Some Notes on Randomized Distributed Protocols

Andrea Clementi
Universitá di Roma Tor Vergata
Rome, Italy
clementi@mat.uniroma2.it

1 Basic Concepts

A probabilistic algorithm $\mathcal{A}$ is an algorithm that has access to a source S of random bits and it can takes decisions according to the outcome of the random bits it asks to S. Given a fixed input x of size n, consider the execution $\mathcal{A}(x)$ and let $r := r(x)$ be the total number of random bits $\mathcal{A}$ asks to S during the execution $\mathcal{A}(x)$. Then, we get that the outcome of $\mathcal{A}(x)$ (correct/failure), its convergence time, and any other "dynamic" parameter, are *random variables* over the probability space

$$\Omega \;=\; (\{0,1\}^r, \mathbb{P}(\cdot)), \text{ where } \mathbb{P}(\cdot) \text{ is the Probability Distribution induced by } \mathcal{A}(x)\,.$$

For instance, if $\mathcal{A}$ chooses the r bits *independently and uniformly at random (i.e. i.u.r)*[1], then $\mathbb{P}(\mathbf{y}) = 1/2^r$, for every binary sequence $\mathbf{y} \in \{0,1\}^r$.

For the sake of simplicity, assume $\mathcal{A}$ be an algorithm for a YES/NO (decision) problem $\Pi(x)$. We can introduce the crucial concept of *error probability* in terms of specific events in Ω and a notion of *good* randomized algorithms.

Definition 1.1. *We say that Algorithm $\mathcal{A}$ for problem Π has* error probability ϵ*, for some real $\epsilon \in [0,1]$, if,* for any input x*, it holds that*

$$\mathbb{P}_\Omega(\mathcal{A}(x) \;=\; \Pi(x)) \;\geq\; 1-\epsilon\,.$$

Moreover, we say that $\mathcal{A}$ solves Π WITH HIGH PROBABILITY *if, for sufficiently large n and for any input x of size n, it holds that*

$$\mathbb{P}_\Omega(\mathcal{A}(x) \;=\; \Pi(x)) \;\geq\; 1-\frac{1}{n^\beta}\,,\ \textit{for some constant } \beta > 0\,.$$

Observe that the notation $\mathbb{P}_\Omega(\cdot)$ says that the probability of the event is that defined by the Probability Space Ω induced by Algorithm $\mathcal{A}$. In the sequel, we will omit this subscript when it is clear from context.

[1] Notice that this means that the source S let each used bit y_i be such that $\mathbb{P}(y_i = 1) = 1/2$.

1.1 Randomized Protocols

We can easily transfer all concepts and notions given above for centralized algorithms to any *Distributed Computational Model* as follows. A fixed Protocol (i.e. distributed algorithm) $\mathcal{A}$, over a fixed graph $G(V, E)$ with $n = |V|$ computing nodes and starting from a given initial configuration x, specifies all actions of every node. Each node v has access to a *private*, **independent** source $s(v)$ of random bits. The decisions taken by each node v according to $\mathcal{A}$, depends (also) on the random sequence of the random bits asked by v to $s(v)$.

Let us assume that the *maximum* number of random bits chosen by any node during the execution of $\mathcal{A}$ on G, with starting configuration x, is $r \geq 0$. Then, it turns out that the outcome of Protocol $\mathcal{A}(G, x)$ (as well as its convergence time or its message complexity) is a random variable which is defined over the probabilty space

$$\Omega \ = \ (\{0,1\}^{n \cdot r}, \mathbb{P}(\cdot)), \text{ where } \mathbb{P}(\cdot) \text{ is the Probability Distribution induced by } \mathcal{A}(G, x)\,.$$

Then, the concepts "*error probability*" and "*with high probability*" can be easily extended to the framework of distributed algorithms/protocols.

2 Warm-Up: Leader Election in an Unlabeled Ring

In the previous lecturers we have seen that the *Leader Election* problem (for short LEP) is not deterministically solvable if nodes have no Unique IDs. This strong negative result can be verified even over a ring of 3 nodes (i.e. a triangle).

Now, let us consider a distributed system formed by a ring $G(V, E)$ of size n, where nodes are fully anonymous (i.e. they don't have any global ID's). Each nodes knows the size of the ring n and, clearly, the fact that it is on a ring. Let us take the standard restrictions about faults and sense of direction.

The main algorithmic question is: Can we design a good and efficient *randomized* protocol for LEP in the framework above?

The answer is yes and the only use of local randomness is in the initial phase where every node in V chooses a label i independently and uniformly at random from an alphabet $[m] = \{1, \ldots m\}$ of sufficiently large size: the hope for the global process is that there is no pair of nodes that get the same label. Then, after this labeling phase, the Protocol works exactly as in the deterministic setting (e.g. take the simplest deterministic protocol we have seen in the previous lectures). We here formally describe the protocol over a synchronous discrete-time communication model, however, it is a simple exercise to adapt the protocol (and its analysis) in the asynchronous model.

- Randomized Protocol $\mathcal{RL}(G, n; m)$. $\quad *$ (the right choice of parameter m will be given later).
- Phase 0. Wake-Up. $\quad *$ (all nodes will be activated)
- Phase 1. Every node v, chooses (independently) and uniformly at random an integer $j_v \in [m]$;
- Phase 2. Every node runs a fixed deterministic protocol for LEP assuming Node Labeling $\{j_v \, : \, v \in V\}$.

2.1 Protocol Analysis

Let us analyze the correctness of Protocol $\mathcal{RL}(G, n; m)$. For the first time in this course, we cannot prove the protocol is always correct, i.e. it *always* converges to the a valid final state of LEP. Indeed, one can easily verify that landing over the possible scenario where every node, after Phase 1, chooses the same label m makes the protocol fail! Clearly, this event is extremely unlike but it is possible!

