Push sum with transmission failures

Balázs Gerencsér* Julien M. Hendrickx*

May 4, 2015

Abstract

The *push-sum* algorithm allows distributed computing of the average on a directed graph, and is particularly relevant when one is restricted to one-way and/or asynchronous communications. We investigate its behavior in the presence of unreliable communication channels where messages can be lost.

We show that convergence still holds, and analyze the error of the final common value we get for the essential case of two nodes, both theoretically and numerically. We compare this error performance with that of the standard consensus algorithm. For the multi-node case, we deduce fundamental properties that implicitly describe the distribution of the final value obtained.

1 Introduction

The ongoing active research on decentralized systems and distributed computation constantly faces the mathematical challenge of aggregating information spread across a huge network [1], [2], [3]. A fundamental case is the task of averaging certain values obtained at the nodes of the network, which is known as the average consensus problem [4]. The difficulty of this task heavily depends on the assumptions on the communication network.

In the simplest case, nodes communicate according to a graph synchronously and without error, that is, they expose their current values to their neighbors and in turn they update their own values using a linear combination of the values they have access to. The system is said to reach average consensus if the values of all nodes converge to the average of the initial values. In this case this can be achieved by properly choosing the coefficients for the linear combinations [5]. We will refer to these as *traditional consensus* methods.

It is known to be impossible in general to reach average consensus using such methods if the communication network is directed and coxsmmunication is asynchronous, as the average usually cannot be kept constant over time in these situations. One way of overcoming this issue is to use the *push-sum algorithm*: nodes not only record a linear combination of other nodes' values, but also keep track of the "amount" of information they actually have. Therefore, two variables per node are required. The algorithm can be asynchronous, with a randomly chosen node communicating towards another at every step, without waiting for a reply. This method is known to efficiently compute the perfect average [6], but only if all transmitted messages are guaranteed to arrive.

There is a strong research effort to understand the performance of these averaging methods while incorporating deficiencies of the communication channels. Such limitations include packet delays [7], packet drops [8], changing connection topology [9], [10], limited or noisy communications [11],[12] or multiple of these [13]. The issue we currently focus on is the presence of transmission failures, resulting in some messages being lost. For a traditional consensus protocol, it is known that the consensus value remains unchanged, only the speed of convergence is slowed down. This is not the case for the push-sum algorithm, an error from the exact average is expected to appear. A clever solution relying on sending the sum of all messages that would have been sent in a classical push-sum algorithm was proposed in [14]. However, this method requires additional capabilities for the nodes; they need to have local identifiers for each potential neighbor, and also an extra memory slot for each communication link.

[*] B. Gerencsér and J. M. Hendrickx are with ICTEAM Institute, Université catholique de Louvain, Belgium balazs.gerencser@uclouvain.be and julien.hendrickx@uclouvain.be Their work is supported by the DYSCO Network (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian Federal Science Policy Office, and by the Concerted Research Action (ARC) of the French Community of Belgium.

In this work, we follow an alternative approach, we analyze the performance of the original push-sum algorithm in the presence of such transmission failures if no additional corrective mechanism is applied, similarly to what was done for traditional consensus in [15]. We present new tools that help understanding the nature of the resulting consensus value. Using these tools, we then derive bounds on the error in the simple case of two nodes. In addition, we perform a numerical comparison of the push-sum algorithm and the traditional consensus method, allowing to determine which is the most efficient one depending on the transmission failure rate and the error tolerance.

The rest of the paper is organized as follows. In Section 2 we formally describe the push-sum algorithm. Section 3 provides the tools needed to preform our analysis. We then develop error bounds in Section 4 for two nodes. Comparison of the push-sum algorithm and traditional consensus is performed in Section 5. Conclusions and further research directions are discussed in Section 6.

2 The push-sum algorithm

In the general setting the push-sum algorithm can be described as follows. Given are n agents with a connection graph and we want the agents to reach average consensus using only limited communication.

Every agent stores a value $x_i(t)$ and also an abstract weight variable $w_i(t)$. The values $x_i(0)$ are the initial measurements, these are the ones to be averaged. The weights are initialized as $w_i(0) = 1$ for all i. At every time step, an (i, j) edge is chosen randomly according to a fixed distribution, in an independent way. Then agent i sends half of what he has (both in terms of value and weight) to agent j. To be more precise, we do the following update:

$$\begin{aligned} x_i(t+1) &= x_i(t)/2, \quad & x_j(t+1) &= x_j(t) + x_i(t)/2, \\ w_i(t+1) &= w_i(t)/2, \quad & w_j(t+1) &= w_j(t) + w_i(t)/2. \end{aligned} \tag{1}$$

The behavior of the algorithm strongly depends on the connection graph and on the way we pick the edges at every step. It is known to work well in a quite general setting even with multiple simultaneous communications [16], meaning that the rescaled measurements approach the average value at every node:

$$\lim_{t\to\infty} \frac{x_i(t)}{w_i(t)} = \frac{\sum x_i(0)}{n}, \qquad \forall i,\ a.s.$$

We analyze the effect of transmission errors: some messages sent might not reach their destination, without the sender being aware of the failure, resulting in a loss of information. Formally, at every time step with probability p we use the update rule (2) below instead of the usual (1), which now corresponds to node i sending a message without node j receiving it.

$$\begin{aligned} x_i(t+1) &= x_i(t)/2, \quad & x_j(t+1) &= x_j(t), \\ w_i(t+1) &= w_i(t)/2, \quad & w_j(t+1) &= w_j(t). \end{aligned} \tag{2}$$

In the sequel we will study the asymptotic behavior of this system for a special case. From now on, we assume the connection graph is complete and edge selection is uniform. We will show that ratio converges even though weights and values go to 0, and analyze then the error between the limiting ratio and the true average. We propose general tools for treating this process with transmission errors. We exploit them in the simple case of two agents to get concrete error bounds.

3 The invariance relation

In this section we will prove an invariance relation that will be of key importance for our analysis. Before getting there, we need to establish basic properties of the push-sum algorithm with transmission failures.

3.1 Convergence

First we show that the push-sum algorithm converges exponentially, meaning that the scaled measurements of different agents will be the same in the limit. Observe that all values and weights go to zero, nevertheless the ratio converges to a meaningful value. Here we only aim to confirm convergence, and do not search for the best rate possible.

Theorem 1. *The push-sum algorithm converges exponentially fast almost surely.*

Proof. Let us denote the maximal rescaled measurement $x_i(t)/w_i(t)$ at time t by $M(t)$ and the minimal one by $m(t)$. Observe that the steps we perform only allow $M(t)$ to decrease and $m(t)$ to increase. We prove the theorem by showing the following equation for some $\gamma > 1$:

$$P(\limsup_{t\to\infty}(M(t) - m(t))\gamma^t \leq 1) = 1. \tag{3}$$

We first point out the opportunity when $M(t) - m(t)$ can decrease. Fix some time t and choose the agent i with the largest weight $w_i(t)$. There is a certain positive probability q that in the following $n-1$ steps, agent i successfully sends one message to every other agent, and that these are the only communication steps in this time frame. Let us call this broadcast-style event $B(t)$. When $B(t)$ occurs, this will pull all the scaled measurements towards $x_i(t)/w_i(t)$, every node will receive at least a 2^{-n+1} portion of $w_i(t)$. If some portion α was received, we have

$$\frac{x_i(t+n-1)}{w_i(t+n-1)} - \frac{x_j(t+n-1)}{w_j(t+n-1)} = \frac{x_i(t)}{w_i(t)} - \frac{x_j(t) + \alpha x_i(t)}{w_j(t) + \alpha x_j(t)} = \frac{x_i(t)w_j(t) - x_j(t)w_i(t)}{w_i(t)(w_j(t) + \alpha w_i(t))}.$$

We have to compare this with

$$\frac{x_i(t)}{w_i(t)} - \frac{x_j(t)}{w_j(t)} = \frac{x_i(t)w_j(t) - x_j(t)w_i(t)}{w_i(t)w_j(t)}.$$

Knowing $w_i(t) \geq w_j(t)$ and the lower bound on α we get for every $j \neq i$ that

$$\left|\frac{x_i(t+n-1)}{w_i(t+n-1)} - \frac{x_j(t+n-1)}{w_j(t+n-1)}\right| \leq \frac{1}{1+2^{-n+1}} \left|\frac{x_i(t)}{w_i(t)} - \frac{x_j(t)}{w_j(t)}\right|.$$

Let us introduce $\rho = 1/(1+2^{-n+1})$. As every rescaled measurement moves towards $x_i(t)/w_i(t)$, so does the maximal and the minimal one. This means we can conclude

$$M(t+n-1) - m(t+n-1) \leq \rho(M(t) - m(t)),$$

again assuming the event $B(t)$. The events $B(t_1)$ and $B(t_2)$ are independent whenever $|t_1 - t_2| \geq n-1$, as they can be expressed using two disjoint sets of independent transmission decisions. For some k, let us count how many of the events $B(0), B(n-1), \ldots, B((k-1)(n-1))$ happen and denote this number by $N(k)$. From standard concentration techniques involving Chernoff bounds we see that

$$P\left(N(k) < \frac{1}{2}qk\right) \leq \exp\left(-\frac{1}{8}qk\right).$$

This implies using $t = k(n-1)$ that

$$P\left(M(t) - m(t) \geq (M(0) - m(0))\rho^{\frac{qt}{2(n-1)}}\right) \leq \exp\left(-\frac{qt}{8(n-1)}\right).$$

As the right hand side sums to a finite number for $t \to \infty$, the events on the left hand side will happen almost surely finitely many times. This confirms the theorem for $\gamma < \rho^{-\frac{q}{2(n-1)}}$. □

In the case when transmission errors cannot occur, convergence can be shown in a much more general setting [16], [17].

