Spill the Tea 🍵: Randomized Rumor Spreading in a Distributed System

Andrew Wei

nowei@cs.washington.edu

CSE 525 Winter 2021

Randomized Algorithms and Probabilistic Analysis

Code: https://github.com/nowei/RandomizedRumorSpreading

1. Introduction

In this paper, we will be summarizing and provide some intuition and empirical results for work by Karp et al. on randomized rumor spreading [1]. Specifically, we will analyze rumor spreading within a synchronous[1] distributed system with randomized communication. This analysis will be performed within the Random Phone Call Model, where we have n nodes and we have rounds of communication. In each round, each node can communicate with a random node. We will analyze the time (rounds of communication) and the number of messages (transmissions) it takes for a single rumor to be spread within this system. We define the rumor as some piece of information that is initially started by one of the nodes.

Imagine a nice setting where we had synchronous, uninterrupted, reliable (non-random) communication, and coordination. Let us also assume that every node had knowledge of every other node and nodes were identified in a way that could be globally ordered. In such a setting, every node in the system can know the rumor within $O(\log n)$ communication rounds with $O(n)$ messages. This can be achieved by having the knowledgeable nodes inform a different unknowledgeable node on each round in a divide and conquer, splitting fashion.

We are not in this setting. Additionally, real networks may suffer from lossy connections, dropped messages, and asynchronous communication. Thus, the setting we will explore may not be the most realistic within modern networks, but they are not entirely unrealistic. Synchronous communication is used within Google's Spanner [2], which uses timestamps and bounds on communication time to resolve event ordering. We can also imagine that random connections can occur in a decentralized setting in which we do not have knowledge of the entire network and who we can communicate with may change over time, like within route discovery among routers or within Named Data Networks [3].

This type of information dissemination is generally called a gossip algorithm, which means that each node can communicate with one neighbor in each round. They fall within the class of epidemic algorithms, which were named so because they were based on epidemic spread.

The constraint of one communication partner is not wholly unrealistic, as you can imagine a chat application that allows users to communicate with each other. In the time you are communicating with another person and reading what they wrote, you may also be receiving messages from others and they could be reading what you had sent previously.

Some applications include things like lazy updates on distributed data stores or peer-to-peer decentralized networking. The randomness of our setting makes it a little hard to map to practical applications, but we can imagine that in a setting like ours, we may want to transmit our rumor to anyone who has not heard it in a wave-like fashion to facilitate a rollout of updates in a random, fair manner. The random aspect provides the system with robustness, simplicity, and scalability, as no state about communication partners must be stored after the round and everyone is constrained to perform actions similarly.

[1] There is a time bound on when messages will be received. These types of networks/settings are usually attained through some type of clock/time coordination.

To begin with, previous work by Demers et al. [8] describe the *push* and *pull* transmission schemes. In the *push* scheme, each node that connects to another node pushes the rumor across the connection. They observed that the fraction of unknowledgeable nodes that do not know a rumor decreases exponentially with the number of transmissions. They show that in such a scheme, a rumor could be spread within $\Theta(n \ln n)$ transmissions within $\Theta(\ln n)$ rounds. In the *pull* scheme, each node that connects to another node pulls the rumor/information from the other node. They show that such a scheme could spread the rumor faster among uninformed nodes and they showed that it takes $\Theta\left(n \sqrt[3]{\ln n}\right)$ transmissions to spread the rumor.

In this paper, we will look into the communication costs of epidemic algorithms using this randomized communication scheme for rumor spreading, following a rumor-mongering approach. Rumor mongering is defined in [8] by Demers et al., where when a node learns about a new rumor, they consider it as "hot" and forwards the rumor to another site at random.

There are a few papers that build off of this idea. Shi et al. use randomized gossip algorithms are tested in an asynchronous algorithm with unreliable, probabilistic communication channels [4]. Another paper by Shi et al. reports some results within this setting with the task of achieving consensus with attractive/repulsive links [5]. Boyd et al. relate the problem of randomized gossip algorithms on the distributed averaging problem and try to relate the problem to Markov chains [6]. Lastly, Peng et al. generalize the task by relaxing the constraint of synchronous communication to that of loosely synchronized communication with the intent of performing aggregation operations [7].

