similar. ♣

Now we prove that the first bit received by the node in the middle of the network is quite unreliable. Later we will show that the output is likely to be determined by this bit.

Lemma 2.2 $\mathbf{Prob}(\mathbf{R}_{\lceil n/2\rceil,1} = \mathbf{R}_{0,1}) = (1+\exp(-\Omega(n)))/2$.

Proof: We will prove by induction that $\mathbf{Prob}(\mathbf{R}_{i,1} = 0) = (1+(1-2p)^{-i})/2$. The equality is true for $i = 0, 1$. If it is true for some i, then

$$\mathbf{Prob}(\mathbf{R}_{i+1,1} = 0) = (1-p)\mathbf{Prob}(\mathbf{R}_{i,1} = 0) + p\mathbf{Prob}(\mathbf{R}_{i,1} = 1) =$$

$$= \frac{(1+(1-2p))}{2}\frac{(1+(1-2p)^{-i})}{2} + \frac{(1-(1-2p))}{2}\frac{(1-(1-2p)^{-i})}{2} =$$

$$= \frac{1+(1-2p)+(1-2p)^{i}+(1-2p)^{i+1}}{4} + \frac{1-(1-2p)-(1-2p)^{i}+(1-2p)^{i+1}}{4} =$$

$$= \frac{1+(1-2p)^{i+1}}{2}.$$

♣

Let ε be an arbitrary positive constant, and let C, ϑ be constants that will be specified in the proof of Lemma 2.4. Define a function ϕ by

$$\phi(0) = 1, \quad \phi(1) = 2, \quad \phi(i+1) = \begin{array}{ll} \lfloor 1.5\phi(i)\rfloor & \text{for } i > 2 \text{ and } \phi(i) \leq n^{2\vartheta}, \\ \lfloor 2(1-\eta)\phi(i)\rfloor & \text{otherwise,} \end{array}$$

denote by w the largest number such that $\phi(w) \leq n^{2\vartheta}$, and by z the smallest number such that $\phi(z) \geq m$, define a function ψ by

$$\psi(i) = 0 \text{ for } i < w, \quad \psi(w) = \lfloor n^{\vartheta}\rfloor, \quad \psi(i+1) = \lceil 2(1+\eta)\psi(i)\rceil \text{ for } i > w,$$

where $0 < \eta < 0.25$ is a constant chosen so that $\phi(z) \geq \eta^{-1}\psi(z)$. It is easy to see that such a choice of η is always possible. Since $\phi(i) \geq 1.4^{i}$ for any non-negative integer i, $w \leq 2\vartheta \log n/\log 1.4$.

Definition 2.3 *Given a natural $v \leq n - z$, denote by Σ_v the set of all couples (i, j) such that $v \leq i \leq v + z$, $1 \leq j \leq m$, $\psi(i - v) \leq j \leq \phi(i - v)$. We call Σ_v a* switch, *v is its* base.

We say that (i, j), $1 \leq i \leq n$, $1 \leq j \leq m$ is left to the switch Σ_v if either $i < v$ or $\phi(i - v) < j$, and (i, j) is right to Σ_v if either $v + z < i$ or $v + w \leq i \leq v + z$ and $j < \psi(i - v)$.

If $\mathbf{S}_{i,j} = \mathbf{R}_{v,1}$ for all $(i, j) \in \Sigma_v$, than we say that a switch Σ_v is activated.

We first prove that the probability that a swich is activated is not too small:

Lemma 2.4 *Given a positive constant $\varepsilon > 0$, and a node $v \leq n - z$, there are constants positive constants C, ϑ such that the probability that the switch Σ_v is activated is at least $n^{-\varepsilon/2}$ for sufficiently large n.*