3.2 Final values

In the previous subsection we have seen that the push-sum algorithm always reaches consensus. However, due to the random transmissions and failures this final value is a random variable.

We set up the definitions to study the behavior of the final consensus value acquired from the push-sum algorithm with transmission failures. We introduce two versions to formulate this, first we define a random function T as the *push-sum function* which results in the final consensus value of a push-sum algorithm given the initial measurements of the agents. We will then define the *push-sum coefficient* which captures how the push-sum algorithm mixes the initial values in a more general setting.

Definition 2. For any real vector space V and a probability space Ω, we call a random function $T : V^n \to V$ a *push-sum function* if it follows the distribution of the output of the algorithm below. We assume there is a probability space Ω and that the algorithm has access to the actual realization $\omega \in \Omega$ to determine the random variables encountered.

To generate $T(\mathcal{V})$ for some $\mathcal{V} = (v_1, v_2, \ldots, v_n)$ initialize $\mathcal{V}(0) = \mathcal{V}$, $w(0) = \mathbf{1} \in \mathbb{R}^n$. At every time step t, choose $i \neq j$ indices in a random uniform way and perform the following updates to get $\mathcal{V}(t+1) = (v_1(t+1), v_2(t+1), \ldots, v_n(t+1))$ and $w(t+1) = (w_1(t+1), w_2(t+1), \ldots, w_n(t+1))$:

$$\begin{aligned} v_i(t+1) &= v_i(t)/2, & v_j(t+1) &= v_j(t) + \chi_{t+1} v_i(t)/2, \\ w_i(t+1) &= w_i(t)/2, & w_j(t+1) &= w_j(t) + \chi_{t+1} w_i(t)/2. \end{aligned}$$

Here χ_{t+1} is an independent binary random variable, $\chi_{t+1} = 1$ if the transmission is successful and $\chi_{t+1} = 0$ if not. We encode the failure probability p by setting $P(\chi_{t+1} = 0) = p$, $P(\chi_{t+1} = 1) = 1 - p$. For all indices k other than i, j, the update is identical:

$$\begin{aligned} v_k(t+1) &= v_k(t), \\ w_k(t+1) &= w_k(t). \end{aligned}$$

Finally, let

$$T(\mathcal{V}) = \lim_{t \to \infty} \frac{v_i(t)}{w_i(t)}. \tag{4}$$

Observe that this is well defined almost surely as the limit is the same for all indices i. This follows from the convergence of the push-sum algorithm (Theorem 1). Still, the function value is random, depending on the choice of indices and χ_{t+1} at each time step.

To simplify the representation we define the random operator $S : V^n \times \mathbb{R}^n \to V^n \times \mathbb{R}^n$ corresponding to a single independent step of the push-sum algorithm such that

$$(\mathcal{V}(t+1), w(t+1)) = S(\mathcal{V}(t), w(t)).$$

Also, let us introduce the operator $N : V^n \times \mathbb{R}^n \to V$ for the final evaluation of the push-sum algorithm:

$$N(\mathcal{V}(t), w(t)) = \frac{v_1(t)}{w_1(t)}.$$

Knowing that the final ratios are the same for all agents, it does not matter which v_i/w_i we use, so we choose v_1/w_1 for simplicity. Using the new notations we can rewrite the push-sum function T as

$$T(\mathcal{V}) = \lim_{t \to \infty} N(S^t(\mathcal{V}, \mathbf{1})). \tag{5}$$

Here with a slight abuse of notation S^t means the composition of t independent copies of S.

Observe that the push-sum function is linear in its input. Therefore we attempt a decomposition as a random linear operator applied to the input vectors. For this we define a special type of push-sum algorithm as follows.

Definition 3. A random variable $\tau \in \mathbb{R}_+^n$ is a *push-sum coefficient* if it follows the distribution of the algorithm below. Again, we assume there is a probability space Ω and that the algorithm has access to the actual realization $\omega \in \Omega$ to determine the random variables encountered.

Initialize $z_i(0) = e_i = (0, 0, \dots, 1, \dots, 0)^T$. At every time step t, choose $i \neq j$ indices in a random uniform way and perform the following updates:

$$\begin{aligned} z_i(t+1) &= z_i(t)/2, \\ z_j(t+1) &= z_j(t) + \chi_{t+1} z_i(t)/2. \end{aligned}$$

Here χ_{t+1} is an independent random variable (corresponding to the success of the transmission), $P(\chi_{t+1} = 0) = p,\ P(\chi_{t+1} = 1) = 1 - p$. For all indices k other than i, j, the update is identical:

$$z_k(t+1) = z_k(t).$$

Finally, let

$$\tau = \lim_{t \to \infty} \frac{z_i(t)}{\|z_i(t)\|_1}.$$

Observe that this is well defined as the limit is the same for all indices i. This follows from the convergence of the push-sum algorithm. Still, the function value is random, depending on the choice of indices and χ_{t+1} at each time step.

Definition 4. In order to simplify our formulas, for any $x \in \mathbb{R}^n$ (possibly random) and n-tuple of vectors $\mathcal{V} = (v_1, v_2, \dots, v_n)$ from some vector space V we define

$$\langle x, \mathcal{V} \rangle = \sum_{i=1}^{n} x_i v_i.$$

We present the connection between push-sum functions and push-sum coefficients.

Proposition 5. *For any vector space V and $\mathcal{V} = (v_1, v_2, \dots, v_n) \in V^n$ define the random function T as*

$$T(\mathcal{V}) = \langle \tau, \mathcal{V} \rangle,$$

where τ is a push-sum coefficient. Then T is a push-sum function.

Proof. Observe that a process generating a push-sum function can be obtained from the algorithm generating τ by the following coupling:

$$\begin{aligned} v_i(t) &= \langle z_i(t), \mathcal{V} \rangle, \\ w_i(t) &= \|z_i(t)\|_1. \end{aligned}$$

It is easy to check that these equations are preserved with each update (we use that all $z_i(t)$ and $w_i(t)$ are non-negative). In the limit, this leads to

$$\lim_{t \to \infty} \frac{v_i(t)}{w_i(t)} = \lim_{t \to \infty} \frac{\langle z_i(t), \mathcal{V} \rangle}{\|z_i(t)\|_1} = \left\langle \lim_{t \to \infty} \frac{z_i(t)}{\|z_i(t)\|_1}, \mathcal{V} \right\rangle = \langle \tau, \mathcal{V} \rangle.$$

We conclude by remarking that the left hand side defines a push-sum function and the right hand side is the definition of T in the proposition. □

Corollary 6. *We may view a push-sum coefficient as a special value of a push-sum function according to the equation*

$$T(\mathcal{E}) = \tau,$$

where $\mathcal{E} = (e_1, e_2, \dots, e_n)$.

Proof. Observe that $\tau = \langle \tau, \mathcal{E} \rangle$ directly from the definition. Combine this with Proposition 5 using $\mathcal{V} = \mathcal{E}$. □

Corollary 6 confirms the intuition that the two representations of the push-sum algorithm are essentially doing the same thing. Note that the construction is slightly different, for push-sum functions we explicitly track the weights $w_i(t)$, while for the push-sum coefficient we just use the norm of the vectors $z_i(t)$.

3.3 Measures

We plan to work with the distribution of the final value of the push-sum algorithm, so we introduce the measures and transformations involved.

Definition 7. Let μ be the distribution of a push-sum coefficient. This measure has a support contained in the unit simplex $\{x \in \mathbb{R}_+^n \mid \|x\|_1 = 1\}$.

We define the induced measure transformation $\langle \sigma, \mathcal{V} \rangle$ for a measure σ on $\mathbb{R}^n$. Formally

$$\langle \sigma, \mathcal{V} \rangle (A) = \sigma\big(\{x \in \mathbb{R}^n \mid \langle x, \mathcal{V} \rangle \in A\}\big).$$

In other words, for any set $A \subset V$ we collect and measure those x for which $\langle x, \mathcal{V} \rangle$ falls in A.

We need the rescaling operator L on $\mathbb{R}_+^n \setminus \{0\}$ as

$$L(x) = \frac{x}{\|x\|_1}.$$

This operator scales its argument to have unit 1-norm. The operator L naturally induces the measure transformation L^*,

$$L^*(\sigma)\ (A) = \sigma\big(\{x \in \mathbb{R}_+^n \setminus \{0\} \mid L(x) \in A\}\big).$$

Now we develop an invariance relation describing μ using the time-homogeneity of the push-sum algorithm. Broadly speaking, we want to exploit that a push-sum algorithm should give the same result if we let it run for one more step.