Our first step is to derive a specific condition, i.e. event, that *deterministically* implies Protocol works correctly. In our case, this event is

$$\mathcal{B} \doteq \text{“there is no pair of different nodes } v, w \in V, \text{ with } v \neq w \text{ such that } j_v = j_w \text{ ”}$$

Indeed, as already remarked, if all labels j_v's are mutually different, then the protocol, in Phase 3, works as a standard deterministic Leader Election protocol under the assumption that all labels are different!

We now bound the probability of Event $\mathcal{B}$ in the probability space yielded by Protocol $\mathcal{RL}$ on input $(G, n; m)$. We can see Phase 2 of the protocol as a classic *Balls-into-Bins process* (see Book [3]), where there are n balls (i.e. the random choices j_v's) which are thrown independently and uniformly at random into m bins (the possible values taken by the labels). The question we are interested on is how large is the probability that any two different balls turn out to fall inside the same bin. To this aim, let us consider a fixed bin $i \in [m]$. Then the probability of event

$$\mathcal{B}_i \doteq \text{“at least two balls go to bin } i''$$

is

$$\begin{aligned}
\text{union of disjoint events:} \quad & \sum_{k=2}^{n} \binom{n}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} \\
& \leq \sum_{k=2}^{n} \binom{n}{k} \left(\frac{1}{m}\right)^k \\
\text{Stirling's Apx} \quad & \leq \sum_{k=2}^{n} \left(\frac{en}{k} \cdot \frac{1}{m}\right)^k \\
& \leq O\left(\frac{n^2}{m^2}\right) + \sum_{k=3}^{n} \left(\frac{en}{k} \cdot \frac{1}{m}\right)^k \\
& \leq O\left(\frac{n^2}{m^2} + \frac{n^4}{m^3}\right) \qquad (1)
\end{aligned}$$

Now, observe that the bound above refers to a fixed (arbitrary) bin i, while the protocol gets in trouble whenever at least one bin gets event $\mathcal{B}_i$, so we need a Union Bound here:

$$\begin{aligned}
\mathbb{P}(\mathcal{B}) = \quad & \mathbb{P}\left(\bigcup_i \mathcal{B}_i\right) \leq \sum_{i=1}^{m} \mathbb{P}\left(\mathcal{B}_i\right) \\
\text{from Eq. (1)} \quad \leq \quad & m \cdot O\left(\frac{n^2}{m^2} + \frac{n^4}{m^3}\right)
\end{aligned}$$

We can thus fix the parameter m in Protocol $\mathcal{RL}(G, n; m)$ as any integer larger than n^3 and get that $\mathbb{P}(\mathcal{B}) = O(1/n)$. We have thus proved that Protocol $\mathcal{RL}(G, n; n^3)$ makes a correct Leader Election on a ring of size n, *with high proabilty.*

3 Majority Consensus via the 3-Majority Dynamics

Let $G = ([n], E)$ be a graph, let C be a finite set of *colors* and let $\mathbf{x} : [n] \to C$ be an initial *coloring* of the nodes of G. If the number of colors is $|C| = h$ we will call $\mathbf{x}$ an *h-coloring.*

The algorithmic goal here is to design an efficient and simple protocol for *Majority Consensus.*

In this task, it is assumed that the initial coloring has some *bias s* towards some *Plurality Color* and the goal is to let the system converge to the monochromatic configuration where all nodes get the plurality color.

Let us clarify here the notion of bias of a configuration. Given any configuration $\mathbf{x}$, for any color $j \in C$, define $x(j)$ as the color-j size, i.e., the number of nodes supporting color j in $\mathbf{x}$. Then, let us order the color sizes $x(1), x(2), \ldots, x(h)$ according to the non-increasing ordering. The bias of $\mathbf{x}$ is defined as $s(\mathbf{x}) \doteq (x(1) - x(2))/2$. We will use the notation s whenever the configuration $\mathbf{x}$ is clear from the context.

Definition 3.1. *Given a distributed system $G = ([n], E)$, the ℓ-Majority-Consenus Problem is defined by the following property. Starting from any coloring $\mathbf{x} : [n] \to C$ having bias $s(\mathbf{x}) \geq \ell$, the system must converge to a stable state where all nodes support the majority color.*

Consider the following family of synchronous protocols

Definition 3.2. *The k-*Majority Protocol *(for short k-*MAJ*) works as follows*

- *At every step,*
  each node independently picks k neighbors (including itself and with repetition) u.a.r. and recolors itself according to the majority of the colors it sees (ties are broken arbitrarily).

Exercise 3.1. *1) Consider the 1-MAJ on the complete graph K_n for the binary coloring, i.e., the case $h = 1$. Show that if, at a given step $t \geq 0$, the system lies in any fixed configuration $\mathbf{x}$ with an arbitrary bias $s = s(\mathbf{x}) \geq 0$ then the* expected *bias at the next step is s as well. Formally, show that*

$$\mathbf{E}\left[s(X_{t+1}) \,|\, s(x_t) = s\right] = s\,.$$

2) Now, consider the 2-MAJ on the complete graph K_n for the binary coloring, i.e., the case $h = 2$. Prove a similar expected *behaviour to that of 1-MAJ.*

The solutions of the exercise above show that $1, 2$-*MAJ* yields no *drift* towards the majority color, no matter what is the current bias the system has: the bias does not increase in average! This fact has two main consequences. First , it holds that

$$\mathbf{P}\,(\text{the system converges to the majority color}) \;=\; \frac{x_0(1)}{n}, \quad \text{where } x_0(1) \text{ is majority size at } t = 0.$$

This implies that, even starting from a bias $s = \Omega(n)$, the error probability of the two dynamics above is still larger than an absolute constant. Second bad news is the fact that the above dynamics are very slow to converge: they take a polynomial number of steps! Proving the above two

consequences requires advanced notions in Markov Chains (so, this part is omitted here) but they are essentially due to the fact that, as remarked above, the dynamics has no expected drift and its convergence is only due to the (unpredictable) randomness of the process.