Proof: Let μ be a positive constant such that

$$(1+\mu)p < \frac{1}{2},$$

$$\frac{3}{4}(1-2p(1-\mu)) < 1-2p,$$

$$(1-2p) < (1+\eta)(1-2p(1+\mu)),$$

$$3(1-2p) < 4(1-2p(1+\mu)),$$

$$(1-\eta)(1-2p(1-\mu)) < (1-\frac{\eta}{2})(1-2p).$$

A choice of μ is always possible, because the left and right hand sides of the inequalities are continuous functions of μ, and the inequalities are fulfilled for $\mu = 0$. Let $C = C(\mu)$ be a constant from Lemma 2.1,

$$\vartheta = \frac{\varepsilon \ln 1.4}{4 \ln \frac{1}{1-C}}.$$

The bound to w implies

$$(1-C)^w \geq n^{-2\vartheta \ln \frac{1}{1-C} / \ln 1.4} \geq n^{-\varepsilon/2}.$$

It is sufficient to prove that for arbitrary assignment σ of boolean values to all $\mathbf{E}_{i,j}$ such that either (i,j) is left to Σ_v or (i,j) is right to Σ_v and $i < v + z$, the probability that Σ_v is activated, conditioned on σ, is at least $n^{-\varepsilon/2}$ for large n.

Choose a fixed assignment σ, and denote by $\mathcal{A}_i$ the event that $\mathbf{S}_{i,j} = \mathbf{R}_{v,1}$ for all j such that $(i,j) \in \Sigma_v$. It is clear that $\mathbf{Prob}(\mathcal{A}_v) = 1$, $\mathbf{Prob}(\mathcal{A}_{v+1}) \geq p(1-p)$. Now we give an upper bound to $\mathbf{Prob}(\mathcal{A}_i \mid \mathcal{A}_{i-1} \wedge \sigma)$ for each $v + 2 \leq i \leq v + z$. We will denote $\mathbf{R}_{i,j} + \cdots + \mathbf{R}_{i,k}$ by $\mathbf{s}_{j,k}$. Note that $\mathbf{s}_{1,j} < j/2$ implies $\mathbf{S}_{i,j} = 0$.

Suppose first $i \leq v + w$, and denote $\alpha = \phi(i - v - 1)$, $\beta = \phi(i - v)$. It follows from Lemma 2.1 that

$$\mathbf{Prob}(\mathbf{s}_{1,j} \leq (1+\mu)pj \text{ for each } j \leq \alpha) \geq 1 - C,$$

which implies $\mathbf{S}_{i,1} = \ldots = \mathbf{S}_{i,\alpha}$ with probability at least $1 - C$, moreover we can suppose that $\mathbf{s}_{1,\alpha} \leq (1+\mu)p\alpha$.

Using Lemma 2.1 again, we can prove that, with probability at least $1 - C$, $\mathbf{s}_{\alpha+1,j} \leq (j - \alpha) - (1 - \mu)p(j - \alpha)$ for each $\alpha < j \leq \beta$. In such a case

$$\mathbf{s}_{1,j} \leq (1 + \mu)p\alpha + (j - \alpha) - (1 - \mu)p(j - \alpha) \leq \frac{j}{2} + (1 + \mu)p\alpha - \alpha + (1 - \mu)p\alpha + j(\frac{1}{2} - p(1 - \mu)) \leq$$

$$\leq \frac{j}{2} - (1 - 2p)\alpha + \frac{3}{4}\alpha(1 - 2p(1 - \mu)) < \frac{j}{2},$$

and therefore $\mathbf{S}_{i,j} = 0$. Thus, we have proved $\mathbf{Prob}(\mathcal{A}_i \mid \mathcal{A}_{i-1} \wedge \sigma) \geq 1 - 2C$.

The situation is more complicated for $v + w < i \leq v + z$. In this case put $\alpha = \psi(i - v - 1) - 1$, $\beta = \psi(i - v)$, $\gamma = \phi(i - v - 1) - 1$, $\delta = \phi(i - v)$, $\kappa = \gamma + \alpha(1 - 2p)$. Lemma 2.1 implies that

$$\mathbf{Prob}(\mathbf{s}_{1,\alpha} \leq \alpha - (1 - \mu)p\alpha) \geq 1 - C^{-\alpha},$$

$$\mathbf{Prob}(\mathbf{s}_{\alpha+1,j} \leq (1 + \mu)p(j - \alpha) \text{ for each } \beta < j \leq \gamma) \geq 1 - C^{-(\beta-\alpha)} \geq 1 - C^{-\alpha},$$

$$\mathbf{Prob}(\mathbf{s}_{\gamma+1,j} \leq (j - \gamma) - (1 - \mu)p(j - \gamma) \text{ for each } j = \kappa, \ldots, \delta)) \geq 1 - C^{-(\kappa-\gamma)} = 1 - C^{-(1-2p)\alpha}.$$

