COS 598a: Current problems in Theory

Assignment 1 submission

The A-Team:
Loukas Georgiadis, Carl Kingsford, Iannis Tourlakis, Tony Wirth
February 26, 2001

Exercise 1

Let the set of nodes, $V = \{0, \ldots, n-1\}$, be labelled so that the source is called 0 and we know the set of edges, E, includes

$$\{(0,1),(0,2),(0,3),\ldots,(0,n-2)\}.$$

Denote the set of nodes *fired* at time t (in a deterministic algorithm) by S_t. The algorithm $S_1 = \{0\}$—which results in nodes 1 to $n-2$ all having the message σ when $t = 1$—and

$$S_t = \{t-1\} \quad \text{for all } t \in \{2,3,\ldots,n-1\},$$

runs in $O(n)$ time. Since there must be at least one edge of the form

$$(i, n-1) \quad \text{where } i \in \{1,2,\ldots,n-2\},$$

and exactly one node fires at a time, node $n-1$ must have σ at the end of the algorithm.

We now show that this linear bound is (asymptotically) tight. Given any deterministic algorithm that takes time $n-2$ or less, we provide a $\mathcal{D}_n$-type graph in which σ is not known by node $n-1$. Without loss of generality, we assume that $S_1 = \{0\}$. (There is no point, or harm, in firing other nodes initially, but why delay firing the source node?) Nodes 1, 2, $\ldots$, $n-2$ have no information other than σ and the fact that they received σ directly from the source node 0. Therefore a deterministic algorithm can be no more complicated than a sequence of sets S_t where $S_t \subseteq \{1,2,\ldots,n-2\}$.

Consider a sequence of sets $S_2, S_3, \ldots, S_{n-2}$; that is, $n-3$ sets. We demonstrate a $\mathcal{D}_n$-type graph for which this sequence *fails*. Start with the *complete* graph in which all edges of the form

$$(i, n-1) \quad \text{where } i \in \{1,2,\ldots,n-2\}$$

appear. If none of the sets S_t is a singleton—except S_1, which we exclude from further discussion—then we have a failure graph, as node $n-1$ will always receive more than one signal at a time. Otherwise, we make several passes through the sets S_t removing edges from E. Denote the nodes with connections to node $n-1$ by C. If we ever find that $S_t \cap C$ is a singleton set then we remove the corresponding single edge from E. The end result is that the sets $S_t \cap C$ are either empty or have two or more nodes. Therefore at the end of the given broadcast algorithm, node $n-1$ in this graph does not know σ. Finally this is indeed a $\mathcal{D}_n$-type graph: we started with $n-2$ connections to node $n-1$. Since there were only $n-3$ sets that could have been singletons, we killed at most $n-3$ connections; hence at least

progress(n)
1. $T = \lceil \log_2 n \rceil$
2. receive m
3. wait until $time \mod T = 0$
4. pick $i \in [1, \ldots, T]$ with probability 2^{-i}
5. repeat i times
6.   send m

FIGURE 1: The algorithm for a $\mathcal{D}_n$ graph

one remains. Thus any deterministic algorithm of time $n-2$ or less cannot succeed on all $\mathcal{D}_n$-type graphs. Hence a deterministic solution is $\Omega(n)$.

EXERCISE 2

Now we show a randomized solution in which in $O(\log n)$ time the probability that all the nodes in a $\mathcal{D}_n$-type graph have the message is at bounded below by a constant.

We start by firing the source node s at time 1. Then all the nodes except s execute routine **progress**(n). The algorithm is called "progress" in that after running it, the probability that the last node gets the message, or the message "progresses" to the last node, will be shown to be bounded from below by a constant.

Now for the analysis of **progress**. Let X_i be the random variable counting the number of nodes that pick a number in $[i, T]$ in line 4 of **progress**. Let p be the probability that some fixed node picks a number greater than or equal to i. Then,

$$p \leq \sum_{k=i}^{\infty} 2^{-k} = 2^{-(i-1)},$$

and

$$\Pr[X_i = j] = \binom{n}{j} p^j (1-p)^{n-j}.$$

But then, for $i = \lceil \log_2 n \rceil + 1$ and $j = 1$ we get that $\Pr[X_i = j] > 1/\epsilon$ for sufficiently large n. But this is precisely the probability of success for **progress**(n).

EXERCISE 3

For this question, we got some ideas for the analysis from *On the Time-Complexity of Broadcast in Radio Networks: An Exponential gap Between Determinism and Randomization* by Bar-Yehuda, Goldreich and Itai.