The main result of this section is the following.

Theorem 8. *After the first step of a push-sum coefficient algorithm we get the n-tuple of vectors $\mathcal{Z}(1) = (z_1(1), z_2(1), \ldots, z_n(1))$. We get an invariance equation for the push-sum coefficient distribution μ:*

$$\mu = \mathbb{E}L^*\left(\langle \mu, \mathcal{Z}(1) \rangle\right). \tag{6}$$

To see the right hand side in a more detail, we first take the inner product of μ with the (random) collection of vectors $\mathcal{Z}(1)$. We map back the resulting measure to the simplex of unit 1-norm non-negative vectors. Finally we take expectation w.r.t. $\mathcal{Z}(1)$.

Proof. Following Corollary 6, $T(\mathcal{E})$ generates a random variable with distribution μ. We launch the push-sum algorithm as specified in Definition 2. It does no harm to take $t+1$ push-sum steps by first taking one, then applying all the remaining steps to the result. Formally, we mean

$$T(\mathcal{E}) = \lim_{t\to\infty} N(S^{t+1}(\mathcal{E}, \mathbf{1})) = \lim_{t\to\infty} N(S^t(\mathcal{Z}(1), w(1))).$$

We define a very similar random variable for comparison. We perform a single step of the push-sum algorithm, which gives us $\mathcal{Z}(1)$. We now treat this as the input for a new push-sum function. Formally, we are looking at

$$T(\mathcal{Z}(1)) = \lim_{t\to\infty} N(S^t(\mathcal{Z}(1), \mathbf{1})).$$

Let us compare the two expressions above.

Claim 1. The vector parts of the two algorithms $[S^t(\mathcal{Z}(1), w(1))]_{\mathcal{V}}$ and $[S^t(\mathcal{Z}(1), \mathbf{1})]_{\mathcal{V}}$ follow the same distribution.

We see two push-sum algorithms, both starting from $\mathcal{Z}(1)$, but with different weights. During the push-sum steps S the weights $w_i(t)$ have no effect on the evolution of the vectors $v_i(t)$, so the vector parts must follow the same distribution.

Claim 2. There exist a scalar ρ which can be random and dependent from our other variables such that

$$T(\mathcal{E}) \overset{d}{=} T(\mathcal{Z}(1))\rho. \tag{7}$$

What happens when we apply the evaluating function N to both $S^t(\mathcal{Z}(1), w(1))$ and $S^t(\mathcal{Z}(1), \mathbf{1})$? We divide the first of the vectors, which are the same, with the first of the weights, which might be different.

Consequently, the directions of the resulting vectors are the same, while the lengths might be different, so there is only a scalar factor difference between the two.

Claim 3. We can also write

$$T(\mathcal{E}) \overset{d}{=} L(T(\mathcal{Z}(1))). \tag{8}$$

From (7) we have

$$\|T(\mathcal{Z}(1))\rho\| \overset{d}{=} \|T(\mathcal{E})\| = 1.$$

Here we use that the push-sum coefficient $T(\mathcal{E})$ always results in a vector of unit 1-norm. This means

$$T(\mathcal{Z}(1))\rho = \frac{T(\mathcal{Z}(1))}{\|T(\mathcal{Z}(1))\|},$$

as both their directions and their norms agree. The right hand side is $L(T(\mathcal{Z}(1)))$ by definition, so we arrive at

$$T(\mathcal{E}) \overset{d}{=} T(\mathcal{Z}(1))\rho = L(T(\mathcal{Z}(1))).$$

Claim 4. The invariance equation (6) holds.

The distribution of the left hand side expression of (8) is μ. We complete the proof by showing that the right hand side of (8) follows the distribution indicated on the right hand side of (6). According to Proposition 5 the right hand side of (8) can be expressed using a push sum-coefficient,

$$LT(\mathcal{Z}(1)) = L\langle \tau, \mathcal{Z}(1)\rangle.$$

Let us express the distribution of this random variable for any realization of $\mathcal{Z}(1)$. The distribution of τ is μ, and the operators acting on the random variables translate to the corresponding measure transformations. We thus arrive at

$$L^*\langle \mu, \mathcal{Z}(1)\rangle.$$

In order to get the overall distribution we have to integrate over possible realizations of $\mathcal{Z}(1)$ which results in

$$\mathbb{E}L^*\langle \mu, \mathcal{Z}(1)\rangle.$$

This is exactly the expression presented on the right hand side of (6).

□

4 Error bound for two agents

We now focus on the case of 2 agents and scalar input, and we are interested how far the final consensus value is from the real average. We quantify this by taking the expectation of the quadratic error. To normalize this error, we assume the input to be $\mathcal{V} = (-1, 1)$ so the real average is 0, and define

$$R = \mathbb{E}T\big((-1,1)\big)^2.$$

We develop lower and upper bounds for this error. In order to do this, we first investigate the push-sum coefficient for 2 agents.

Let us interpret previous results for this particular case. The measure μ is now supported on the segment $(1,0)^\top - (0,1)^\top$ of the real plane. The invariance equation of Theorem 8 becomes

Corollary 9. *For 2 agents the following equation holds for the push-sum coefficient distribution μ:*

$$\mu = \frac{1-p}{2}D_1\mu + \frac{1-p}{2}D_2\mu + \frac{p}{2}F_1\mu + \frac{p}{2}F_2\mu, \tag{9}$$

where D_1, D_2 are the measure transformations corresponding to a delivered message from node 1 to 2 and 2 to 1, respectively, while F_1, F_2 are the transformations for failed transmissions by node 1 and 2, respectively.

Proof. In the two agent setting there are four cases for the first step depending on which node is transmitting and whether it is successful or not. Let us expand the formulation of Theorem 8.

$$\begin{aligned}\mu = \mathbb{E}L^* \left(\langle \mu, \mathcal{Z}(1)\rangle\right) = P(1 \to 2 \text{ success})L^*\Big(\big\langle \mu, \mathbb{E}(\mathcal{Z}(1) \mid 1 \to 2 \text{ success})\big\rangle\Big)\\ + P(2 \to 1 \text{ success})L^*\Big(\big\langle \mu, \mathbb{E}(\mathcal{Z}(1) \mid 2 \to 1 \text{ success})\big\rangle\Big)\\ + P(1 \to 2 \text{ failure})L^*\Big(\big\langle \mu, \mathbb{E}(\mathcal{Z}(1) \mid 1 \to 2 \text{ failure})\big\rangle\Big)\\ + P(2 \to 1 \text{ failure})L^*\Big(\big\langle \mu, \mathbb{E}(\mathcal{Z}(1) \mid 2 \to 1 \text{ failure})\big\rangle\Big).\end{aligned}$$

For the first term, the probability of occurring is $\frac{1-p}{2}$. Conditioning on the event that the first step was a $1 \to 2$ successful transmission, we get $\mathcal{Z}(1) = \big((1/2, 0)^\top, (1/2, 1)^\top\big)$. The measure μ is supported on the segment $(1, 0)^\top - (0, 1)^\top$, so it is easy to check that the inner product $\big\langle \mu, \mathbb{E}(\mathcal{Z}(1) \mid 1 \to 2 \text{ success})\big\rangle$ will be connecting $(1/2, 0)^\top - (1/2, 1)^\top$. As the inner product is linear, this measure will be the same as μ moved by a similarity transformation to span the segment between the two new endpoints. The L^* transformation maps back the measure to $(1, 0)^\top - (0, 1)^\top$. The series of transformations is illustrated in Figure 1. The measure μ on the blue segment is moved to the orange segment, then mapped along the yellow lines back to the blue line.

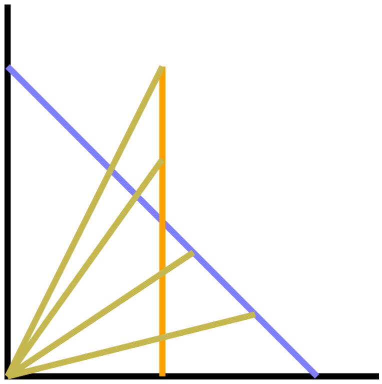

Figure 1: The transformation D_1

The other cases are interpreted in a similar way, only the conditional values of $\mathcal{Z}(1)$ change. For $2 \to 1$ successful transmission it is $\big((1, 1/2)^\top - (0, 1/2)^\top\big)$, for $1 \to 2$ failed transmission it is $\big((1/2, 0)^\top - (0, 1)^\top\big)$, for $2 \to 1$ failed transmission it is $\big((1, 0)^\top - (0, 1/2)^\top\big)$. The visual representation of all four cases is shown in Figure 2.