For the above facts, we will focus on 3-*MAJ*.

3.1 Majority Consensus via 3-MAJ

We analyze 3-*MAJ* in the case of two colors, i.e., $h = 2$.

Definition 3.3. *For a 2-coloring* $\mathbf{x} : [n] \to \{\texttt{red}, \texttt{blue}\}$*, we say that* $\mathbf{x}$ *is* ω-unbalanced *if its bias is s.t.* $s(\mathbf{x}) \geq \omega$.

In the next lemma we show that, if the initial configuration is sufficiently unbalanced, then 3-MAJ solves Majority Consensus within $O(\log n)$ rounds, w.h.p.

Lemma 3.4. *If* $G \equiv K_n$ *and the starting 2-coloring is* $\Omega(\sqrt{n \log n})$*-unbalanced, 3-MAJ converges to the majority color after* $O(\log n)$ *time steps, w.h.p.*

Proof. Let X_t be the random variable counting the number of `red` nodes at time t. For every node i let Y_i the indicator random variable of the event "node i is `red` at the next step". For every $a = 0, 1, \ldots, n$ it holds that

$$\mathbf{P}\left(Y_i = 1 \,|\, X_t = a\right) = \left(\frac{a}{n}\right)^3 + 3\frac{a^2(n-a)}{n^3} = \frac{a^2}{n^3}(3n - 2a)$$

Hence, the expected number of `red` nodes at the next time step is

$$\mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] \;=\; \sum_i \mathbf{E}\left[Y_i \,|\, X_t = a\right] \;=\; \left(\frac{a}{n}\right)^2 (3n - 2a) \tag{2}$$

Wlog, we assume `red` is the minority color, i.e.

$$s \;=\; s(\mathbf{x}) \;=\; \frac{x(\text{blue}) - x(\text{red})}{2} \;=\; \frac{(b-a)}{2} \;\geq\; c\sqrt{n \log n}\,,$$

and then we split the analysis in three phases according to the range the minority a falls in.

Phase 1 - The Age of Fast Bias Growth (a moves from $n/2 - \Theta\left(\sqrt{n \log n}\right)$ to $n/4$):

Suppose that the number of `red` nodes is $X_t = a$ for some $a = n/2 - s$ where $c\sqrt{n \log n} \leqslant s \leqslant n/4$ for some positive constant c. Now we show that $X_{t+1} \leqslant n/2 - (9/8)s$, w.h.p. Observe that, if this is true, then from step t to step $t+1$, the system moved from a bias s to a new bias $(1 + \Theta(1)) \cdot s$. So, we get an *exponential growth* of the bias, a good news for the completion time of this phase!

Observe that function

$$f(a) \;= a^2(3n - 2a) \;\text{ in Eq. (2)}$$

is *increasing* for every $0 < a < n$. Hence, for $a \leqslant n/2 - s$ we have that

$$\begin{aligned}
\mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] &= \left(\frac{a}{n}\right)^2 (3n - 2a) \leqslant \left(\frac{n}{2} - s\right)^2 (3n - 2(n/2 - s)) \\
&= \frac{n}{2} - \frac{3}{2} \cdot s + 2 \cdot \frac{s^3}{n^2} \\
&\leq \frac{n}{2} - \frac{3}{2} \cdot s + s \cdot \left(2 \cdot \frac{s^2}{n^2}\right) \\
\text{since } \ s \leqslant n/4 \ &\leq \frac{n}{2} - \frac{5}{4} \cdot s.
\end{aligned}$$

Now, let's first discuss this inequality for the expected behaviour of the process:

$$\mathbf{E}\left[X_{t+1} \,|\, X_t = a,\, S_t = b - a = s\right] \,\leq\, \frac{n}{2} - \frac{5}{4} \cdot s \tag{3}$$

Notice that random variables Y_i's are independent conditional on X_t. We can thus apply the Chernoff bound (Additive Form, i.e. Eq. 21) with

$$\mu \,= \frac{n}{2} - \frac{5}{4} \cdot s \ \text{ and } \ \lambda = \frac{1}{8} \cdot s$$

we then get, for every $s \leqslant n/4$ (i.e for any $a \geqslant n/4$),

$$\mathbf{P}\left(X_{t+1} \geqslant \left(\frac{n}{2} - \frac{5}{4} \cdot s\right) + \frac{1}{8} \cdot s \,\middle|\, X_t = a\right) = \tag{4}$$

$$\mathbf{P}\left(X_{t+1} \geqslant \frac{n}{2} - \frac{9}{8} \cdot s \,\middle|\, X_t = a\right) \leqslant e^{-s^2/(64n)} \tag{5}$$

Recall that in this phase, $s \geqslant c\sqrt{n \log n}$ for some constant $c > 0$, then the last term above is

$$e^{-s^2/(64n)} \,\leq\, \frac{1}{n^{\Theta(1)}}$$

and so we get that $X_{t+1} \leqslant (n/2) - (9/8)s$, w.h.p. Thus, when $c\sqrt{n \log n} \leqslant s \leqslant n/4$ the *unbalance* of the coloring increases exponentially w.h.p.