If all the above statements are satisfied, then for $\beta \leq j \leq \gamma$

$$\mathbf{s}_{1,j} \leq \mathbf{s}_{1,\alpha} + \mathbf{s}_{\alpha+1,j} \leq (\alpha - (1 - \mu)p\alpha) + (1 + \mu)p(j - \alpha) \leq \frac{j}{2} + \alpha(1 - 2p) - \frac{j}{2}(1 - 2p(1 + \mu)) \leq$$

$$\leq \frac{j}{2} + \alpha(1 - 2p) - \frac{\beta}{2}(1 - 2p(1 + \mu)) \leq \frac{j}{2} + \alpha(1 - 2p) - \alpha(1 + \eta)(1 - 2p(1 + \mu)) < \frac{j}{2},$$

if $\gamma < j \leq \kappa$ then

$$\mathbf{s}_{1,j} \leq \mathbf{s}_{1,\alpha} + \mathbf{s}_{\alpha+1,\gamma} + (j - \gamma) \leq$$

$$\leq (\alpha - (1 - \mu)p\alpha) + (1 + \mu)p(\gamma - \alpha) + (j - \gamma) \leq \frac{j}{2} + \alpha(1 - 2p) - \gamma(1 - (1 + \mu)p) + \frac{j}{2} \leq$$

$$\leq \frac{j}{2} + \alpha(1 - 2p) - \gamma(1 - (1 + \mu)p) + \frac{\gamma}{2} + \frac{\alpha}{2}(1 - 2p) \leq \frac{j}{2} + \frac{3\alpha}{2}(1 - 2p) - \frac{\gamma}{2}(1 - 2p(1 + \mu)) \leq$$

$$\leq \frac{j}{2} + \frac{3\alpha}{2}(1 - 2p) - \frac{4\alpha}{2}(1 - 2p(1 + \mu)) < \frac{j}{2},$$

and finally if $\kappa < j \leq \delta$ then

$$\mathbf{s}_{1,j} \leq (\alpha - (1 - \mu)p\alpha) + (1 + \mu)p(\gamma - \alpha) + (j - \gamma) - (1 - \mu)p(j - \gamma) \leq \frac{j}{2} - (\gamma - \alpha)(1 - 2p) + \frac{j}{2}(1 - 2p(1 - \mu)) \leq$$

$$\leq \frac{j}{2} - (\gamma - \alpha)(1 - 2p) + \frac{\delta}{2}(1 - 2p(1 - \mu)) \leq \frac{j}{2} - (1 - \frac{\eta}{2})\gamma(1 - 2p) + (1 - \eta)\gamma(1 - 2p(1 - \mu)) < \frac{j}{2},$$

which implies $\mathcal{A}_i$ with probability at least $1 - 3C^{-(1-2p)\alpha}$.

Since probabilities $\mathbf{Prob}(\mathcal{A}_i|\mathcal{A}_{i-1} \wedge \sigma)$ depend on different columns of $\mathbf{E}$, they are independent, and the probability that the switch is activated is equal to their product, and therefore it is at least

$$(1-2C)^w \prod_{i=w}^{z} \left(1 - 3C^{-(1-2p)\psi(i)}\right) = \Omega((1-2C)^w) = \Omega(n^{-\varepsilon/2}).$$

♣

It follows immediately that

Lemma 2.5 *The probability that there is an activated switch with the base greater or equal to $n/2$ is at least*

$$1 - \exp(-\Omega(n^{1-\varepsilon})).$$

Proof: Consider switches with bases $v_0 = \lceil n/2 \rceil, v_0 + z, v_0 + 2z, v_0 + 3z, \ldots$. Their number is at least $(n-1-z)/2z = \Omega(n/\ln n)$. Since they are disjoint, probabilities that they are not activated are independent, and therefore the probability that no one is activated is at most

$$(1 - n^{-\varepsilon/2})^{\Omega(n/\ln n)} \leq \exp(-n^{-\varepsilon/2})^{\Omega(n/\ln n)} = \exp(-\Omega(n^{1-\varepsilon/2}/\ln n)) = \exp(-\Omega(n^{1-\varepsilon})).$$

