Solution-1:

1. Suppose we have m rounds and a vertex v has not reached value $\log n$. Consider that we have tossed $H = H_a + H_b$ heads, and we change the vertex of observation H_a times. We change the vertex v of observation when the current vertex does not increment despite getting a head, then we shift to the vertex in the neighbourhood of v that has count less than $count(v) - b$.

Index	Vertex	Counter	Coin
m	v	k	H
$m-1$	v	$k-1$	H
$m-2$	v'	$k-b-2$	T
$m-3$	v''	$k-2b-3$	H
...	...	...	...
1		$k-H_a(b+1)-H_b$	T

The probability of a delay-sequence of length m and H heads: $P(H\ heads) = p^H(1-p)^{m-H}$
The number of such a delay-sequences with H heads is:

$$\#\ of\ delay\ sequences\ with\ H\ heads = \binom{m}{H} \sum_{\forall possible\ H_a} \binom{H}{H_a} d^{H_a} \le \binom{m}{H} \sum_{H_a=0}^{H} \binom{H}{H_a} d^{H_a} = \binom{m}{H}(d+1)^H$$

Probability of the delay sequence of length m where H heads occur (for any given node):

$$P(delay\ sequence\ with\ H\ heads) \le (\#\ of\ sequences\ with\ H\ heads)\cdot P(H\ heads)$$
$$\le \binom{m}{H}(d+1)^H p^H(1-p)^{m-H} \le \binom{m}{H}(d+1)^H\alpha^m;\ \ \alpha = \max(p, 1-p)$$
$$\Rightarrow P(delay\ sequence\ with\ H\ heads) \le \binom{m}{H}(d+1)^H\alpha^m$$

To account for the probability of occurrence of a delay sequence starting at a particular node, we need only sum over all possible H. This summation automatically accounts for all possible k, since in $\#\ of\ sequences\ with\ H\ heads$ we have already accounted for all possible H_a.

$$\therefore P(delay\ sequence\ occurs\ at\ node) = \sum_{\forall possible\ H} P(delay\ sequence\ with\ H\ heads)$$

Also, since $k - H_a(b+1) - H_b \ge 0 \Rightarrow k \ge bH_a + H \Rightarrow H \le k < \log n. \Rightarrow H < \log n$

Therefore, the probability of a delay sequence occurring would be (using union bound):

$$P(atleast\ one\ delay\ sequence) \le (number\ of\ nodes) \times \sum_{\forall possible\ H} P(delay\ sequence\ with\ H\ heads)$$
$$\le n\sum_{H=0}^{\log n} P(delay\ sequence\ with\ H\ heads) \le n\sum_{H=0}^{\log n}\binom{m}{H}(d+1)^H\alpha^m$$
$$\le n\log n\binom{m}{\log n}(d+1)^{\log n}\alpha^m \le n\log n\left(\frac{me}{\log n}\right)^{\log n}(d+1)^{\log n}\alpha^m$$

Choose $m = C\log n$, then:

$$P(atleast\ one\ delay\ sequence) \le n\log n\ \left(Ce(d+1)\alpha^C\right)^{\log n}$$

We can find such a $C > 1$, such that $(Ce(d+1)\alpha^C) < e^{-3}$:

$$Ce(d+1)\alpha^C < e^{-3}$$
$$\Rightarrow \log C + C\log\alpha < \log\frac{e^{-4}}{d+1}$$
$$\Rightarrow C\log\frac{1}{\alpha} > \log\left(\frac{d+1}{e^{-4}}\right) + \log C$$

$$\Rightarrow C > \frac{\log C}{\log 1/\alpha} + \frac{\log\left(\frac{d+1}{e^{-4}}\right)}{\log 1/\alpha} = k_1 \log C + k_2; \quad for\ some\ k_1, k_2 > 0$$

Since $x > O(\log x)$, we can find such a C. Suppose C_1 is as such and let $m = C_1 \log n$.
Then, $P(atleast\ one\ delay\ sequence) \leq n \log n\ \ (C_1 e(d+1)\alpha^{C_1})^{\log n} < n \log n\ \ (e^{-3})^{\log n} < \frac{n \log n}{n^3} < \frac{\log n}{n^2}$

Hence, $P(atleast\ one\ delay\ sequence) < \frac{\log n}{n^2}$, this is very small for large n. Thus, with high probability no delay sequence occurs for $m = C_1 \log n$. And C_1 is a function of b, d, p ($\because \alpha = \max(p, 1-p)$).

Thus within $O(\log n)$ rounds, all counters will reach value $\log n$ with high probability.

Solution-2:

1. The algorithm, chosen Heads for all neighbours to make analysis easier. One can also have a complementary choice with $p' = 1 - p$ and the algorithm would still be the same.

Algorithm-1: A distributed algorithm to compute independent set

1. Each node i tosses a coin which gives heads with probability p;
2. Each node i sends the outcome of its coin toss to its neighbours and, in turn, receives their outcome as well;
3. Each node i adds itself to set S if ... **it got Tails** and all its **neighbours got Heads**;

2. Consider a set S obtained from Algorithm-1.

   Clearly, for any $v \in S$, we know that $v \in V$. $\Rightarrow S \subset V$

   We shall prove by contradiction that S is an independent set:

   Let $i, j \in S$ be two vertices.

   $$\Rightarrow i\ \&\ j\ got\ Tails\ and\ all\ neighbours\ of\ both\ i\ and\ j\ got\ Heads$$

   Now, assume that there is an edge from i to j, $(i,j) \in E$.

   $$\Rightarrow j\ is\ a\ neighbour\ of\ i, and\ j\ got\ Tails$$
   $$\Rightarrow One\ of\ the\ neighbours\ of\ i\ got\ Tails$$
   $$\Rightarrow i\ must\ not\ be\ in\ S$$

   We have arrived at a contradiction.
   Our assumption that there exists an edge from i to j (or an edge from j to i) must be wrong. Hence for any two arbitrary vertices $i, j \in S$, $(i,j) \notin E$.