Let us name $\mathcal{E}_t$ the event

$$\mathcal{E}_t = \text{“}X_t \leqslant \max\left\{\frac{n}{4},\ \frac{n}{2} - (9/8)^t\right\}\text{”}$$

Observe that from (5) it follows that, for every $t \in \mathbb{N}$, we have

$$\mathbf{P}\left(\mathcal{E}_{t+1} \,\middle|\, \bigcap_{i=1}^{t} \mathcal{E}_i\right) \geqslant 1 - n^{-\alpha}\,, \text{ for some constant } \alpha > 0\,.$$

Thus, for $T = \frac{\log(n/4)}{\log(9/8)} = \mathcal{O}(\log n)$ the probability that the number of `red` nodes has gone below $n/4$ within the first T time steps is

$$\begin{aligned}
\mathbf{P}\left(\exists t \in [0, T] \,:\, X_t \leqslant n/4\right) &\geqslant \mathbf{P}\left(\bigcap_{t=1}^{T} \mathcal{E}_t\right) \geqslant \prod_{t=1}^{T} \mathbf{P}\left(\mathcal{E}_t \,\middle|\, \bigcap_{i=1}^{t-1} \mathcal{E}_i\right) \\
&\geqslant (1 - n^{-\alpha})^T \geqslant 1 - 2Tn^{-\alpha} \geqslant 1 - n^{-\alpha/2}
\end{aligned}$$

Phase 2 - The Age of Fast Decrease of the `reds`: a decreases from $n/4$ to $\mathcal{O}(\log n)$.
If $X_t = a$ with $a \leqslant (1/4)n$, from Eq. (2) we get

$$\mathbf{E}\left[X_{t+1} \mid X_t = a\right] = \left(\frac{a}{n}\right)^2 (3n - 2a) \leqslant$$
$$a \cdot \left(\frac{n/4}{n^2}\right)(3n) \leq \frac{3}{4}a$$

We can apply Chernoff bound (Multiplicative form) Eq. (16) with

$$\mu \ = \frac{3}{4}a \ \text{ and } \ \delta = \frac{1}{20}\,.$$

and, we can fix a a suitable positive constant β, such that

$$\mathbf{P}\left(X_{t+1} \geqslant \frac{4}{5}a \;\middle|\; X_t = a\right) \leqslant e^{-\beta a}$$

Hence as long as $a = \Omega(\log n)$, the number of `red` nodes decreases exponentially w.h.p. By reasoning as in the previous phase we get that after further $\mathcal{O}(\log n)$ time steps the number of `red` nodes is $\mathcal{O}(\log n)$, w.h.p.

Phase 3 - The Death of the `reds`: a decreases from from $\mathcal{O}(\log n)$ to 0:
Observe that for $a = \mathcal{O}(\log n)$, in Eq. (2) we easily get that

$$\mathbf{E}\left[X_{t+1} \,|\, X_t = a\right] \ \leqslant \ O\left(\frac{\log^2 n}{n}\right)\,.$$

Hence, using Markov's Inequality (see Eq. (15)) with

$$t \ = 1 \ \text{ and } \ \mu \ = \ \frac{\log^2 n}{n}\,,$$

we obtain

$$\mathbf{P}\left(X_{t+1} \geqslant 1 \,|\, X_t = a\right) \ \leqslant \ O\left(\frac{\log^2 n}{n}\right)$$

and since X_{t+1} is integer valued, it follows that all nodes are `blue`, w.h.p. □

Exercise 3.2. $**$ *In the previous lemma we showed that, if 3-MAJ starts from a 2-coloring that is sufficiently unbalanced then after $O(\log n)$ time steps the system gets into the stable "majority" configuration. A natural question is whether the lemma still holds if we use a suitable "lazy" version of 2-MAJ. For the 2-MAJ protocol over a 2-coloring we need to specify a way of breaking ties. A natural way for that is the* inertial *way:* In case of ties keep your current color*. Observe that this updating rule depends on 3 values: the two sampled ones plus that supported by the node.*
Show that if each node runs this "lazy" version of 2-MAJ then, if the system start from a $\Theta(\sqrt{n \log n})$-unbalanced 2-coloring, after $O(\log n)$ time steps all nodes have the same color, w.h.p.

4 Distributed Construction of Sparse Expanders

The construction of scalable, sparse graphs having good connectivity properties is a crucial issue in Network Design. We here focus on simple and efficient distributed protocols for this task.

For any $n \geq 1$, and a positive integer parameter d, we consider the following synchronous protocol that works on the complete graph K_n over the set V of n nodes. We denote this randomized protocol as RTA (*Request-Then-Accept*) and it is defined as follows:

- RTA(n, d)
- **Sending Action:** Each node v picks d random other notes $w_1, \ldots, w_d$ u.i.r, and sends to each of them a request of establishing edge (v, w_i) and stores edge (v, w_i) in its neighborhood list N_v.
- **Receiving Action:** Each node v receiving a link-request from node w, accept it and stores link (v, w) in N_v.
- The output graph is $G(V, E)$ s.t. $E = \cup_v N_v$.

Notice that G is in general a multigraph with self-loop. Nodes can map each multiple edge to a simple one without any relevant consequences on the results described in this note.

In the next subsections we derive some properties of the random graph G such as small maximum node-degree and good expansion.