♣

Lemmas 2.2 and 2.5 imply that

Theorem 2.6 *Let d be a constant. The probability that the temporary-majority algorithm gives correct result when run for at most n^d steps is*

$$\frac{1}{2}(1 + \exp(-\Omega(n^{1-\varepsilon})))$$

for arbitrary constant $\varepsilon > 0$.

Proof: Choose first randomly entries $\mathbf{E}_{i,j}$ of the matrix $\mathbf{E}$ with $i \geq n/2$. It follows from Lemma 2.5 that it is very likely that there is an activated switch Σ_v, $v \geq n/2$. If a switch Σ_v is activated, all values $\mathbf{R}_{i,j}$ and $\mathbf{S}_{i,j}$ "right" to the switch are uniquely determined by $\mathbf{R}_{v,1}$, which in turn is determined by the value $\mathbf{R}_{\lceil n/2 \rceil,1}$.

Now the theorem follows from Lemma 2.2. ♣

The pictures in the end of the paper shows broadcasts in a network with $N = 500$ nodes and randomly generated faults with different probabilities p. Each picture is a visual representation of a square boolean matrix of order 500. A black (white, resp.) dot in the i-th column from left and j-th row from the bottom means that the j-th bit received by the node $\mathcal{N}_i$ is 1 (0, resp.). It can be clearly seen that if first bits received by some node are incorrect, the fault is unlikely to be corrected.

3 Linear algorithms

In this paragraph we will also study a *delay* of a network. We say that a network works with delay at most D, if any node $\mathcal{N}_i$ receives the last bit from $\mathcal{N}_{i-1}$ *before* $(t_i + D)$-th step, where t_i is the step when the node $\mathcal{N}_i$ received the first bit.

Definition 3.1 *We write $A_\pi(N, T, D, P)$ if there exists a network with $N+1$ nodes, broadcast time T, delay D, probability of a single fault π, and the probability that the global result of the broadcast is incorrect less than P.*

Lemma 3.2 $A_\pi(1, 1, 1, \pi)$.

Proof: is obvious. ♣

The next two lemmas show how to build larger networks with good properties from smaller ones.

Lemma 3.3 *Suppose $A_\pi(N, T, D, P)$. If ℓ is a natural number, then $A_\pi(N\ell, T\ell, D, R)$, where $R = 1 - (1-P)^\ell$.*

Proof: The network is a serial composition of ℓ copies of the network realizing $A_\pi(N, T, D, P)$. Nodes of the composed network are $0, 1, \dots, N\ell$, the protocol is defined inductively as follows: the vertex Ni, $i = 0, \dots, \ell-1$, receives a bit of information at the beginning of Ti-th step, and transmits it to the vertex $N(i+1)$ using protocol $A_\pi(N, T, D, P)$. The bit is available at $N(i+1)$ in the beginning of the step $Ti + T = T(i+1)$. The probability that all ℓ basic transmissions according to the protocol $A_\pi(N, T, D, P)$ are correct is at least $(1-P)^\ell$. ♣

Lemma 3.4 *Suppose $A_\pi(N, T, D, R)$. If k is a natural number, then $A_\pi(N, T + D(k-1), Dk, S)$, where*

$$S = \sum_{i \geq k/2} \binom{k}{i} R^i (1-R)^{k-i}.$$

Proof: Send the input bit k times through the network using the protocol $A_\pi(N, T, D, P)$. These broadcasts can follow each other after D steps. Therefore all k copies of the input bit are transmitted in time $T + (k-1)D$ and with delay kD. The global result is the majority bit among k bits received by the terminal node (breaking ties arbitrarily). S is clearly the probability that the number of correct bits received by the terminal node is smaller of equal to $k/2$. ♣