Fix a graph $G = (V, E)$. Consider a broadcasting algorithm where in the first step the source s sends m to all and in the subsequent steps all other nodes execute procedure **broadcast**(n, k) from Figure 2.

```
broadcast(n, k)
1. T = ⌈log_2 n⌉
2. receive m
3. repeat k times
4.     wait until time mod T = 0
5.     pick i ∈ [1, ..., T] with probability 2^{-i}
6.     repeat i times
7.         send m
```

FIGURE 2: The general broadcasting algorithm

Note that all nodes that know m at some time $t \mod T = 0$ execute lines 5–7 of **broadcast**$(n, n/\bar{c})$ concurrently. That is, they all essentially execute **progress**(n). By the analysis in Exercise 2, **progress**(n) fails with probability at most c where c is a constant. Let $\bar{c} = 1 - c$. Note that each node that knows m begins each iteration of the loop at line 3 at the same time. Hence, each iteration of the loop at line 3 will be called a *stage* of the broadcast algorithm. Note that each stage requires $O(\log n)$ steps.

Now let us begin the analysis of the broadcast algorithm. First some definitions. Let $p(u, w)$ denote the shortest path between nodes u and w in V and let $|p(u, w)|$ be the length of this path. For any subset $U \subseteq V$ and $w \in W = V - U$ define the distance between U and w to be $d(U, w) = \min_{u \in U} |p(u, w)|$. Finally let U_i be the random variable consisting of the set of nodes that know the message after the ith stage of the algorithm.

First we show that for all nodes w in the graph, the probability that w has not received the message after $n/\bar{c}$ stages is exponentially small. To that end, fix a node w in the graph and define a sequence $\{Y_j\}$ of random variables by $Y_j = d(U_{j-1}, w) - d(U_j, w)$. Intuitively, provided that w has not received m by stage j, Y_j denotes how much "closer" m gets to w during stage j. Note that by definition of the algorithm these variables can either be 0 or 1. Moreover, if w has not received m by stage j, then

$$Y_j = \begin{cases} 1 \text{ with probability } > \bar{c} \\ 0 \text{ with probability } \leq c \end{cases} .$$

Let D denote the length of the shortest path from the source s to w. Then the probability that v does not know m after stage $n/\bar{c}$ is,

$$\Pr[d(U_{n/\bar{c}}, w) > 0] = \Pr[\sum_{j=0}^{n/\bar{c}} Y_i < D] \leq \Pr[\sum_{j=0}^{n/\bar{c}} Y_i < n] = \Pr[Y < n],$$

where $Y = \sum_{j=1}^{n/\bar{c}} Y_j$. Note that if $w \notin U_j$ then the probability of reducing the shortest path after stage j from U_j to w is at least $\bar{c}$. Hence, if w does not get the message by stage k then considering the events $\{Y_j = 1\}_{j=1}^{k}$ equiprobable (and thus independent) provides an upper bound for the actual probability

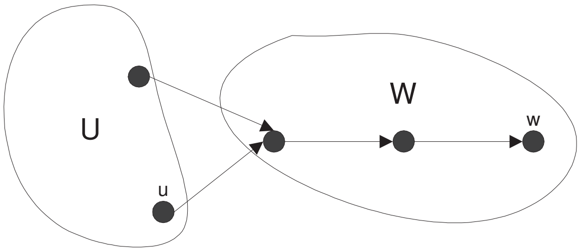

FIGURE 3 : **progress** reduces the length of the shortest path between u (that has not been reached) and the set of nodes that have the message

of not reaching w after k stages. This way we can apply a Chernoff bound, and since $E[Y] = \frac{n}{\bar{c}} \sum_j E[Y_j] > n$ we have that,

$$\begin{aligned} \Pr[Y \le (1-\varepsilon)n] &\le \Pr[Y \le (1-\varepsilon)E[Y]] \\ &\le \exp\left(-E[Y]\varepsilon^2/2\right) \\ &\le \exp\left(-n\varepsilon^2/2\right). \end{aligned}$$

So as desired, the probability for a fixed node not knowing m after $n/\bar{c}$ stages is exponentially small. Hence, the probability that all nodes know m after $n/\bar{c}$ stages is bounded below by a constant. So since each stage takes $O(\log n)$ steps, it follows that with constant probability all nodes know m after $O(n \log n)$ steps of the given broadcasting algorithm.