4.1 Average and Maximum Degree of G

We label the nodes in an arbitrary fixed ordering s.t. we let $V = \{1, 2, \ldots, n\} = [n]$. By construction, the degree $d(v)$ of a vertex v is given by

$$\Delta(v) = d + \Delta_v^{\text{in}} ,$$

where Δ_v^{in} is the random variable (r.v.) counting the number of link requests received (and accepted) by v. We have the following simple

Fact 4.1. *Given any node v, it holds that $\mathbb{E}(\Delta_v^{\text{in}}) = d$ and, thus, $\mathbb{E}(\Delta(v)) = 2d$.*

Proof. Observe that the protocol produces dn total link-requests. Let us give them any fixed ordering $\{1, 2, \ldots, dn\} = [dn]$. Consider any fixed $v \in [n]$ and define, for any $j \in [dn]$ the binary r.v. X_j^v such that

$$X_j^v = 1 \text{ iff link-request } j \text{ has been sent to } v.$$

We can thus write the r.v. Δ_v^{in} as

$$\Delta_v^{\text{in}} = \sum_{j \in [dn]} X_j^v . \tag{6}$$

Moreover, since each request has been flipped i.u.r., it easily holds that

$$\mathbb{P}(X_j^v = 1) = \frac{1}{n} .$$

So,

$$\mathbb{E}(\Delta_v^{\text{in}}) = \mathbb{E}\left(\sum_{j\in[dn]} X_j^v\right) = \sum_{j\in[dn]} \mathbb{E}\left(X_j^v\right) = \sum_{j\in[dn]} \mathbb{P}(X_j^v = 1) = \frac{dn}{n} = d\,. \tag{7}$$

□

The above fact is not surprisingly at all! Indeed, the total number of link-requests is (deterministically) equal to dn, so $|E| = dn$ and each link request is selected randomly, so it is clear that, *in average*, each node must accept d link-requests. However, the bound $O(d)$ holds only *in average* and says not too much about what is the *maximum* (in-)degree of any node. We thus need stronger, concentration arguments to derive a bound on the maximum degree of a node which holds *with high probability*.

We prove the following bound.

Theorem 4.2. *For any $n \geq 1$, for any* absolute *constant*[2] *$d \geq 1$, the graph $G(V, E)$ constructed by* RTA*(n, d) has maximum degree $\Theta(\log n / \log\log n)$, w.h.p.*

Proof. From Fact 4.1, we already know that, for any fixed node $v \in [n]$, it holds that

$$\mathbb{E}(\Delta(v) = d + \Delta_v^{\text{in}}) = 2d\,.$$

We can thus focus only on the distribution of the r.v. Δ_v^{in}. As in the proof of Fact 4.1, we can write again

$$\Delta_v^{\text{in}} = \sum_{j\in[dn]} X_j^v\,, \tag{8}$$

where X_j^v are binary r.v.s which are mutually independent, uniformly distributed, and their expected sum is $\mu = d$. Hence, we can apply the Chernoff's bound in Eq. (19) for a sufficiently large β so that

$$\mathbb{P}\left(\Delta_v^{\text{in}} \;\geq\; \beta\frac{\log n}{\log\log n}\right) \;\leq\; \frac{1}{n^{\alpha}},\ \text{for some fixed}\ \ \alpha = \alpha(\beta) \geq 2\,.$$

We have not completed our job, yet! The above bound is for *one*, arbitrarily fixed node v. To bound the maximum degree we need the bound hold for all nodes, w.h.p. However, we are lucky since we can consider the union of the "bad" events

$$\mathcal{E}_v \;=\; \text{“}\ \Delta_v^{\text{in}} \;\geq\; \beta\frac{\log n}{\log\log n}\ \text{”},\ v \in [n]\,.$$

Then, from the last inequality, and since $\alpha \geq 2$, we get that

$$\mathbb{P}\left(\bigcup_v \mathcal{E}_v\right) \;\leq\; \sum_v \mathbb{P}(\mathcal{E}_v) \;\leq\; \frac{1}{n}\,.$$

[2] In particular, d does not increase with n.

The fact that $\Delta_v^{\text{in}} = \Omega(\log n / \log\log n)$ w.h.p. is rather more difficult to show: a good way to get it is to use a tight approximation of Binomial Distribution via a Poisson one. This part can be found in [3].

□

Graphs with Logarithmic Degree. An important degree range for the output random graph G is when $d = \Theta(\log n)$. Indeed, using the same analysis in the proof of Theorem 4.2 (the only difference is the version of the Chernoff's Bound we use: take Eq. (16) and Eq.(17)), we get that the distribution of the node-degrees is highly concentrated to its expectation which is $\Theta(\log n)$, *for every* node.

Theorem 4.3. *For any $n \geq 1$, for $d = \beta \log n$ for some absolute constant $\beta > 0$, every node of the graph $G(V, E)$ constructed by* RTA*(n, d) has degree $\Theta(\log n)$, w.h.p.*

4.2 Expansion of G

In the previous subsection, we proved that, if $d = O(\log n)$, the random graph G has maximal degree which is bounded by a logarithmic function, w.h.p. In network applications, these properties often represent necessary topology constraints in order to have efficient topology control and low local congestion. Other fundamental issues in this setting are those concerning the diameter and the fault-tolerance of G. It is clear that we would like G to have a small diameter and a good connectivity even if some edges will not work. To analyze the above two features, we need to introduce a fundamental concept in graph theory:

Definition 4.4. *Given a graph $G(V, E)$ with $V = [n]$, the (node)-expansion of any subset $S \subset [n]$ is defined as*

$$|N(S)| \quad \textit{where} \quad N(S) \doteq \{w \in V - S : (v, w) \in E \ \textit{for some} \ v \in S\}.$$

Then, for a fixed $\alpha \geq 0$, we say graph G is an α-expander *if any subset $S \subset [n]$ with $|S| \leq n/2$ has expansion at least $\alpha \cdot |S|$, i.e., $|N(S)| \geq alpha \cdot |S|$.*

We now prove that if we set $d = \beta \cdot \log n$ for a suitable constant $\beta > 0$, then the output random graph G is an $\Omega(1)$-expander, w.h.p. So, Protocol RTA returns an almost-sparse network having very good connectivity properties.