Combining the two previous lemmas we can obtain

Lemma 3.5 *Suppose $A_\pi(N, CN, D, R)$. If k, ℓ are natural numbers, then $A_\pi(N\ell, CN(1+(k-1)/\ell), Dk, S)$, where*

$$R = 1-(1-P)^\ell, \qquad S = \sum_{i\geq k/2} \binom{k}{i} R^i (1-R)^{k-i}.$$

Proof: Follows immediately from the fact that the delay can not be smaller than the broadcast time. ♣

Next two simple technical lemmas give bounds to probabilities that occure in Lemma 3.3 and Lemma 3.4.

Lemma 3.6

$$(1-P)^{\lceil \frac{1}{16P} \rceil} \geq \frac{7}{8} \textit{ for } 16P \leq 1.$$

Proof:

$$(1-P)^{\lceil \frac{1}{16P} \rceil} \geq (1-P)(1-P)^{\frac{1}{16P}} \geq \left(1-\frac{1}{16}\right)\left(1-\frac{1}{16}\right) > \frac{7}{8}$$

for $16P \leq 1$. ♣

Lemma 3.7

$$\sum_{i\geq k/2} \binom{k}{i} \left(\frac{1}{8}\right)^i \left(\frac{7}{8}\right)^{k-i} \leq \frac{1}{16} 2^{-k/2} \quad \textit{for } k \geq 44.$$

Proof: Denote

$$a_i = \binom{k}{i} \left(\frac{1}{8}\right)^i \left(\frac{7}{8}\right)^{k-i}.$$

Since

$$\frac{a_{i+1}}{a_i} = \frac{k-i}{i+1}\frac{\frac{1}{8}}{\frac{7}{8}} \leq \frac{\frac{k}{2}}{\frac{k}{2}+1}\frac{1}{7} < \frac{1}{7}$$

for $i \geq k/2$,

$$\sum_{i\geq k/2} a_i \leq 2a_{\lceil k/2 \rceil} \leq$$

$$\leq \frac{7}{6}\cdot 2^k \left(\frac{1}{8}\right)^{\lceil k/2 \rceil} \left(\frac{7}{8}\right)^{\lfloor k/2 \rfloor} \leq \frac{7}{6}\cdot 2^k \left(\frac{7}{64}\right)^{k/2} = \frac{7}{6}\left(4\cdot\frac{7}{64}\right)^{k/2} = \frac{7}{6}\left(\frac{8}{7}\right)^{-k/2} \cdot \frac{1}{16}2^{-k/2} \leq \frac{1}{16}2^{-k/2}$$

for $k \geq 44$. ♣

As a corollary we obtain

Lemma 3.8 *If $A_\pi(N,T,D,P)$, $P \le 1/16$, $\ell \le (16P)^{-1}$, $k \ge 44$, then*

$$A_\pi\left(N\ell, T\ell\left(1+\frac{k-1}{\ell}\right), Dk, \frac{2^{-k/2}}{16}\right).$$

Proof: See previous lemmas with $\ell = \left\lceil \frac{1}{16P} \right\rceil$. ♣

There are many possible variations of parameters ℓ, k of the preceeding lemma, that give a linear time algorithm. One of them is used to prove the next theorem:

Theorem 3.9 *Let $0 \le \pi < 1/2$, $\varepsilon > 0$ be constants. There exists a constant C such that $A_\pi(N, CN, C\log R \log^{1+\varepsilon} N, R)$ for each natural N and $R \ge 2^{-n^{1-\varepsilon}}$.*

Proof: Let $\vartheta, > 0$ be a constant. Put $Q = \sqrt{2}$, define

$$\lambda_0(x) = x, \quad \lambda_j(x) = Q^{\lambda_{j-1}(x)} \quad \text{for } i > 0.$$