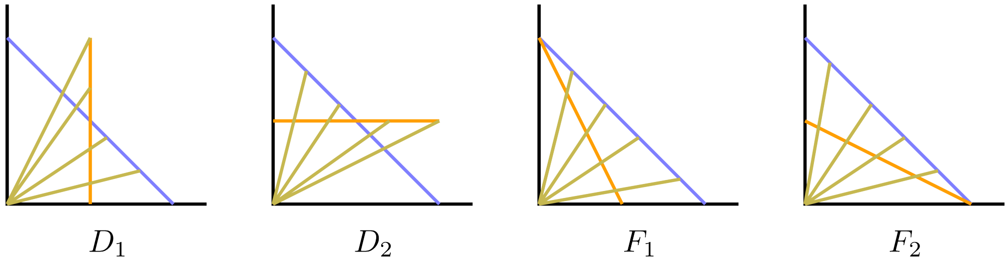

Figure 2: The four transformations

□

In fact μ is inherently a measure on a one dimensional set, so let us interpret it this way. It is convenient to parametrize the diagonal segment $(1, 0)^\top - (0, 1)^\top$ by the first coordinate. As a result we have to work with measures ν on the unit interval $[0,1]$. Working with ν does preserve the symmetry of the problem. Let us restate Corollary 9 in terms of ν.

We need to reformulate D_i, F_i in terms of ν, which we demonstrate for D_1. The point $x \in [0,1]$ corresponds to $(x, 1-x)^\top$ on the diagonal line supporting μ. For the inner product we get

$$\left\langle (x, 1-x), \left((1/2, 0)^\top, (1/2, 1)^\top\right) \right\rangle = (1/2, 1-x)^\top$$

Then using L this is renormalized to have unit 1-norm:

$$\frac{1}{1/2 + 1 - x}(1/2, 1-x)^\top = \left(\frac{1}{3-2x}, \frac{2-2x}{3-2x}\right)^\top.$$

Finally, to switch back to the domain of $[0,1]$ we have to take the first coordinate of the result. In a similar way, we get the four functions corresponding to the preceding four measure transformations:

$$d_1(x) = \frac{1}{3-2x}, \qquad f_1(x) = \frac{x}{2-x},$$
$$d_2(x) = \frac{2x}{1+2x}, \qquad f_2(x) = \frac{2x}{1+x}.$$

Using these functions we are ready to restate Corollary 9:

Corollary 10. *The following invariance equation holds for ν, the push-sum coefficient measure for two agents represented on $[0,1]$.*

$$\nu = \frac{1-p}{2} d_1^*(\nu) + \frac{1-p}{2} d_2^*(\nu) + \frac{p}{2} f_1^*(\nu) + \frac{p}{2} f_2^*(\nu). \tag{10}$$

4.1 Lower bound

The general plan is to develop simple necessary conditions on ν based on (10). We then get a lower bound by optimizing the error on the broader class of measures only constrained by the necessary condition obtained.

The first step of simplification is dropping terms from the right hand side of (10) and using the non-negativity of the terms:

$$\nu \geq \frac{p}{2} f_1^*(\nu) + \frac{p}{2} f_2^*(\nu). \tag{11}$$

Let us introduce a useful subdivision of $[0,1]$: For any $k \in \mathbb{N}$ let

$$a_{-k} = \frac{1}{2^k + 1}, \qquad a_k = 1 - \frac{1}{2^k + 1}.$$

Lemma 11. *For any $k \in \mathbb{Z}$ we have*

$$f_1(a_k) = a_{k-1},$$
$$f_2(a_k) = a_{k+1}.$$

Proof. This can be directly checked using the formulas defining f_i, a_k. □

We disregard the rather complicated fine structure of ν for a moment and only explore how it acts on the intervals defined by a_i. For this, let us define

$$s_i = \nu\big((a_{-i-1}, a_{-i})\big) = \nu\big((a_i, a_{i+1})\big),$$
$$t_i = \nu\big(\{a_{-i}\}\big) = \nu\big(\{a_i\}\big).$$

Here we use the straightforward symmetry of ν and the strict monotonicity of f_i. Combining (11) with Lemma 11 we get the following inequalities.

Lemma 12. *We have*

$$\begin{aligned} s_0 &\geq \frac{p}{2}s_0 + \frac{p}{2}s_1, \\ s_i &\geq \frac{p}{2}s_{i-1} + \frac{p}{2}s_{i+1}, \quad i \geq 1, \\ t_i &\geq \frac{p}{2}t_{i-1} + \frac{p}{2}t_{i+1}, \quad i \geq 0. \end{aligned}$$

We derive an even simpler condition on s_i and t_i.

Lemma 13. *Let* $\gamma = \frac{1-\sqrt{1-p^2}}{p}$. *For any* $i \geq 0$ *we have*

$$\begin{aligned} \gamma s_i &\leq s_{i+1}, \\ \gamma t_i &\leq t_{i+1}. \end{aligned}$$

To give some intuition, γ is $\approx p/2$ for p near 0, but reaches 1 as p increases to 1.

Proof. We show the inequality for s_i, the proof is the same for t_i. Assume the claim does not hold for a given i, meaning $\gamma s_i > s_{i+1}$. We define the auxiliary series $\tilde{s}_i, \tilde{s}_{i+1}, \ldots$ as follows:

$$\tilde{s}_i = s_i, \qquad \tilde{s}_{i+1} = s_{i+1},$$

$$\tilde{s}_j = \frac{2}{p}\tilde{s}_{j-1} - \tilde{s}_{j-2}, \qquad j \geq i+2. \tag{12}$$

We will show that $\tilde{s}_j \geq s_j$ and $\tilde{s}_j \to -\infty$. This would imply $s_j \to -\infty$ which is impossible for entries of a probability distribution.

First, to compare the two series, let us define $d_j = s_j - \tilde{s}_j$. Using Lemma 12 and the definition of $\tilde{s}_j$ the recursion scheme for this series becomes

$$d_i = 0, \qquad d_{i+1} = 0,$$

$$d_j \leq \frac{2}{p}d_{j-1} - d_{j-2}, \qquad j \geq i+2.$$

It is easy to check that $d_j \leq d_{j-1} \leq 0$ implies $d_{j+1} \leq d_j \leq 0$. We immediately get $d_j \leq 0$ in general and

$$\tilde{s}_j \geq s_j, \qquad j \geq i. \tag{13}$$

Now let us compute the series $(\tilde{s}_j)$. It is defined by a second order recursion, so we get the two canonical solutions by solving

$$x^2 = \frac{2}{p}x - 1.$$

The two roots are

$$x_1 = \frac{1-\sqrt{1-p^2}}{p} = \gamma, \qquad x_2 = \frac{1+\sqrt{1-p^2}}{p} = \tilde{\gamma}.$$

These are different except the pathological case of $p = 1$. Any solution of the recursion (12) must be a mixture of γ^j and $\tilde{\gamma}^j$. Then any element of the sequence can be expressed as

$$\tilde{s}_j = \alpha\gamma^{j-i} + \tilde{\alpha}\tilde{\gamma}^{j-i}, \tag{14}$$

for a proper choice of $\alpha, \tilde{\alpha}$. Therefore we need to find $\alpha, \tilde{\alpha}$ such that

$$\begin{aligned} \tilde{s}_i &= \alpha + \tilde{\alpha} \\ \tilde{s}_{i+1} &= \alpha\gamma + \tilde{\alpha}\tilde{\gamma}, \end{aligned} \tag{15}$$

By the condition $\gamma s_i > s_{i+1}$ we get $\tilde{\alpha} < 0$ while solving (15). Adding the fact that $\gamma < 1 < \tilde{\gamma}$, this implies $\tilde{s}_j \to -\infty$.

To sum up, if we had $\gamma s_i > s_{i+1}$, then we would get $\tilde{s}_j \to -\infty$, and by (13) also $s_j \to -\infty$, which is impossible for probabilities. □

Based on this condition, we can prove the following error bound:

Theorem 14. *The following bound holds for the quadratic error R of the push-sum function:*

$$R \geq \gamma - 4(1-\gamma)\sum_{i=1}^{\infty} \frac{(2\gamma)^i}{(2^i+1)^2} \geq \gamma - \frac{8}{9}\gamma(1-\gamma) - \frac{2}{2-\gamma}\gamma^2(1-\gamma).$$

For $p \to 0$, both lower bounds are asymptotically $\frac{p}{18} + O(p^2)$.

The numerical performance of the two bounds is shown in Figure 3.

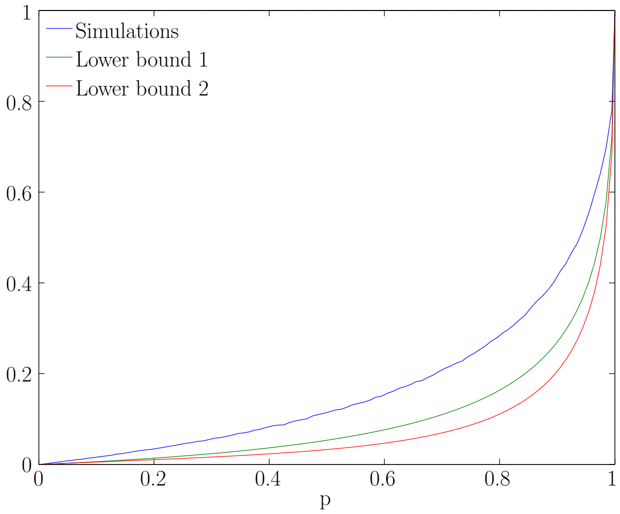

Figure 3: Performance of the two lower bounds presented in Theorem 14 for different transmission failure probabilities p.