Theorem 4.5. *For any $n \geq 1$, for $d = \beta \log n$ for a sufficiently-large absolute constant $\beta > 0$, the graph $G(V, E)$ constructed by* RTA*(n, d) is an $\Omega(1)$-expander, w.h.p.*

Proof. To make the analysis simpler, we view the algorithm process as if it was organized in d consecutive and mutually independent *Phases* $j = 1, \ldots, d$. During Phase j, each node sends (and thus might receive and accept) the j-th link request to a node chosen i.u.r. Consider any possible size $s \leq n/2$ and fix any subset $S \subset V$ with $|S| = s$. Then, for any fixed phase j, for any fixed node $v \in V - S$, define the r.v.

$$Y_v^{(j)} \doteq 1 \quad \text{iff the } j\text{th link request of } v \text{ was toward some node} \quad u \in S\,.$$

We easily have that

$$\mathbb{P}(Y_v^{(j)} = 1) \;=\; \frac{s}{n}\,.$$

Now, as for the following random subset of nodes in $V - S$

$$N^{(j)}(S) \doteq \{v \in V - S \ : \ Y_v^{(j)} = 1\},$$

since $s \leq n/2$, for any fixed j, we easily get

$$\mathbb{E}(|N^{(j)}(S)|) = \mathbb{E}\left(\sum_{v \in V-S} Y_v^{(j)}\right) = (n-s) \cdot \frac{s}{n} \geq \frac{s}{2}. \tag{9}$$

Observe that $|N^{(j)}(S)|$ is a sum of $n - s$ independent binary r.v.s, so we can apply Chernoff Bound (Eq. 17) and get that

$$\mathbb{P}\left(|N^{(j)}(S)| \leq \left(1 - \frac{1}{2}\right)\frac{s}{2}\right) = \mathbb{P}\left(|N^{(j)}(S)| \leq \frac{s}{4}\right) \leq e^{-\frac{s}{16}}. \tag{10}$$

Define also the following "bad" events

$$\mathcal{B}^j(S) \doteq \text{“ } |N^{(j)}(S)| \leq \frac{s}{4} \text{ ” and } \mathcal{B}(S) \doteq \bigcap_{j=1}^{d} \mathcal{B}^j(S).$$

Since events $\mathcal{B}^j(S)$'s, for $j = 1, \ldots, d$ are mutually independent, then Eq. (10) immediatly implies that

$$\mathbb{P}(\mathcal{B}(S)) \leq \left(e^{-\frac{s}{16}}\right)^d = e^{-\frac{ds}{16}}. \tag{11}$$

Now, observe that if event $\mathcal{B}(S)$ does not take place then the expansion of S is at least $s/4$ and so we get that S has expansion $1/4$ and we are done. Now, we need to derive an upper bound on the probability that a bad S does exist. By making the union bound over all possible S of size s and applying the union bound, we have that this probability is at most

$$\binom{n}{s} \cdot e^{-\frac{1}{16}ds} \leq \left(\frac{en}{s}\right)^s \cdot e^{-\frac{1}{16}ds} = e^{s + s\log n - s\log s - (1/16)ds}.$$

The last term can be made not larger than $1/n^2$, provided that we choose our algorithm parameter d such that $d = \beta \log n$, for a sufficiently-large absolute constant $\beta > 0$. We need just one more union bound here! Indeed, the obtained bound holds for any *fixed* size $s \leq n/2$ and, thus, we need the bound hold *for all* s. But, since the number of possible sizes is $n/2$, we easily get that the probability that a bad set of any size (not larger than $n/2$) exists is not larger than $n/2 \cdot (1/n^2) \leq 1/n$, and we are done. □

5 Distributed Information Processing in Expander Graphs

Our interest in $\Omega(1)$-expander graphs is well-motivated by their strong connectivity properties that make such graph very useful in understanding the strctural driven factors behind important epidemic processes. A basic one of such properties is that they have small diameter.

Theorem 5.1. *Consider an infinite family of graphs of increasing size*

$$\{G_n(V_n, E_n) \quad \textit{with } V_n = [n]\,,\ n \geq 1\,\}.$$

If, an absolute constant $\alpha > 0$ *exists such that, for sufficiently large* n*, graph* G_n *is an* α*-expander, then its diameter is* $O(\log n)$*.*
Moreover, under the same assumption above, in order to fully-disconnect any node subset S *from the rest of the graph, the number of faulty links (or faulty nodes)must be at least linear in the size of* S*.*

Proof. We consider a fixed, sufficiently large graph $G_n(V_n, E_n)$ and omit the subscrit n for all the rest of the subsection, i.e.

$$G(V, E) \ \doteq\ G_n(V_n, E_n)\,.$$

Fix any $s \in V$, and let's make a Breadth-First Search (BFS) starting from s. Since G is α-expander, we define

$$L_t \ \doteq\ \{v \in V \ :\ d(s, v) = t\}\ , t \ =\ 0, 1, \ldots n-1.$$

Observe that $L_0 = \{s\}$ and $L_1 = N(s) \geq \alpha$, so $L_1 \geq 1$ since $|N(s)|$ is an integer. Now, let us define the following family of subsets:

$$I_0 \doteq L_0 \ \text{ and } \ I_t \doteq I_{t-1} \cup L_t\,,\ t \ =\ 1, 2, \ldots n-1\,.$$

Notice that, by construction, it holds that $N(I_t) = L_{t+1}$ and $|I_t| = |I_{t-1}| + |L_t|$. Hence, since G is an α-expander, as long as $|I_{t-1}| \leq n/2$, we get

$$|I_t| \ =\ |I_{t-1}| + |L_t| \ \geq\ (1+\alpha) \cdot |I_{t-1}| \ \geq\ (1+\alpha)^{t-1}.$$