2. Main Contributions

The main contributions of the paper are as follows:

1. We will prove that in the *push&pull* model, the rumor can be spread in $O(\ln n)$ rounds and with $O(n \ln \ln n)$ transmissions
2. Introduce the *median-counter* algorithm to determine when to terminate, which takes $O(\ln n)$ rounds with $O(n \ln \ln n)$ transmissions

3. Random Phone Call Model

Here, we define the Random Phone more formally. We have the following setting:

- We have rounds denoted by t.
- Let r be the rumor we are transmitting.
- We denote the set of nodes, V. Edges on round t are denoted as E_t, where we have that $E_t \subset V \times V$.
- We have $|V| = n$ nodes and on each round, each of the nodes can communicate with another, random node. We will call such an edge a connection. The other node is drawn independently and uniformly at random from the set of nodes.
- When a rumor is spread through a connection, the rumor is maintained on the sending node and the receiver obtains a copy of this rumor. Obtaining multiple copies of a rumor

doesn't provide any new information about the rumor and is equivalent to receiving a single rumor.

- On each round, t, a node can choose whether to transfer the rumor to the random node it has connected to or not. The rumor that gets passed to the random node during a round cannot be forwarded by the random node to another random node during the same round.
- The state on a node, at round t, is defined as:
    - its address;
    - the number of nodes in the system, n;
    - the address of the random connection for the round;
    - information from previous rounds, e.g. round number; and
    - the age of the rumor, initially at 0 and incremented on each round

Generally, a rumor may start on any round, but we concern ourselves with the time it takes to spread a single rumor. The goal is to spread the rumor such that all participants know the rumor within a small number of rounds and a small number of transmissions.

3.1. Push&Pull Scheme

We will be utilizing the *push&pull* scheme also introduced in [8] and analyzed in [1]. Essentially, the rumor is both pushed and pulled in each communication round. Intuitively, this combines the exponential growth of the push algorithm and quadratic shrinking of the pull algorithm. We consider both pushing and pulling on each round, but observe that it is unnecessary for knowledgeable nodes to pull the rumor. We will use this extra communication to express knowledge of the termination condition.

The age of the rumor is incremented on each round and is attached to the rumor. Everyone pushes and pulls the rumor until they learn that the age of the rumor is higher than $t_{max} = \log_3 n + O(\log\log n)$. Then we will prove the following later:

Theorem 3.1.1.

All nodes are informed of the rumor in time $\log_3 n + O(\log\log n)$ with $O(n\log\log n)$ messages with high probability.

4. Median Counter Algorithm

This algorithm helps estimate the growth of the set of informed nodes to determine when to stop spreading the rumor. This is a distributed termination mechanism, basically fancy words for learning about when to end. Each node (besides the initial knowledgeable node) goes through 4 states related to the knowledge of the rumor:

1. **State A** – Node is uninformed about the rumor.
    i. If the rumor is heard from those in State B, the node transitions to State B-1.
    ii. If the rumor is heard from a node in State C, then the node transitions to State C.
2. **State B-*m*** – Node is informed about the rumor and holds a counter for the number of rounds it has heard rumors with similar counter value (initially starts in State B-1).

i. If, in a round, the node receives more rumors from other nodes in State B with a value greater than or equal to m than values lower than m, then the node transitions to State B-(m + 1).
ii. If a node in State B-m reaches a maximum counter value, which we denote as ctr_{max}, or receives a rumor from a node in State C, then the node transitions to State C.

3. **State C** – A node stays in this phase for at most $O(\ln \ln n)$ rounds before switching to State D.
4. **State D** – Stops pushing and pulling the rumors.

The counters help determine when to switch from an exponential growth phase to a quadratic shrinking phase. They maintain that even if the counter mechanism fails, they set a hard stop on nodes spreading the rumor after $O(\ln n)$ rounds, which seems kind of cheap in an algorithm, as it seems like some safety mechanism for if the algorithm doesn't work. There is a poorly drawn state transition diagram in Appendix A, which we omit here due to space reasons.

We will prove the correctness of the algorithm later.

5. Overview of the proofs

We expand on the proofs given in the paper.

Theorem 3.1.1.

Split the phases into 4 phases of rounds and analyze each phase:

1. [Starting out] Starting out, one node knows the rumor and we can roughly double the number of informed nodes on each round
2. [Exponential Growth] The number of nodes that know the rumor grow fast. Informed nodes are either called on and their rumors are pulled or they send their rumors to unknowledgeable nodes. Similar mechanics to *push* algorithm.
3. [Quadratic Shrinking] The number nodes that don't know the rumor shrink faster. Similar mechanics to *pull* algorithm.
4. [Few Uninformed Remaining] Most nodes are informed at this point, so the probability that an uninformed node is not sent the rumor or pulls the rumor is small.