Proof. Let us first separate the measure ν as follows:

$$\nu_a = \nu\big|_{\{a_i : i \in \mathbb{Z}\}},$$
$$\nu_c = \nu - \nu_a.$$

In other words, ν_a corresponds to the t_i terms while ν_c corresponds to the s_i terms. For the overall weights we define $M_a = \nu_a([0,1])$ and $M_c = \nu_c([0,1])$. Concerning the errors, we introduce the following notation for the error corresponding to a single point:

$$r(x) = (1-2x)^2.$$

We define R_a, R_c for the overall errors of ν_a, ν_c. As we created a separation of the original measure ν we immediately have

$$1 = M_a + M_c,$$
$$R = R_a + R_c.$$

Let us investigate ν_c. Observe that the error of $\nu\big|_{(a_i,a_{i+1})}$ is larger than the error at a_i with weight s_i $(i \geq 0)$. The symmetrical arguments hold for $i < 0$. We get

$$R_c \geq 2\sum_{i=0}^{\infty} s_i r(a_i),$$

here $r(a_i)$ refers to the error corresponding to the point a_i. We now check how low can the right hand side be while keeping the overall weight constant, that is, $2\sum s_i = M_c = \nu_c([0,1])$.

Let us define the series (s_i^*) such that $2\sum s_i^* = \nu_c([0,1])$ and $\gamma s_i^* = s_{i+1}^*$ for all $i \geq 0$. This can be seen as the extremal series that still satisfies the claim of Lemma13. We will show that the error corresponding to this series is at most that of (s_i). To this end, let us define $d_i = s_i - s_i^*$. Clearly $\sum d_i = 0$. We also have

$$d_{i+1} = s_{i+1} - s_{i+1}^* \geq \gamma s_i - \gamma s_i^* = \gamma d_i.$$

From this we see that $d_i > 0$ implies that all the later terms of (d_i) are also positive. Therefore we see the following structure of the series d_i. It has to begin with some non-positive terms up to some index I to ensure that the sum is 0. After that, all terms are positive.

Furthermore, we see

$$\sum_{i=0}^{I}(-d_i)r(a_i) \leq \sum_{i=I+1}^{\infty} d_i r(a_i). \tag{16}$$

This holds because the sum of the positive weights $-d_i$ on the left hand side and the sum of d_i on the right hand side are the same, but the coefficients $r(a_i)$ are larger on the right hand side.

Now let us compare the error of the two series.

$$\sum_{i=0}^{\infty} s_i r(a_i) - \sum_{i=0}^{\infty} s_i^* r(a_i) = \sum_{i=0}^{\infty} d_i r(a_i) \geq 0.$$

The first equality is simply the definition of d_i, the inequality follows from (16). As (s_i^*) well defined, it gives a lower bound of the quadratic error of ν_c.

$$R_c \geq \sum_{i=0}^{\infty} M_c(1-\gamma)\gamma^i r(a_i).$$

We can treat ν_a the same way using t_i. Note that now there is only one central atom at a_0 while there were two intervals of interest around a_0 for ν_c. Therefore the weights for the series (t_i^*) are slightly different, and we get the lower bound on the error

$$R_a \geq \frac{1-\gamma}{1+\gamma} r(a_0) + \sum_{i=1}^{\infty} 2M_a \frac{1-\gamma}{1+\gamma}\gamma^i r(a_i).$$

Let us now add up the two lower bounds while using that the error at a_0 is 0.

$$R \geq \left(M_c + \frac{2M_a}{1+\gamma}\right)\sum_{i=1}^{\infty}(1-\gamma)\gamma^i r(a_i).$$

Knowing that $M_c + M_a = 1$ and $\gamma \leq 1$ this expression is minimal if $M_c = 1, M_a = 0$. With this setting, we arrive at a universal lower bound on R as follows.

$$\begin{aligned} R &\geq (1-\gamma)\sum_{i=1}^{\infty}\gamma^i (1-2a_i)^2 = (1-\gamma)\sum_{i=1}^{\infty}\gamma^i\left(\frac{2^i-1}{2^i+1}\right)^2 = (1-\gamma)\sum_{i=1}^{\infty}\gamma^i\left(1-\frac{4\cdot 2^i}{(2^i+1)^2}\right) \\ &= (1-\gamma)\sum_{i=1}^{\infty}\gamma^i - (1-\gamma)\sum_{i=1}^{\infty}\gamma^i\frac{4\cdot 2^i}{(2^i+1)^2} = \gamma - 4(1-\gamma)\sum_{i=1}^{\infty}\frac{(2\gamma)^i}{(2^i+1)^2} = \dots \end{aligned}$$

This is exactly the first lower bound we wanted to show. For the second, simpler claim we decrease the denominators $(2^i+1)^2$ to 2^{2i} in order to get a geometric series that is easy to sum.

$$\begin{aligned} \dots &= \gamma - 4(1-\gamma)\frac{2\gamma}{9} - 4(1-\gamma)\sum_{i=2}^{\infty}\frac{(2\gamma)^i}{(2^i+1)^2} \geq \gamma - 4(1-\gamma)\frac{2\gamma}{9} - 4(1-\gamma)\sum_{i=2}^{\infty}\frac{(2\gamma)^i}{2^{2i}} \\ &= \gamma - 4(1-\gamma)\frac{2\gamma}{9} - 4(1-\gamma)\frac{\gamma^2}{4}\frac{1}{1-\frac{\gamma}{2}} = \gamma - \frac{8}{9}\gamma(1-\gamma) - \frac{2}{2-\gamma}\gamma^2(1-\gamma). \end{aligned}$$

This is the second lower bound we were aiming for. The asymptotic rate near $p=0$ for both lower bounds follows easily using $\gamma = p/2 + O(p^2)$ for small p. □

Having a look at Figure 3 we see that the lower bounds we obtained qualitatively capture the real behavior of the algorithm. We get an error linear in the failure rate for $p \approx 0$ and then we get an error approaching 1 as $p \approx 1$. Still, we did some strong simplification steps so quantitatively we do experience a gap between the simulated and the proven values.

4.2 General upper bound

In this subsection we present an upper bound that is valid for all p and that is asymptotically tight both around $p = 0$ and $p = 1$. The numerical performance of the upper bound is shown in Figure 4. Although it shows that the bound is conservative, we already recover the linear growth rate near $p = 0$. In the next subsection we will complement this error bound with one that gives a much better estimate near $p = 1$.

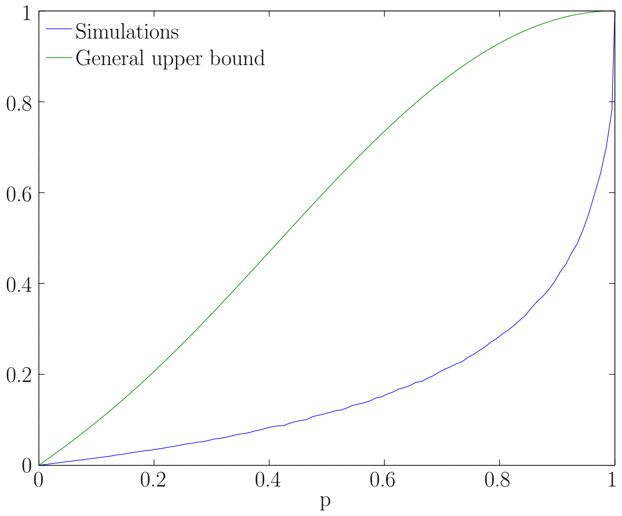

Figure 4: Performance of upper bound stated in Theorem 15

Once again we base our studies on the invariance equation (10). We express ν as a mixture of measures that are supported strictly within $[0, 1]$. In the end we get the bound below.

Theorem 15. *The following bound holds for the error of the push-sum function:*

$$R \leq \frac{p(1-p)^2}{3+p} + \frac{p}{25(1+p^2)} \left(18 + 23p + 50p^2 - 41p^3\right).$$

Proof. We give an intuition on the main ideas used in this proof. For a measure π with restricted support in $[x, y]$ for some $0 \leq x < y \leq 1$, we immediately have an upper bound on the error: we simply find the point of the interval furthest from 1/2, and obtain $\max((1-2x)^2, (1-2y)^2)$.

When we apply one of the transformations corresponding to a transmission on π, we get a new interval containing the new support. By studying the evolution of the interval we get an evolving upper bound.

For a measure π that can be expressed as the mixture of different measures with restricted supports, again we get an upper bound on the error by taking the weighted average of the error bounds for the individual measures according to the idea above.