From the above inequality, it must hold that $|I_\tau| \geq n/2$ for some $\tau = O(\log n)$. This means that the number of nodes within distance τ from s is at least $n/2$. Now, consider a node (if any) $w \in V - I_\tau$ and repeat the same BFS process starting from w. Then, again, thanks to the expansion of G, after $\tau' = O(\log n)$ levels of the BFS tree rooted at w, we get that the corresponding subset $I'_{\tau'}$ has size at least $n/2$. The proof is now completed by observing that the two BFS trees (the one rooted at s and that rooted at w) must either share at least one node or be connected by at least one link. □

The theorem above easily implies that if we run a Flooding protocol to perform a broadcast operation over an $\Omega(1)$-expander graph, then its completion time is logarithmic. However, Flooding is a very-expensive protocol in terms of message complexity, especially in dense graphs.

5.1 Push-Pull Protocols for Broadcast

We are now interested in Broadcast Protocols that are fast (possibly like Flooding) but they are message-saving processes. To this aim, we consider the well-known PUSH and PULL protocols.

- PULL(G(V,E), s)
- at round $t = 0$, the source s is *informed* (i.e. it has a message),
  while all other nodes are in the *non-informed* state
  (i.e. they know nothing about the source message)

- at each round $t \geq 1$, every *non-informed* node v chooses one of its neighbors u.i.r, call it w; then, if w is *informed*, v receives (i.e. pulls) a copy of the source message and gets *informed*.

The PUSH protocol works similarly but, there, every informed node sends (*pushes*) the source message to one random neighbor.

Observe that, at every round, there are at most n active links where a copy of the source message is transmitted, so the communication pattern, at every round, is a sparse subgraph. Moreover, using Chernoff's bound, it is possibile to prove that, if G is (almost) regular, then at every round the maximum number of active links which are incident to some node (the node congestion) is $O(\log n)$, w.h.p.

The above arguments tell us that the message complexity of the PULL protocol seems to be rather good. Let's analyze now the convergence time over an expander graph. For the sake of simplicity, we will here consider regular graphs having very good expansion and prove the following theorem.

Theorem 5.2. *Let $G(V, E)$ be a Δ-regular graph having expansion[3] $\Omega(\Delta)$, with any sufficiently large $\Delta = \Omega(1)$. Then, starting from any source node $s \in V$, the* PULL *protocol completes the broadcast operation on G within $O(\log n)$ rounds, w.h.p.*
This also implies that, for any $\Delta \leq n - 1$, the message complexity of the protocol is $O(n \log n)$, w.h.p.

Idea of the proof. We give here only an analysis in expectation while we leave the concentration argument as homework (see later for more details).

We first need to define the key random variables in the following recursive way. For any node subset I, define as usual the set

$$N(I) \;=\; \{v \in V - I \;:\; \exists u \in I \;\text{ s.t. }\; (u, v) \in E\}\,.$$

Let $I_t \subseteq V$ be the subset of nodes which are in the informed state at (the end of) round t. Then the following recursive relationships hold

$$I_0 \;=\; \{s\} \qquad\qquad m_0 \;=\; |I_0| = 1 \tag{12}$$

$$I_{t-1} \subseteq I_t \;\subseteq\; I_{t-1} \bigcup N(I_{t-1}) \qquad m_{t-1} \;\leq\; m_t = |I_t| \;\leq\; m_{t-1} + |N(I_{t-1})| \tag{13}$$

Our next goal is to prove that the expected size of the informed nodes increases exponentially till $n/2$. For any $t \geq 1$, look at the state the system is at the beginning of round t and consider the subset of informed nodes I_{t-1} (notice that this is not a random variable). Then, consider any node in the subset $N(I_{t-1})$ when it does the pull action. Then, define the binary r.v.

$$Y_v \;=\; 1 \;\text{ iff } v \text{ pulls from some node in } \; I_{t-1} \;.$$

It clearly holds that

$$m_t = |I_t| \;=\; m_{t-1} + \sum_{v \in N(I_{t-1})} Y_v\,.$$

[3]Notice that the expansion required in this theorem is stronger that that proved in the previous section for the output graph of Protocol RTA(n, d). However, using more complex and tight concentration arguments, it is possible to show that the output graph turns out to have in fact this stronger expansion as well.

As usual we need to give a good upper bound on $\mathbb{P}(Y_v = 1)$. Let's do it. Since the graph is Δ-regular and $v \in N(I_{t-1})$, then one of its link must be towards some informed node. So,

$$\mathbb{P}(Y_v \ = \ 1) \ \geq \frac{1}{\Delta}\,.$$

So, using linearity of expectation, we easily get

$$\mathbb{E}(m_t) \ = m_{t-1} + \sum_{v \in N(I_{t-1})} \mathbb{E}(Y_v) \ \geq \ m_{t-1} \ + \ |N(I_{t-1})| \cdot \frac{1}{\Delta}\,. \tag{14}$$

We can now use the hypothesis on the expansion of G: since G has expansion $\alpha \cdot \Delta$, for some absolute constant $\alpha > 0$, as long as subset I_{t-1} has size not larger than $n/2$, it holds that

$$|N(I_{t-1})| \geq \alpha \cdot \Delta \cdot |I_{t-1}| \ = \ \alpha \cdot \Delta \cdot m_{t-1}\,.$$

We apply the bound above to Eq. (14) and get that, for any $t \geq 1$, it holds

$$\mathbb{E}(m_t) \ \geq \ m_{t-1} \ + \ |N(I_{t-1})| \cdot \frac{1}{\Delta} \ \geq \ m_{t-1} + \alpha \cdot m_{t-1} \ = \ (1+\alpha) m_{t-1}\,.$$

The above recursive equation shows that, *in average*, the number of informed nodes increases by a constant factor. This easily implies that, after $T_1 = \beta_\alpha \log n$ rounds (where β_α is a sufficiently large constant depending only on α), we have that $\mathbb{E}(m_{T_1}) \geq n/2$. Once the information-spreading has infected at least $n/2$ nodes, we need to focus on the size of the subset of non-informed nodes at round t, i.e. subset $V - I_t$ which has now size not larger than $n/2$. Then, using its expansion guaranteed by the hypothesis, we can show that its expected size decreases by a constant factor as well. This argument is similar to the first phase of our epidemic process described above and it is left as a first simple excercise.