We will analyze the cost and duration of each stage and combine them in the end.

Correctness of Median-Counter algorithm

The argument follows similarly to the proof for Theorem 3.1.1.

5. Details of the proofs

Note that the proof is somewhat rushed/condensed due to space constraints.

Theorem 3.1.1.

Let S_t and U_t be the set of knowledgeable and unknowledgeable nodes at round t, with $s_t = |S_t|$ and $u_t = |U_t|$. We have 4 phases:

1. **Start-up**
   The first rumor is created in round 1. This phase ends when we have $(\ln n)^4$ knowledgeable nodes. Starting with a single rumor, we can double the number of nodes that know the rumor with probability $1 - \frac{1}{n}$. The probability that we double the rumor in the first c rounds is $1 - \frac{1}{n^c}$, so we double the rumor in c rounds with high probability. Note that in the *push* model, in a round, the expected number of newly informed nodes is $\mathbb{E}[s_{t+1} - s_t] = s_t - \Theta\left(\frac{s_t^2}{n}\right)$, which is roughly doubled on each round for $s_t \leq \sqrt{n}$. Then notice that in the *push&pull* model, it must be at least this in expectation. We then observe that $(\ln n)^4$ is sufficiently small (and for sufficiently large n) we can double s_t within some constant number of rounds c. Since each doubling takes a constant number of steps until we get to $(\ln n)^4$, we can do this within $O(\ln \ln n)$ rounds. This can be seen by the fact that $(\ln n)^4 = 2^{O(\ln \ln n)}$.

2. **Exponential Growth**
   This phase ends when we reach at least $n/\ln n$ informed nodes. Notice in expectation, the number of messages containing the rumor sent in round t is $2s_{t-1}$, as informednodes both push and are pulled. Note that $s_{t-1} \geq (\ln n)^4$.
   From this, we get that there are $m = \left(2 \pm o\left(\frac{1}{\ln n}\right)\right) s_{t-1}$ messages sent with high probability by applying Chernoff bounds:
   $\Pr(X \leq (1-\delta)\mu) \leq e^{-\frac{\mu\delta^2}{2}}$ for $0 < \delta < 1$:
   $$\begin{aligned}\Pr\left(m \leq \left(1 - o\left(\frac{1}{\ln n}\right)\right) 2s_{t-1}\right) &\leq e^{-\frac{2s_{t-1}o\left(\frac{1}{\ln n}\right)^2}{2}} \\ &\leq e^{-(\ln n)^4 o\left(\frac{1}{\ln n}\right)^2} \\ &= e^{-o(\ln n)^2} \\ &= o(1)\end{aligned}$$
   $\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2}{2+\delta}\mu}$ for $\delta > 0$:
   $$\begin{aligned}\Pr\left(m \geq \left(1 + o\left(\frac{1}{\ln n}\right)\right) 2s_{t-1}\right) &\leq e^{-\frac{o\left(\frac{1}{\ln n}\right)^2}{2+o\left(\frac{1}{\ln n}\right)}2s_{t-1}} \\ &\leq e^{-\frac{2(\ln n)^4 o\left(\frac{1}{\ln n}\right)^2}{2+o\left(\frac{1}{\ln n}\right)}} \\ &= e^{\frac{2o(\ln n)^2}{-2+o(\ln n)}} \\ &= o(1)\end{aligned}$$
   Thus, combining the complements of these probabilities, we get the desired result with high probability.

Then we must discount the messages that are sent to nodes that already have the rumor (prob s_{t-1}/n) or to the same node (prob m/n). Then since $s_{t-1} \leq n/\ln n$, the probability that this happens is upper bounded by the following:

$$\begin{aligned}
\frac{s_{t-1}}{n} + \frac{m}{n} &\leq \frac{s_{t-1}}{n} + \frac{\left(2 + o\left(\frac{1}{\ln n}\right)\right)s_{t-1}}{n}, \qquad m = \left(2 \pm o\left(\frac{1}{\ln n}\right)\right)s_{t-1} \\
&\leq \frac{\left(3 + o\left(\frac{1}{\ln n}\right)\right)\frac{n}{\ln n}}{n}, \qquad s_{t-1} \leq \frac{n}{\ln n} \\
&= \frac{3 + o\left(\frac{1}{\ln n}\right)}{\ln n}
\end{aligned}$$