We will now convert these ideas to precise statements and apply them for the push-sum coefficient measure ν to get an upper bound on the error of the push-sum function.

For measures on $[0, 1]$ we define the following intervals that will serve as possible restrictions on the support. A visual representation is shown in Figure 5.

$$\begin{aligned} \alpha_i &= \left[\frac{2^i}{2^{i+1}+1}, \frac{2^i+1}{2^{i+1}+1}\right], \\ \beta_0' &= \left[\frac{1}{5}, \frac{2}{3}\right], \qquad \beta_1' = \left[0, \frac{2}{3}\right], \\ \beta_0'' &= \left[\frac{1}{3}, \frac{4}{5}\right], \qquad \beta_1'' = \left[\frac{1}{3}, 1\right], \\ \gamma_0 &= \left[\frac{1}{5}, \frac{4}{5}\right], \qquad \gamma_1 = [0,1]. \end{aligned} \tag{17}$$

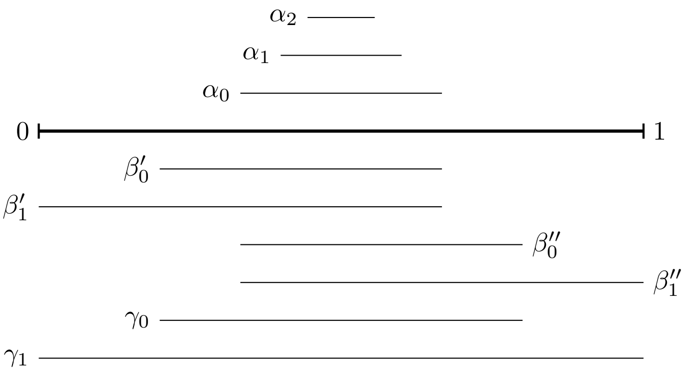

Figure 5: Intervals for supports we focus on.

We construct a related abstract Markov chain. This chain has states $A_0, A_1, \ldots$ and $B_0', B_1', B_0'', B_1'', C_0, C_1$. These will correspond to possible different supports $\alpha_0, \alpha_1, \ldots$ and $\beta_0', \beta_1', \beta_0'', \beta_1'', \gamma_0, \gamma_1$. From each state there are four possible transitions with probabilities $(1-p)/2, (1-p)/2, p/2, p/2$. The transitions are shown in Figure 6 and 7. These will correspond to the four measure transformations used in (10). For example, whenever $supp(\pi) \subseteq \alpha_0$, we have $supp(d_1^*(\pi)) \subseteq \alpha_1$. And indeed, our Markov chain has a transition from A_0 to A_1. Moreover, the transition probability of the Markov chain matches the coefficient of the measure transformation in (10).

This Markov chain is irreducible, aperiodic and positive recurrent. Therefore the distribution will approach the unique stationary distribution from any starting distribution.

For a measure μ on $[0,1]$ we define the following operation to get μ^+ in the spirit of (10).

$$\mu^+ = \frac{1-p}{2} d_1^*(\mu) + \frac{1-p}{2} d_2^*(\mu) + \frac{p}{2} f_1^*(\mu) + \frac{p}{2} f_2^*(\mu), \tag{18}$$

where the transformations d_i^*, f_i^* are the ones in (10).

We link the four possible transformations on some measure μ with the transitions of the Markov chain. We say that the state s and the measure μ are *consistent* with each other whenever

$$\begin{aligned} &\text{if } s = A_i \text{ then } supp(\mu) \subset \alpha_i, \\ &\text{if } s = B_i' \text{ then } supp(\mu) \subset \beta_i', \\ &\text{if } s = B_i'' \text{ then } supp(\mu) \subset \beta_i'', \\ &\text{if } s = C_i \text{ then } supp(\mu) \subset \gamma_i. \end{aligned} \tag{19}$$

Lemma 16. *a) Take one of the states s of the Markov chain and a measure μ on $[0,1]$ that are consistent with each other. There is a pairing of the transitions of the Markov chain and the transformations of measures that correspond to each other, let s^+ be the next state of the Markov chain after a certain transition and μ^+ the measure resulting from the corresponding transformation. Then s^+ is consistent with μ^+.*

b) *Let $X = x_{A_0}, x_{A_1}, \dots$ be a probability distribution on the states of the Markov-chain. Let $\mu_{A_0}, \mu_{A_1}, \dots$ be probability measures consistent with $A_0, A_1, \dots$. Define*

$$\mu = x_{A_0}\mu_{A_0} + x_{A_1}\mu_{A_1} + \dots.$$

Let $X^+ = x^+_{A_0}, x^+_{A_1}, \dots$ be the probability distribution when taking one step of the Markov chain starting from X. Let μ^+ be the mixture of the four transformed version of μ according to (18). *Then there exists probability measures $\mu^+_{A_0}, \mu^+_{A_1}, \dots$ consistent with $A_0, A_1, \dots$ such that*

$$\mu^+ = x^+_{A_0}\mu^+_{A_0} + x^+_{A_1}\mu^+_{A_1} + \dots.$$

Proof. a) This is an elementary but tedious exercise, we only present a list of claims to confirm.

For all pairs α_i, A_i (and similarly for the other states and intervals), one has to find the image of α_i under the four transformations in (18). Also, one has to gather the possible states following A_i for the Markov chain. It has to be verified that the images of α_i fall within the intervals corresponding to the new state of the Markov chain.

Even more, it should be confirmed that the weights match: the measure transformations with coefficients $(1-p)/2$ and $p/2$ in (18) have to correspond to Markov chain transitions with transition probability $(1-p)/2$ and $p/2$.

b) We need to apply *a*) multiple times. For example, assume for a moment that the Markov chain is at B'_0 and we are given a $\mu_{B'_0}$ supported within β'_0. The Markov chain can step to B'_0, A_0, C_0, B'_1 with transition probabilities $(1-p)/2, (1-p)/2, p/2, p/2$. According to *a*) the four measure transformations corresponding to these four transitions will lead to four measures supported within $\beta'_0, \alpha_0, \gamma_0, \beta'_1$.

When we relate the setting to the full Markov chain and μ, we observe that the probability of the Markov chain being in B'_0 is $x_{B'_0}$ and so is the weight of $\mu_{B'_0}$ in the expression of μ.

After a step in the Markov chain, the probability distribution at any node, say A_0, is the aggregated probability of the incoming transitions, from B'_0, B'_1, B''_0, B''_1. In the same way $\mu^+_{A_0}$ is the mixture of measures coming from the corresponding transformations of $\mu_{B'_0}, \mu_{B'_1}, \mu_{B''_0}, \mu_{B''_1}$. Observe that the initial probability distribution of the Markov chain and the weights in the expression of μ match. Even more, the transition probabilities and the mixture weight are the same. Therefore the new probability distribution of the Markov chain an the mixture weights of the new measure must agree as well.

□

Let us now initiate the process with $\mu = \nu$, the push-sum coefficient measure, and with the probability distribution δ_{C_1} for the Markov chain putting all weight on C_1. This μ is indeed supported within $[0,1]$, so this is a consistent choice. Remixing μ according to (18) does not change it as it is the solution of the invariance equation (10) mentioned before. On the other hand, the Markov chain will approach its unique stationary distribution step by step. According to Lemma 16 we get that ν can be expressed as the mixture of measures supported within $\alpha_0, \alpha_1, \dots$ with the weights being the values of the stationary distribution of the Markov chain.

We now calculate this stationary distribution. Let a_i, c_i be the stationary probabilities of being in states A_i, C_i. Using the symmetry of B'_i and B''_i let b_i the stationary probabilities of being in states B'_i or B''_i. Also, let us define the total weights of the different types of states as

$$\begin{aligned} S_A &= \sum_{i=0}^{\infty} a_i, \\ S_B &= b_0 + b_1, \\ S_C &= c_0 + c_1. \end{aligned}$$

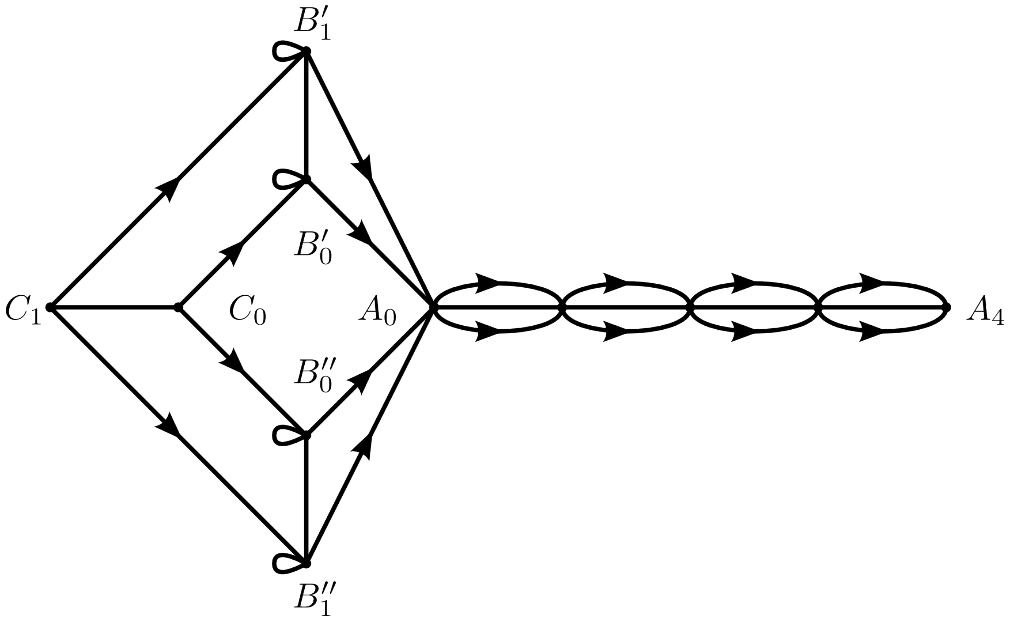

Figure 6: Transitions with probability $(1-p)/2$.