Proving that the above expected behaviour holds also w.h.p. can be done, for a range of values of m_t, by applying the Chernoff's bounds at every round as we made for the proof in the previous section for the 3-MAJ protocol (see the proof of Lemma 3.4). This part is left as the following Homework$**$. □

Exercise 5.1. $**$ *Assume that for some $T_0 \geq 1$, the process reaches a state where $m_{T_0} \geq \beta \log n$ (you can choose any constant $\beta > 0$ here). Then, using the right form of Chernoff's bound, show that, at every round $t > T_0$,* with probability at least $1/n^2$, *it holds that*

$$m_{t+1} \ \geq \ (1+\gamma_\alpha) m_t\,, \quad \textit{for some constant} \ \ \gamma_\alpha > 0.$$

While doing this analysis, get the reason why for $m_t = o(\log n)$, it is not possible to claim the above inequality hold w.h.p.

Actually, proving that, starting from one single source, with high probability*, the* Pull *protocol informs $\beta \log n$ nodes within $T_0 = O(\log n)$, requires a simple but smart argument which is left as an homework which gives credits!*

A Useful inequalities

A.1 Markov Inequality

Let X by any random variable assuming only non-negative values and which has expectation μ. Then, for any real $t \geq 0$, it holds that

$$\mathbf{P}\left(X \geq t\right) \;\leq\; \frac{\mu}{t}\,. \tag{15}$$

A.2 Chernoff Bound moltiplicative form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^{n} X_i$ and $\mathbf{E}\left[X\right] \leq \mu$. Then, for any $0 < \delta < 1$ the following Chernoff bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \leq e^{-\mu\delta^2/3}. \tag{16}$$

$$\mathbf{P}\left(X \leq (1-\delta)\mu'\right) \leq e^{-\mu'\delta^2/2}. \tag{17}$$

Moreover, for any $\delta > 0$, the following Chernoff bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu}. \tag{18}$$

From the last inequality, we get the following useful fact

Fact A.1. *Let $\mu \geq 1$ and let α be any positive constant. Then, for a sufficiently large constant $\beta > 0$, sufficiently large n, it holds that*

$$\mathbf{P}\left(X \geq \left(1 + \beta \cdot \frac{\log n}{\log\log n}\right)\mu\right) \leq e^{-\alpha\mu\log n}. \tag{19}$$

Proof. Consider Eq. 18 and fix $\delta := \beta \cdot \frac{\log n}{\log\log n}$, then make simple calculations and asymptotical approximations (the details are left as an homework). □

A.3 Chernoff Bound additive form

Let $X_1, \ldots, X_n$ be independent 0-1 random variables. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}\left[X\right]$. Then the following Chernoff bounds hold:
for any $0 < \lambda < n - \mu$,

$$\mathbf{P}\left(X \leq \mu - \lambda\right) \leq e^{-2\lambda^2/n}, \tag{20}$$

for any $0 < \lambda < \mu$,

$$\mathbf{P}\left(X \geq \mu + \lambda\right) \leq e^{-2\lambda^2/n}. \tag{21}$$

A.4 The Method of Bounded Differences

In Section 3.2, we used the *Method of Bounded Differences*. In particular, we applied the following concentration bound [1, 2].

Theorem A.2. *Let* $\mathbf{Y} = (Y_1, \ldots, Y_m)$ *be independent r.v.s, with* Y_j *taking values in a set* A_j. *Suppose the real-valued funcion* f *defined on* $\prod A_j$ *satisfies*

$$|f(\mathbf{y}) - f(\mathbf{y}')| \leq \beta_j$$

whenever vectors $\mathbf{y}$ *and* $\mathbf{y}'$ *differs only in the* j*-th coordinate. Let* μ *be the expected value of r.v.* $F(\mathbf{Y})$. *Then, for any* $M > 0$, *it holds that*

$$\Pr\left(F(\mathbf{Y}) - \mu \geq M\right) \leq e^{-\frac{2M^2}{\sum_{j=1}^{m} \beta_j^2}}.$$

A.5 Reverse Chernoff Bound

Let $X_1, \ldots, X_n$ be independent 0-1 random variables, $X = \sum_{i=1}^{n} X_i$, $\mu = \mathbf{E}[X]$ and $\delta \in (0, 1/2]$. If $\mu \leq \frac{1}{2}n$ and $\delta^2\mu \geq 3$ then the following bounds hold:

$$\mathbf{P}\left(X \geq (1+\delta)\mu\right) \geq e^{-9\delta^2\mu}, \tag{22}$$

$$\mathbf{P}\left(X \leq (1-\delta)\mu\right) \geq e^{-9\delta^2\mu}. \tag{23}$$

References

[1] D.P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[2] Colin McDiarmid. Concentration. in probabilistic methods for algorithmic discrete mathematics, mcdiarmid c., ramirez-alfonsin j., reed b. (eds). *Algorithms and Combinatorics, Springer, Berlin, Heidelberg*, 16:195–248, 1998.

[3] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.