Then we have that

$$\begin{aligned}
\mathbb{E}[s_t] &= s_{t-1} + m\left(1 - \frac{3 + o\left(\frac{1}{\ln n}\right)}{\ln n}\right) \\
&= s_{t-1} + \left(2 \pm o\left(\frac{1}{\ln n}\right)\right)s_{t-1}\left(1 - \frac{3 + o\left(\frac{1}{\ln n}\right)}{\ln n}\right) \\
&\leq s_{t-1} + \left(2 - o\left(\frac{1}{\ln n}\right)\right)s_{t-1} \\
&= s_{t-1}\left(3 - O\left(\frac{1}{\ln n}\right)\right)
\end{aligned}$$

Then from a Chernoff bound (following a similar scheme to above) and the fact that $\mathbb{E}[s_t] \geq (\ln n)^4$, we get

$$s_t = \left(1 \pm o\left(\frac{1}{\ln n}\right)\right)\mathbb{E}[s_t] = s_{t-1}\left(3 \pm O\left(\frac{1}{\ln n}\right)\right)$$

So we get that this phase takes $\log_3 n \pm O(\ln\ln n)$ rounds with high probability. Note that

$$\left(3 \pm O\left(\frac{1}{\ln n}\right)\right)^x (\ln n)^4 = \frac{n}{\ln n}$$

Gives us some form of $x = \log_3 n \pm O(\ln\ln n)$.

3. **Quadratic Shrinking**

This phase ends when there are at most $\sqrt{n}(\ln n)^4$ unknowledgeable nodes. From the quadratic decrease property, which has that if we start a round with ϵn uninformed nodes, each node has an ϵ chance of being uninformed, so we get that the expected number of uninformed players is $\epsilon^2 n$ at the end of the round. If we only observe the pulls:

$$\mathbb{E}\left[\frac{u_t}{n}\right] \leq \mathbb{E}\left[\frac{u_{t-1}}{n}\right] = \left(\frac{u_{t-1}}{n}\right)^2$$

Then from a Chernoff bound, we get that (with high probability):

$$u_t \le \left(1+\frac{1}{\ln n}\right)\frac{(u_{t-1})^2}{n}$$

We get this from using the following Chernoff bound:

$\Pr(X \ge (1+\delta)\mu) \le e^{-\frac{\delta^2}{2+\delta}\mu}$ for $\delta > 0$:

$$\begin{aligned}
\Pr\left(u_t \ge \left(1+\frac{1}{\ln n}\right)\frac{(u_{t-1})^2}{n}\right) &\le e^{-\frac{\left(\frac{1}{\ln n}\right)^2}{2+\frac{1}{\ln n}}\frac{(u_{t-1})^2}{n}} \\
&\le e^{-\frac{\left(\frac{1}{\ln n}\right)^2}{2+\frac{1}{\ln n}}\frac{n \ln n^4}{n}}, \qquad u_{t-1} \ge \sqrt{n}(\ln n)^4 \\
&= e^{-\frac{(\ln n)^2}{2+\frac{1}{\ln n}}} \\
&= e^{\frac{(\ln n)^2}{\ln n - 2}} \\
&= o(1)
\end{aligned}$$

Then the complement gives us the desired result. Then we need to show that we need $O(\ln\ln n)$ rounds until the number of unknowledgeable nodes decreases from $\frac{n}{\ln n}$ to $\sqrt{n}(\ln n)^4$ (for sufficiently large n [this isn't even lower until like $n \ge 3.4306 \times 10^{15}$]). To do this, it suffices to show that with $u_0 = \frac{n}{\ln n}$, $u_t = \sqrt{n}(\ln n)^4$, $t = O(\ln\ln n)$

$$u_t \le \left(1+\frac{1}{\ln n}\right)\frac{(u_{t-1})^2}{n}$$

holds. The original paper claims that this is an easy and simple calculation. We as a reviewer are unsure whether this is actually an easy calculation or if we are just bad at math. We begin

$$\begin{aligned}
u_t &\le \left(1+\frac{1}{\ln n}\right)\frac{(u_{t-1})^2}{n} = \left(\frac{\ln n + 1}{\ln n}\right)\frac{(u_{t-1})^2}{n} = \left(\frac{\ln n + 1}{n \ln n}\right)(u_{t-1})^2 \\
&\le \left(\frac{\ln n + 1}{n \ln n}\right)^3 (u_{t-2})^4 \\
&\le \left(\frac{\ln n + 1}{n \ln n}\right)^{\sum_{i=0}^{O(\ln\ln n)} 2^i} (u_0)^{2^{O(\ln\ln n)}} \\
&\le \left(\frac{\ln n + 1}{n \ln n}\right)^{2^{O(\ln\ln n)+1}} (u_0)^{2^{O(\ln\ln n)}} \\
&\le \left(\frac{\ln n + 1}{n \ln n}\right)^{2O(\ln n)} (u_0)^{O(\ln n)} \\
\sqrt{n}(\ln n)^4 &\overset{?}{\le} \left(\frac{\ln n + 1}{n \ln n}\right)^{2O(\ln n)} \left(\frac{n}{\ln n}\right)^{O(\ln n)}
\end{aligned}$$