Looking at the scheme of possible transitions, we immediately find that

$$\begin{aligned}
S_A &= (1-p)S_A + \frac{1-p}{2}S_B,\\
S_B &= pS_A + \frac{1}{2}S_B + (1-p)S_C,\\
S_C &= \frac{p}{2}S_B + pS_C.
\end{aligned}$$

Knowing also that these three sum up to one, we get

$$S_A = (1-p)^2, \quad S_B = 2p(1-p), \quad S_C = p^2.$$

For a_i we see the simple recursion $a_i = (1-p)a_{i-1}$. Taking into account their sum S_A we get

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

For the other nodes, we have the equations.

$$\begin{aligned}
b_0 &= pS_A + \frac{1-p}{2}b_0 + (1-p)c_0,\\
c_0 &= \frac{p}{2}b_0.
\end{aligned}$$

These finally lead to

$$\begin{aligned}
b_0 &= \frac{2p(1-p)^2}{1+p^2},\\
b_1 &= \frac{2p^2(1-p^2)}{1+p^2},\\
c_0 &= \frac{p^2(1-p)^2}{1+p^2},\\
c_1 &= \frac{2p^3}{1+p^2}.
\end{aligned}$$

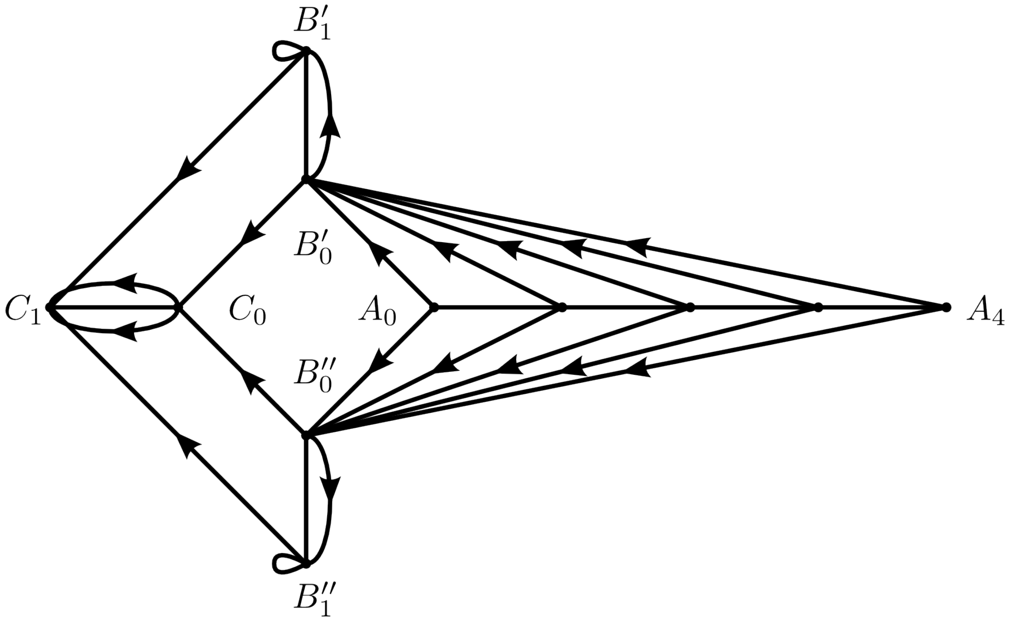

Figure 7: Transitions with probability $p/2$.

We get an upper bound on the error R of the measure ν if we combine upper bounds for the different intervals $\alpha_i, \beta_i', \beta_i'', \gamma_i$ with the weights of the stationary distribution of the Markov-chain. Recall that we have an error bound for a certain interval if we find the furthest point z from $1/2$ and then evaluate $(1-2z)^2$ for the quadratic error. We will use r to denote these errors for the different intervals. By omitting the obvious calculations we get

$$\begin{aligned} r(\alpha_i) &= \frac{1}{(2^{i+1}+1)^2}, \\ r(\beta_0') = r(\beta_0'') &= \frac{9}{25}, \\ r(\beta_1') = r(\beta_1'') &= 1, \\ r(\gamma_0) &= \frac{9}{25}, \\ r(\gamma_1) &= 1. \end{aligned}$$

Finally, let us combine all our estimates.

$$\begin{aligned} R &\leq \sum_{i=0}^{\infty} a_i r(A_i) + b_0 r(B_0) + b_1 r(B_1) + c_0 r(C_0) + c_1 r(C_1) \\ &= \sum_{i=0}^{\infty} \frac{p(1-p)^{i+2}}{(2^{i+1}+1)^2} + \frac{9}{25}\frac{2p(1-p)^2}{1+p^2} + 1\frac{2p^2(1-p^2)}{1+p^2} + \frac{9}{25}\frac{p^2(1-p)^2}{1+p^2} + 1\frac{2p^3}{1+p^2}. \\ &\leq \sum_{i=0}^{\infty} \frac{p(1-p)^{i+2}}{(2^{i+1})^2} + \frac{p}{25(1+p^2)}\left(18+23p+50p^2-41p^3\right) \\ &= \frac{p(1-p)^2}{4}\frac{1}{1-(1-p)/4} + \frac{p}{25(1+p^2)}\left(18+23p+50p^2-41p^3\right) \\ &= \frac{p(1-p)^2}{3+p} + \frac{p}{25(1+p^2)}\left(18+23p+50p^2-41p^3\right). \end{aligned}$$

This is exactly the not so simple formula we were aiming for. □

4.3 Upper bound for high failure rate

When we compare simulation results with the general upper bound on Figure 4, we see that the bound is rather conservative for high p. In this subsection we introduce an alternative upper bound tailored specially for the case of p near 1.

We only investigate the first few steps of the process. We stop after the second successful transmission and check the ratios at the two nodes. We claim that the error corresponding to the ratio further from 0 is a valid upper bound on the final error of this instance. Indeed, the ratios only get closer, so the error of the worse can only get better through the subsequent steps.

We analyze the case where the two successful transmission occur in different directions, we assume (without loss of generality) from node 1 to 2 and then from node 2 to 1. If the two transmissions happen in the same direction, we just assume that the error is trivially bounded by 1. Let us describe the joint distribution of the ratios at the 2 nodes at this point in time.

We focus on the values $x_i(t)$ first. We clearly start with

$$-1 \quad \text{and} \quad 1.$$

Assume that before the first successful transmission, node 1 had k_1 unsuccessful attempts and node 2 had l_1 of those. After the first successful transmission from node 1 to 2 we have the values

$$-2^{-k_1-1} \quad \text{and} \quad -2^{-k_1-1}+2^{-l_1}.$$

We now assume k_2 and l_2 unsuccessful transmissions, respectively. After the second successful transmission, now from node 2 to node 1 we get

$$-2^{-k_1-k_2-1}-2^{-k_1-l_2-2}+2^{-l_1-l_2-1} \quad \text{and} \quad -2^{-k_1-l_2-2}+2^{-l_1-l_2-1}.$$

We can do the same computation for the weights $w_i(t)$, the only difference is that they are both initialized as 1. Note that the weights and values evolve together, so the values k_i, l_i stay the same. For simplicity, we introduce the notation $t_1 = k_1 - l_1$ and $t_2 = l_2 - k_2$. In the end, for the ratios we get

$$\frac{x_1}{w_1} = \frac{-2^{t_2}+2^{t_1}-\frac{1}{2}}{2^{t_2}+2^{t_1}+\frac{1}{2}} \quad \text{and} \quad \frac{x_2}{w_2} = \frac{2^{t_1}-\frac{1}{2}}{2^{t_1}+\frac{1}{2}}.$$

Next we describe the distribution of t_i. The event $\{t_i = a\}$ means that there is some l such that l halvings have been made in one direction, $l+a$ in the other. For non-negative a this leads to the formula

$$P(t_i = a) = \sum_{l=0}^{\infty}(1-p)p^{2l+a}\binom{2l+a}{l}2^{-2l-a}.$$

By symmetry we have $P(t_i = a) = P(t_i = -a)$. Using the generating functions of binomial coefficients the formula can be simplified to

$$P(t_i = a) = \sqrt{\frac{1-p}{1+p}}\left(\frac{1-\sqrt{1-p^2}}{p}\right)^{|a|}. \tag{20}$$

To get an error bound, we have to check which ratio x_i/w_i is the worse based on the values of t_1, t_2, and use the error corresponding to that ratio. Evaluating which ratio is the worse is a simple but somewhat tedious exercise. The summary of the different cases is shown in Figure 8. Knowing that t_1, t_2 are integers We identify the following regions:

a) $t_1 \leq -1$, both ratios are non-positive, the first is the worse.

b) $t_1 \geq 0,\ t_1 \geq t_2 + 1$, both ratios are non-negative, the second is the worse.

c) $t_1 \geq 0,\ t_1 \leq t_2,\ 2t_1 \geq t_2$, one is positive, one is negative, the second is the worse.

d) $t_1 \geq 0,\ 2t_1 + 1 \leq t_2$, one is positive, one is negative, the first is the worse.