   Therefore, $S \subset V$ and for $i, j \in S, (i,j) \notin E \Rightarrow S$ is an independent set.

3. Expected size of set S:

   Define:

   $$X_i = \begin{cases} 1, & node\ i\ adds\ self\ to\ S \\ 0, & otherwise \end{cases}$$

$$X = \sum_{i \in V} X_i$$

Clearly, $X = |S|$. Now, $P(X_i = 1) = P(all\ neighbours\ get\ Heads) * P(i\ gets\ Tails)$.

$$\Rightarrow P(X_i = 1) = p^d(1-p)$$
$$\Rightarrow \mathbf{E}[X_i] = p^d(1-p)$$

By linearity of expectation,

$$\mathbf{E}[X] = \sum_{i \in V} \mathbf{E}[X_i] = np^d(1-p)$$
$$\Rightarrow \mathbf{E}[|S|] = np^d(1-p)$$

4. The value of p maximizing $\mathbf{E}[|S|]$:
   We first compute the derivatives:

$$\frac{\partial \mathbf{E}[|S|]}{\partial p} = ndp^{d-1} - n(d+1)p^d$$
$$\frac{\partial^2 \mathbf{E}[|S|]}{\partial^2 p} = ndp^{d-2}\big((d-1) - (d+1)p\big)$$

   For maxima, $\partial \mathbf{E}[|S|]/\delta p = 0$ and $\partial^2 \mathbf{E}[|S|]/\partial^2 p < 0$:

$$\Rightarrow dp^{d-1} - n(d+1)p^d = 0 \Rightarrow p = \frac{d}{d+1}$$
$$\frac{\partial^2 \mathbf{E}[|S|]}{\partial^2 p}\Big|_{p=\frac{d}{d+1}} = nd\left(\frac{d}{d+1}\right)^{d-2}\left((d-1) - \frac{(d+1)d}{d+1}\right) < 0$$

   Thus, for $p = \frac{d}{d+1}$, the expected size of $|S|$ is maximized.

$$\mathbf{E}[|S|] = n\left(\frac{d}{d+1}\right)^d\left(1 - \frac{d}{d+1}\right) = \frac{nd^d}{(d+1)^{d+1}} = \left(\frac{d}{d+1}\right)^d \cdot \frac{n}{d+1}$$
$$\Rightarrow \mathbf{E}[|S|] = \frac{1}{\left(1+\frac{1}{d}\right)^d}\frac{n}{d+1} \cong \frac{1}{e} \cdot \frac{n}{d+1}\ for\ large\ enough\ d$$

Solution-3:

1. The following deterministic algorithm would be to keep $k+1$ alive links. But since we have no prior knowledge of k, we will assume the worst and choose to keep $n/2 + 1$ alive links.

Algorithm-2: A deterministic algorithm to ensure that v remains connected to at least one slave vertex
Repeat for each day: Successfully connect to slave vertices until $n/2 + 1$ active links

   Since the number of links breaking among the $n/2\ + 1$ can be at most $n/2$. There will always be at least one slave vertex connected the next day. Clearly, the deterministic algorithm always keeps $\frac{n}{2} + 1$ alive links, and scales linearly with n.

2. Randomized algorithm:

Algorithm-3: A randomized algorithm to ensure that v remains connected to at least one slave vertex
Repeat for each day:

	Successfully connect to uniformly randomly picked slave vertices until $\lceil 4\log_2 n \rceil$ active links

The Algorithm-3 fails if all of the connected links break on the next day. Say there are a active links in the previous day (i.e., $(i-1)^{th}$ day) and k_i be the number of links that are failing today (i^{th} day). Then,

$$E_i = event\ that\ the\ algorithm\ fails\ on\ the\ i^{th}\ day$$

$$P(E_i) = P(no\ slave\ vertex\ linked) = \left(\frac{k_i}{n}\right)^a \leq \frac{1}{n^4}\ for\ a \geq \left\lceil 4\log_{\frac{n}{k_i}} n \right\rceil$$

Since we have no prior knowledge of k_i, except $k_i \leq n/2$. We have,

$$\left\lceil 4\log_{\frac{n}{k_i}} n \right\rceil \leq \lceil 4\log_2 n \rceil$$

$$\Rightarrow P(E_i) \leq \frac{1}{n^4}\ for\ a = \lceil 4\log_2 n \rceil$$

Using the union bound:

$$P(fails\ within\ n^2\ days) = P\left(\bigcup_{i=1}^{n^2} E_i\right) \leq n^2 P(E_i) \leq \frac{n^2}{n^4} = \frac{1}{n^2}\quad [union\ bound]$$

$$P(success) = 1 - P(fails\ within\ n^2\ days) \geq 1 - \frac{1}{n^2}$$

Thus, for $a = \lceil 4\log_2 n \rceil$ this randomized algorithm succeeds with high probability.
This randomized algorithm keeps $\lceil 4\log_2 n \rceil$ active links.