We as a reviewer cannot confirm whether this is true or not, so we leave this proof as an exercise for the reader.

4. **Final Phase**
   Since we have that the number of unknowledgeable nodes is less than $\sqrt{n}(\ln n)^4$. We get that, just from pulling, players have a probability of $\frac{n-\sqrt{n}(\ln n)^4}{n} \approx 1$ [for large enough n] of getting the rumor in each round. Thus, we only need a constant number of rounds until all players are informed.

The exponential-growth phase takes $\log_3 n + O(\ln \ln n)$ rounds and only sends $O(n)$ messages. Then the other phases run in $O(\ln \ln n)$ rounds and if they send $O(n)$ messages on each round, we get that we can spread the rumor to every node in $\log_3 n + O(\ln \ln n)$ rounds and $O(n \ln \ln n)$ transmissions with high probability.

Median-Counter algorithm

Due to paper length constraints, we omit exact proof of the median-counter algorithm. But we draw some parallels between it and the *push&pull* scheme. The counter on State B is used to determine when the exponential growth ends and when the quadratic shrinking phase starts. When a counter value for State B reaches ctr_{max}, this tells us that $n/\text{polylog}(n)$ of the nodes are knowledgeable with high probability, so we enter the quadratic shrinking phase and we only need to run for $O(\ln \ln n)$ more rounds. Then we run some constant number more rounds to conclude with a high probability that all nodes are informed.

7. Empirical Results

We wrote a simulation of *push*, *pull*, and *push&pull* schemes. We (unrealistically) run the simulation until all nodes are informed rather than for a bounded number of steps. We do this because we wanted to get intuition for the end behavior and how many extra rounds it would take to inform all nodes. Note that we also implemented the *median-counter* algorithm, but we omit the *median-counter* algorithm from these results. We decided to do this as we did not find time to correctly tune the constant factors of the given running times.

Our simulation is a bit unrealistic for the following reasons:

- In a distributed system, no node will know that all other nodes know of the rumor unless there is explicit knowledge of state about the other nodes that is kept and disseminated, which is why a termination algorithm like the median counter algorithm or a termination condition is important. (Thus, having a good termination algorithm is important, as we want to stay close to the edge in terms of round numbers for which it is likely that all nodes are informed of the rumor.) Instead, we opt for having a master with a global view of the system that tells us when to stop.
- We sample and apply the same communication partner assignments for each scheme each round, e.g. if node 7 was assigned node 19 as the neighbor for one round, it would be the same for all the schemes. Note that the outcomes may be different based on the different schemes and some schemes could have already had all the nodes informed and don't need to do more work. We also select the same initial node as the rumor originator.
  - This allows us to apply the same setting to all the schemes and see how they perform relatively against each other.

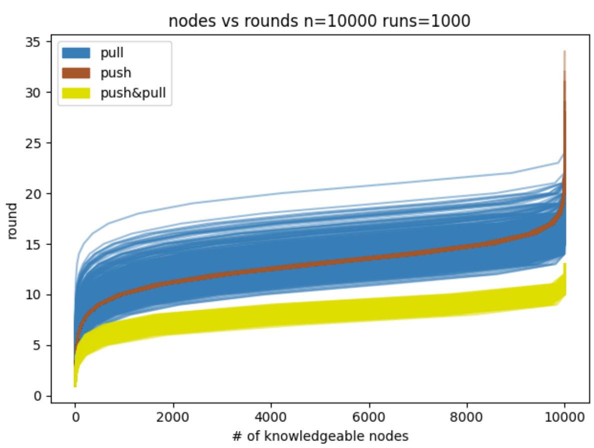
We model the transmission counts (*push* and *pull*) for each scheme in the following manner:

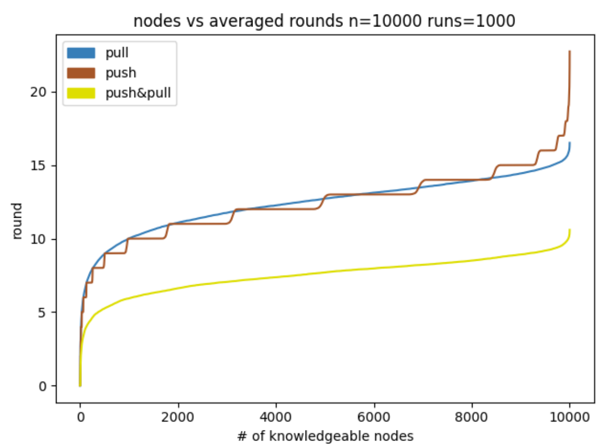
- *push*: Only push if we have the rumor (at most one transmission per node per round)
- *pull*: Only pull if we don't have the rumor (at most one transmission per node per round)
- *push&pull*: Only push if we have the rumor (one transmission) and only pull if we don't have the rumor (one transmission), which differs from the description in the paper where it assumes that we pull even when we have the rumor and push when we don't.
- *median-counter* [if we had included it]: Only pull if we didn't have the rumor (one transmission), both push and pull if we had the rumor (2 transmissions)

We record the following for the setting of n = 10,000 nodes and 1,000 runs of the schemes:

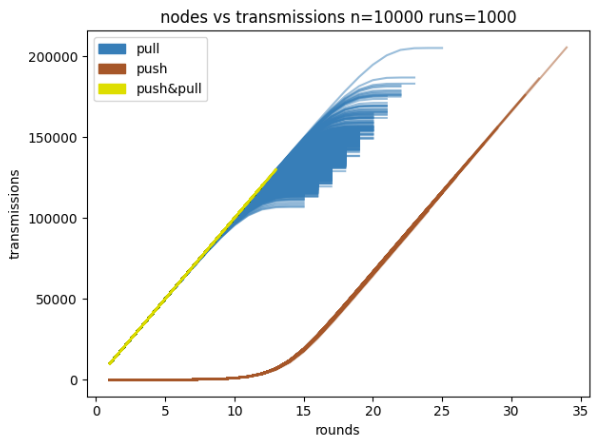
- Graphs plotting each of the 1,000 runs for each scheme against each other with:
    - Number of knowledgeable nodes vs. round number
    - Number of rounds vs. number of transmissions for rounds
- Graphs plotting the average of the 1,000 runs for each scheme against each other with:
    - Number of knowledgeable nodes vs. round numbers averaged over runs
    - Number of rounds vs. number of transmissions for rounds averaged per round in which it appears
- Histograms with transmission and number of rounds it took until all nodes were informed

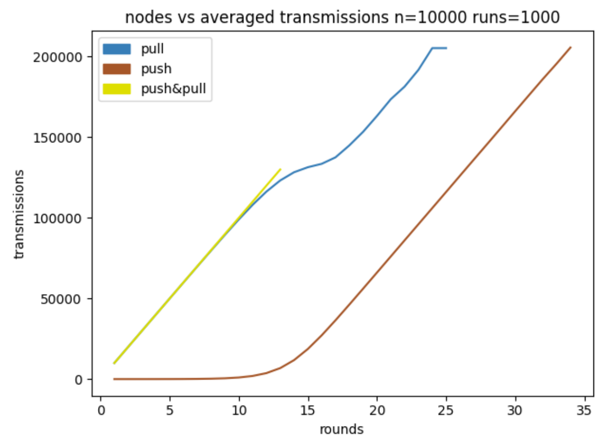
We present the visualizations and include observations we make below them.