Let s be the largest integer such that $\lambda_4(s) \le N$. Put

$$L_i = \lambda_4(i) \quad \text{for } i = 1,2,\ldots, \qquad K_i = \lambda_3(i+1) \quad \text{for } i = 1,2,\ldots,$$

$$\ell_i = \lceil L_i \rceil \quad \text{for } i = 1,\ldots,s, \qquad \ell_s = \lceil N(\ell_1 \ldots \ell^{s-1})^{-1} \rceil,$$

$$k_i = \max(44, \lceil K_i \rceil) \quad \text{for } i = 1,\ldots,s-2,$$

$$k_{s-1} = \max(\lceil 2\log_2 N \rceil, 44), \qquad k_s = \max(\lceil 2\log_2 R \rceil, 44),$$

$$N_0 = 1, \quad T_0 = 1, \quad D_0 = 1, \quad P_0 = \pi,$$

$$N_{i+1} = N_i\ell_i, \quad T_{i+1} = T_i\ell_i(1 + k_i/\ell_i), \quad D_{i+1} = D_i k_i, \quad P_{i+1} = Q^{-k_i}/16.$$

Note that $k_i \le K_i + 1$ for $5 \le i \le s-2$, $N \le N_s$.

It follows from the preceeding lemmas that $A_\pi(N_i, T_i, D_i, P_i)$ for $i = 1,\ldots,s$. If i is sufficiently large, then $2^{i/4}(i+1) < Q^i$,

$$2^{i/4}K_i = 2^{i/4}\lambda_3(i+1) < \lambda_3(Q^i) = \lambda_4(i) = \ell_i,$$

and therefore

$$\prod_{i=1}^{s-2}\left(1 + \frac{k_i - 1}{\ell_i}\right) \le \prod_{i=1}^{\infty} \exp\left(\frac{K_i}{\ell_i}\right) \le \exp\left(\sum_{i=1}^{\infty}\frac{K_i}{\ell_i}\right) < \infty,$$

$N < \lambda_4(s+1) \le \lambda_4(Q^{s-1}) = \lambda_5(s-1)$ implies $\log_Q N < \lambda_4(s-1) \le \ell_{s-1}$, and therefore

$$\frac{k_{s-1}}{\ell_{s-1}} \le \frac{O(\log N)}{\log_Q N} = O(1),$$

and finally

$$\frac{k_s}{\ell_s} \le \frac{O(N^{1-\vartheta})}{N(\ell_1 \dots \ell_{s-1})^{-1}} = O(N^{-\vartheta})\ell_1 \dots \ell_{s-1} \le O(N^{-\vartheta})Q^{\lambda_3(1)+\dots+\lambda_3(s-1)} \le$$

$$\le O(N^{-\vartheta})Q^{\vartheta\lambda_3(s)/2} = O(N^{-\vartheta})(\lambda_4(s))^{\vartheta/2} \le O(N^{-\vartheta})N^{\vartheta/2} = o(1),$$

which together implies a linear time bound.

Since $2\lambda_2(i-1) < \lambda_2(i)$ for $i \ge 6$,

$$\lambda_2(1) + \dots + \lambda_2(s-1) \le 7\lambda_2(s-1) = 7Q^{Q^{s-1}} = 7Q^{Q^s Q^{-1}} = 7\left(Q^{Q^s}\right)^{Q^{-1}} \le \varepsilon Q^{Q^s},$$

for sufficiently large s and therefore

$$K_1 \dots K_{s-2} = Q^{\lambda_2(2)} \dots Q^{\lambda_2(s-1)} = Q^{\lambda_2(2)+\dots+\lambda_2(s-1)} \le$$

$$\le Q^{\varepsilon\lambda_2(s)} = (\lambda_3(s))^{\varepsilon} = (\log_Q \lambda_4(s))^{\varepsilon} = (\log_Q N)^{\varepsilon} = (2\log_2 N)^{\varepsilon} \le 2\log_2^{\varepsilon} N,$$

which gives the delay bound. ♣

The bound $O(\log^{1+\varepsilon} N)$ to the delay time, proved in Theorem 3.9, can be improved, but it is not clear wheather there are algorithms combining the optimal (i.e. linear) broadcast time with the optimal (i.e. logarithmic) delay time, or what is the best value of the time-delay product.