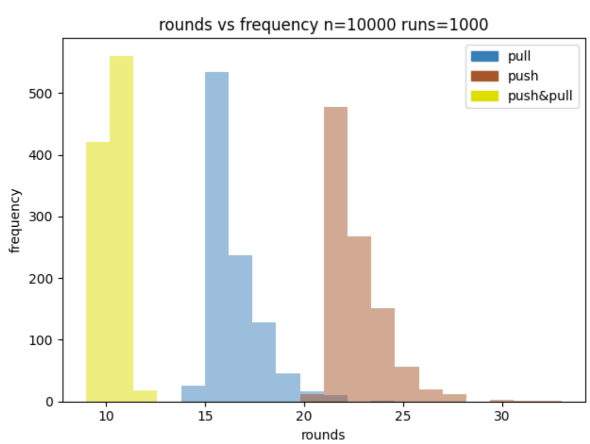


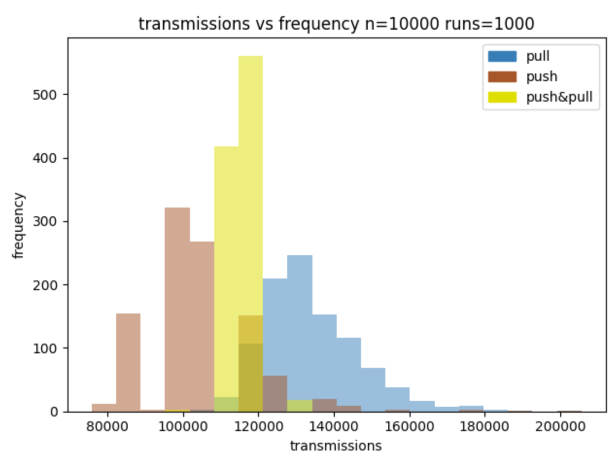
We observe that the variance of the *pull* scheme is higher than that of the *push* and *push&pull* schemes, usually varying significantly in terms of how long it takes to start to pick up. We also notice that the *push* scheme seems to increase dramatically towards the end, which makes sense, as the probability of hitting a node that doesn't know the rumor already decreases as there are more knowledgeable nodes. Note that there are ridges in the averaged graph because some numbers for the number of knowledgeable nodes are never hit, so the average stays the same over some periods. We see that the *push&pull* consistently takes less rounds than the *pull* and *push* schemes separately, which makes sense, as it combines beneficial properties of both schemes.





We observe that the number of transmissions varies a lot towards the end for *pull* runs, likely dependent on how long it takes to start up. We note that *push&pull* and *pull* are initially generally linear, as all nodes participate in the message transmissions, either from pushing or pulling. Then it takes a bit more time for the push scheme to pick up in the number of transmissions, as it takes a while for the exponential growth pattern to grow sufficiently. The average calculation that is performed on the right ends up not being a great average, as it doesn't generate the median case behavior over the runs, but the average of transmissions as it goes over the runs. There is also some amount of clutter within the left graph, as the overlaid *push* runs may hide the fact that some *push* runs ran in less rounds. We see that the transmission patterns for *push&pull* and *push* are generally consistent over the number of rounds, with more variance for the pull scheme.





These graphs parrot some previous results, where we observe that the *push&pull* scheme takes the lowest number of rounds, followed by the *pull*, and then by the *push* schemes. We see that the number of transmissions in the *push* scheme is generally lower than the number of transmissions in the *pull* scheme, which was unclear in the previous pair of graphs. This intuitively makes some sense, as the work done by the *pull* scheme is mostly wasted in the beginning, while the work done by the *push* scheme is mostly wasted towards the end, when it has more nodes trying to transmit the rumor to unknowledgeable nodes. This may indicate that the quadratic-shrinking property takes longer to start up than the exponential-growth phase takes to pick up.

We will talk about some ideas for possible improvements that can be made.

8. Future Directions

We note that to obtain a transmission-optimal algorithm in the same setting, we can just keep state on which nodes we have seen and only have the rumor originator spread the rumor when they connect with a node that they have not seen before. This would reduce the problem down to coupon collections with $n-1$ transmissions and would run in time $\Theta(n \log n)$ rounds.

Currently, the algorithm does not use much state besides the counter and whether it has the rumor or not. We may consider an algorithm that uses and maintains more state information, like keeping track internally of how many distinct nodes we know that know the rumor and transmitting that information along with the rumor. The issue with this is that once a node finds out that every node in the system knows the rumor, how does it inform all the other nodes? It must essentially spread a new rumor of its own.

Alternatively, we can use this method of keeping track of whether other nodes know the rumor and from there, determine a good time to stop when we reach or maintain some level of knowledge about the number of nodes in the system that have heard the rumor. While this does not deterministically stop when every node has heard the rumor, we may be able to get some bounds on this process.

One last thing could be to prove more reliable bounds that don't need to go start being true after $n \geq 3.4306 \times 10^{15}$.

9. Conclusion

We learned more about rumor spreading in a very constrained setting, along with the patterns associated with different spreading schemes. We also back up some theoretical conclusions with experimental results. We believe that this has given us some insight on how information is spread in settings where we only have one random communication partner at a time.

To conclude, learning more about how information is spread is important, as we can use what we learn about these settings to create more realistic models for simulations that can be applied to the world and eventually to people.

References

[1] R. Karp, C. Schindelhauer, S. Shenker and B. Vocking, "Randomized rumor spreading," Proceedings 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 2000, pp. 565-574, doi: 10.1109/SFCS.2000.892324.

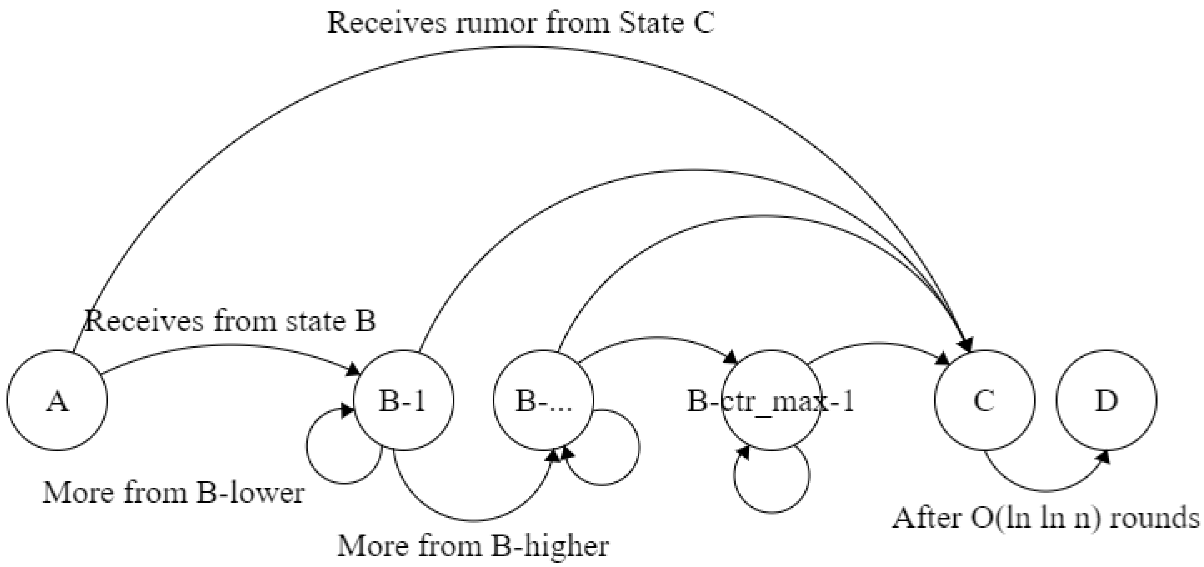
[2] James C. Corbett, et al. "Spanner: Google's Globally-Distributed Database." OSDI.

[3] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang and L. Zhang, "A Brief Introduction to Named Data Networking," MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 2018, pp. 1-6, doi: 10.1109/MILCOM.2018.8599682.

[4] G. Shi, M. Johansson and K. H. Johansson, "Randomized gossiping with unreliable communication: Dependent or independent node updates," 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 2012, pp. 4846-4851, doi: 10.1109/CDC.2012.6426729.
[5] G. Shi, A. Proutiere, M. Johansson and K. H. Johansson, "Randomized consensus with attractive and repulsive links," 52nd IEEE Conference on Decision and Control, Firenze, Italy, 2013, pp. 2599-2604, doi: 10.1109/CDC.2013.6760274.
[6] S. Boyd, A. Ghosh, B. Prabhakar and D. Shah, "Analysis and optimization of randomized gossip algorithms," 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), Nassau, Bahamas, 2004, pp. 5310-5315 Vol.5, doi: 10.1109/CDC.2004.1429652.
[7] Dongsheng Peng, Weidong Liu, Chuang Lin, Zhen Chen and Jiaxing Song, "A loosely synchronized gossip-based algorithm for aggregate information computation," 2008 33rd IEEE Conference on Local Computer Networks (LCN), Montreal, QC, Canada, 2008, pp. 451-455, doi: 10.1109/LCN.2008.4664203.
[8] Alan Demers, Dan Greene, Carl Houser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1988. Epidemic algorithms for replicated database maintenance. SIGOPS Oper. Syst. Rev. 22, 1 (Jan., 1988), 8–32. DOI:https://doi.org/10.1145/43921.43922

Appendix

A. State transition diagrams for Median-Counter algorithm



Self-loops on States B-*m* occur when there are more rumors heard from other nodes in State B with a value lower than the current value maintained than nodes in State B with higher values. Then it transitions to the next State B when it hears from more nodes in State B with a higher value than or equal value with the current value maintained.