Finally we give some examples of networks obtained using Lemma 3.8 in the case $\pi = 0.1$ and $N \geq 1000000$. All of them are obtained by repeating the construction described by Lemma 3.3 and Lemma 3.4 four times with parameters k_i, ℓ_i, $i = 1, 2, 3, 4$. We denote the length, time, delay, and probability of incorrect broadcast by N, $T = CN$, D, P respectively.

The following examples show how small can be the error probability for $T \leq 9000000$ and how small can be the time for $P < 10^{-6}$.

N	C	D	P	k_1	ℓ_1	k_2	ℓ_2	k_3	ℓ_3	k_4	ℓ_4
1000000	8.999774	320411	$0.52 \cdot 10^{-436}$	7	1	7	32	13	50	503	625
1000512	8.319200	59325	$0.32 \cdot 10^{-7}$	5	1	7	12	15	54	113	1544

The broadcast time of these networks is much better than that of the classical network with similar probability of the global error, see the next table for $\pi = 0.1$:

N	C	D	P
100	19	19	$< 10^{-3}$
100	31	31	$< 10^{-6}$
100	45	45	$< 10^{-9}$
100	57	57	$< 10^{-12}$
10000	27	27	$< 10^{-3}$
10000	41	41	$< 10^{-6}$
10000	53	53	$< 10^{-9}$
10000	67	67	$< 10^{-12}$
1000000	35	35	$< 10^{-3}$
1000000	49	49	$< 10^{-6}$
1000000	63	63	$< 10^{-9}$
1000000	67	67	$< 10^{-12}$

A disadvantage of linear time networks is their larger delay when compared to the classical networks. In some cases it would be useful to use networks that are a tradeoff between time and delay requirements, see next examples with $N \geq 1000000$ and error probability less than 10^{-6}:

N	C	D	P	k_1	ℓ_1	k_2	ℓ_2	k_3	ℓ_3
1000019	9.984499	5733	$0.58 \cdot 10^{-6}$	7	1	21	47	39	21277
1000000	10.938564	1197	$0.92 \cdot 10^{-6}$	7	1	19	32	9	31250
1000035	11.978659	693	$0.88 \cdot 10^{-6}$	7	1	33	45	3	22223
1000012	12.923554	483	$0.28 \cdot 10^{-6}$	7	1	23	26	3	38462

4 Conclusions

We have proved that one bit broadcast can be done in linear time in a linear network with constant probability of (reversible and independent) errors. This compares favorably with $O(\log N)$ time of previous algorithms. Moreover, our algorithms are very reliable; the probability of an incorrect broadcast can be almost exponentially small, while keeping the linear time bound.

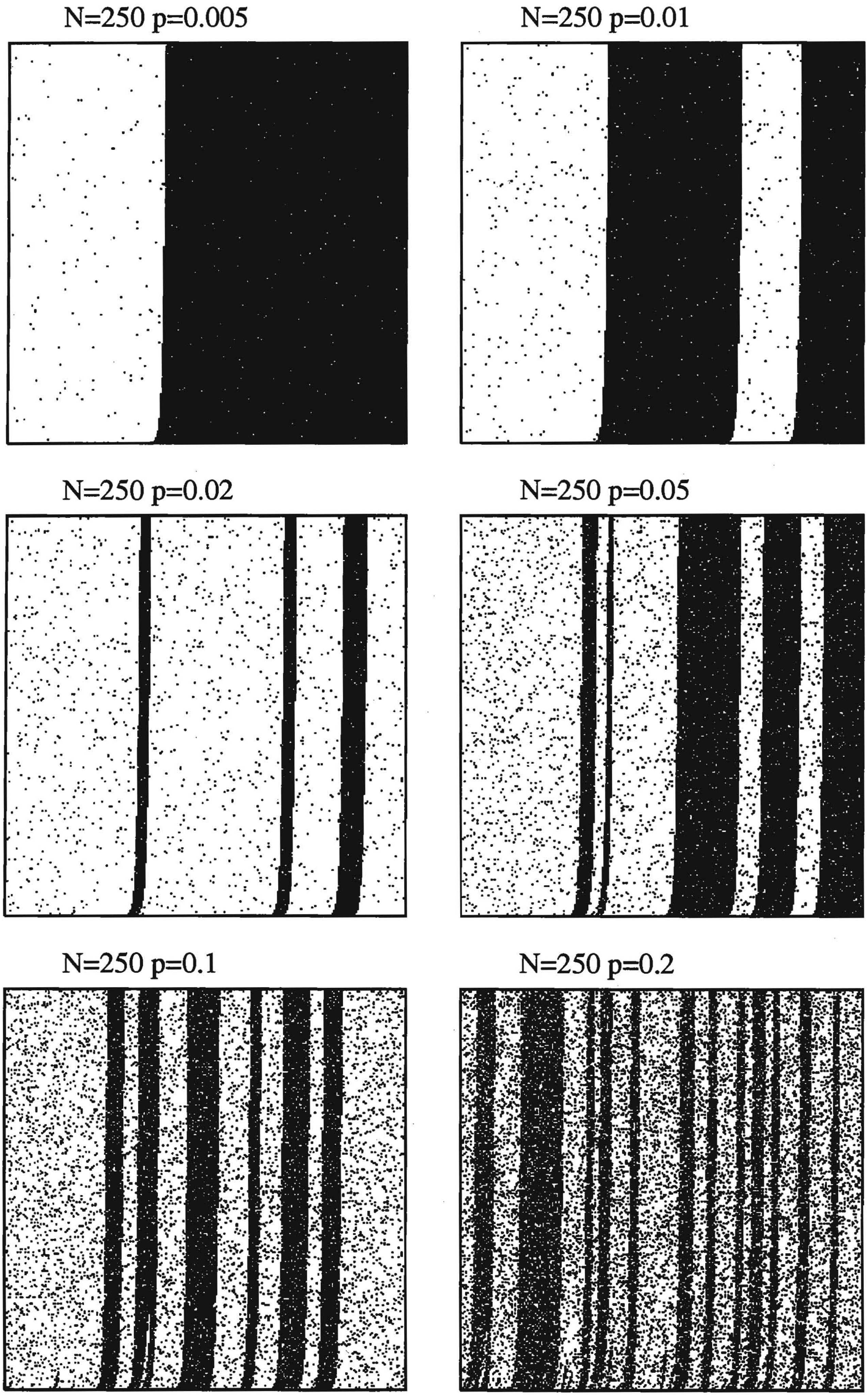
A disadvantage of our algorithms is their greater delay compared to an $O(\log N)$ delay of the classical algorithms. We have proved that it is possible to broadcast in linear time with $O(\log^{1+\varepsilon} N)$ delay for each positive constant ε. Though this bound can be improved, it is not clear, whether the optimal $O(N \log N)$ bound to the product of time and delay can be achieved.

In many cases we need to send a longer sequence of bits. If the time and delay of the network are T, D, resp., then k bits can be sent in time $T+(k-1)D$, which is equal to $O(N+k\log^{1+\varepsilon} N)$ for our networks, $O((N+k)\log N)$ for the classical network. Since it is not difficult to prove $\Omega(\log N)$ lower bound for the delay, it follows that the classical algorithm is optimal for $k = \Omega(N)$. However, if $k = O(N/\log^{1+\varepsilon} N)$, our algorithm is optimal up to a multiplicative constant, because it still needs linear time. Therefore for smaller k and sufficiently large N, our algorithm completes a broadcast of all k bits before the terminal node of the classical network receives the first bit. The assymptotically optimal algorithm is not known only for slightly sublinear k, e.g. $k = N/\log N$.

Our algorithm can be used in practical situations. They are especially suitable for extremely reliable longer networks that have to transmit a short allert message in the shortest possible time. If the error probability of any node-to-node transmission is e.g. 10%, we can send a message over a network with $N = 1,000,000$ nodes in time $9N$, and probability of an error less than 10^{-436}, while the classical network needs $35N$ steps for reliability only 99.9%.

References

[1] Alon,N., Spencer, J., and Erdős, P., *The Probabilistic Method*, J. Wiley and Sons, New York, 1992



N=250 p=0.005

N=250 p=0.01

N=250 p=0.02

N=250 p=0.05

N=250 p=0.1

N=250 p=0.2