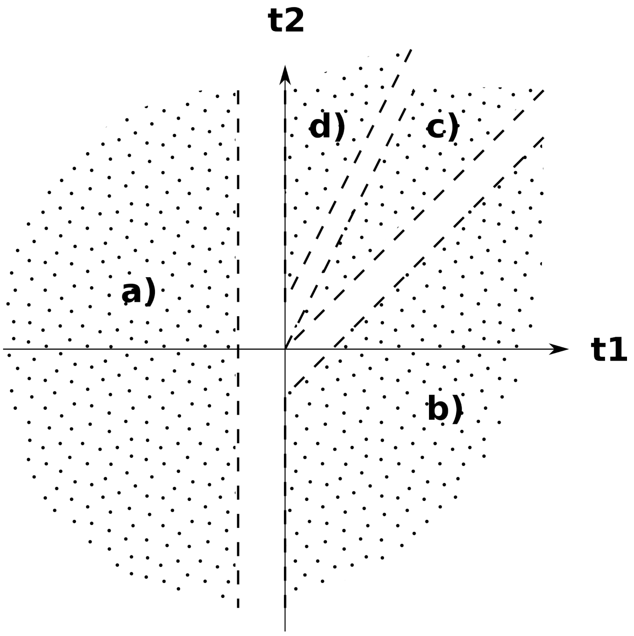

Figure 8: Different cases based on t_1, t_2.

The overall error bound is obtained by summing up all the cases, from which we get

$$R \leq \frac{1}{2} + \frac{1}{2} \sum_{a_1=-\infty}^{\infty} \sum_{a_2=-\infty}^{\infty} P(t_1 = a_1) P(t_2 = a_2) r(a_1, a_2).$$

Here $r(a_1, a_2)$ is the error corresponding to the situation when $t_1 = a_1,\ t_2 = a_2$. These are well defined expressions based on the two ratios and on the actual region for the pair. This is the square of one of the two ratios, depending on which region we are in, as described above. By substituting the formulas for the probabilities we arrive at the following statement.

Theorem 17. *Using the definition of* $r(a_1, a_2)$ *above, we get the following bound for the quadratic error:*

$$R \leq \frac{1}{2} + \frac{1-p}{2(1+p)} \sum_{a_1=-\infty}^{\infty} \sum_{a_2=-\infty}^{\infty} \left(\frac{1 - \sqrt{1-p^2}}{p} \right)^{|a_1|+|a_2|} r(a_1, a_2).$$

The numerical performance of this upper bound is shown in Figure 9. We see indeed that this upper bound captures the behavior of the error when p is close to 1.

5 Comparison of methods

After analyzing the push-sum algorithm, let us compare its behavior to traditional consensus. Let us quickly remind what this algorithm is.

Again, every agent stores its measurement $x_i(t)$. At every time step, an ij edge is chosen randomly, then agent i sends his measurement and thus influences agent j. To be more precise, we do the following update:

$$\begin{aligned} x_i(t+1) &= x_i(t), \\ x_j(t+1) &= (x_j(t) + x_i(t))/2. \end{aligned} \tag{21}$$

It is easy to see that this also converges to consensus. However, even with no communication error the consensus value might deviate from the real average. On the other hand, transmission failures won't increase this error, only slow down the process.

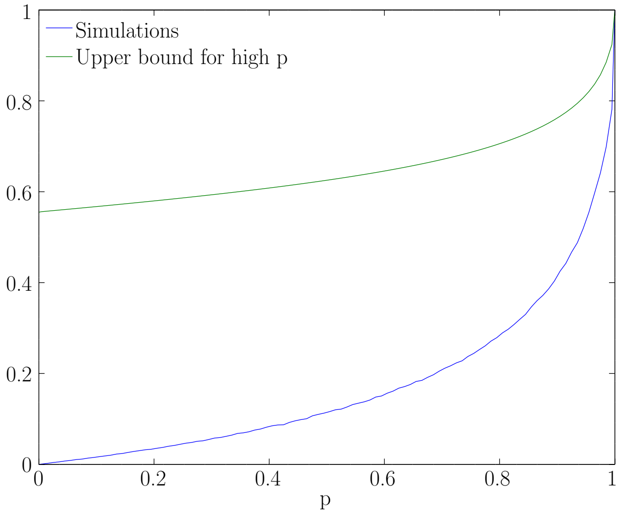

Figure 9: Performance of upper bound for high p

For a good comparison it will be useful to allow a bit of tweaking of both algorithms. In both cases, we have a hard-coded parameter of influence set at $1/2$. This is the ratio sent for the push-sum algorithm and the strength of influence for traditional consensus. We get valid and convergent algorithms if we change this value to some other α.

This way the algorithm has two parameters: the external error probability p and the chosen influence ratio α. We also have two performance metrics: the error from the real average and the speed to reach consensus. Therefore we get a legitimate comparison in the following way. Given an error probability p and a desired error R, we choose α for both algorithms to achieve this error. Knowing that now they perform the same computation, the comparing the speed of the two will tell us which one is more efficient.

Therefore we generate a large number of instances, randomly choosing p and α and compute the speed and error the algorithms give. We generated over 60M instances for the push-sum algorithm, and over 35M for traditional consensus (for the latter we can simplify a bit as p only simply scales the speed without changing the error, we can fix it to 1 and then scale it to other values). In Figure 10 we numerically compare the performance of the two algorithms for a network of fully connected 5 nodes. We identify the following regions:

a) Traditional consensus performs faster. This is the case when we want low error despite the extremely high p.

b) Push-sum is the faster choice.

c) None of the random instances of the push-sum algorithm ended up in this region.

To understand the region c), note that the push-sum algorithm will always give 0 error when $p = 0$, so it is reasonable to see that we do not get high errors when p is very small.

We conclude that the push-sum algorithm is the better alternative for a wide range of setups, but we should be careful when the transmission failure probability p is extremely high.

6 Conclusions

We have analyzed the push-sum algorithm in the presence of transmission failures. This algorithm was originally designed to perform perfect averaging on a network with directed communication. When transmission failures are possible, the values of the nodes of the network still converge to a common value, but this might not be the exact average of the original inputs.

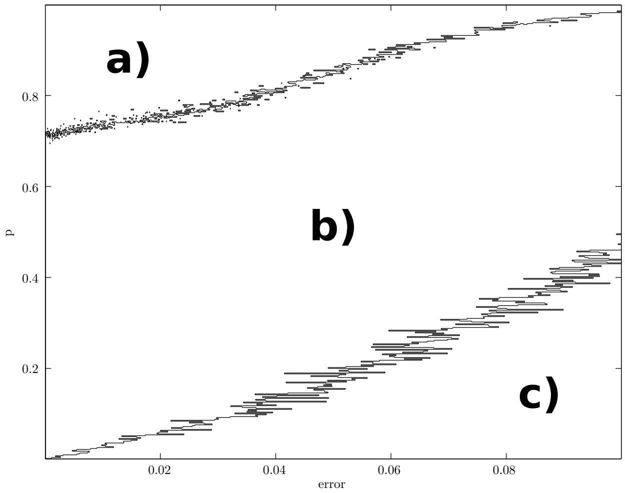

Figure 10: Comparison of push-sum and traditional consensus for 5 nodes

The final value is a random variable determined by the sequence of communication steps and the sequence of transmission failures. We develop new tools to better understand the resulting value, and we form an equation that implicitly describes the distribution of this random variable. Further investigation is performed for the simple case when there are only two nodes, we develop lower and upper bounds on the expected error.

There are very natural follow-up questions to consider for future research. For the case of two nodes, the error bounds do still have a gap between them, there is still room for improvement. It would be even more interesting to extend the results for networks with more nodes. We already have an insight on the distribution of the final value, but it is not straightforward how this could be translated into numerical bounds.

Acknowledgments

We would like to thank Asuman Ozdaglar for the inspiring discussion that eventually lead to this research question being investigated.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[2] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[3] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal of Parallel and Distributed Computing*, vol. 7, no. 2, pp. 279–301, 1989.

[4] J. N. Tsitsiklis, *Problems in decentralized decision making and computation.* PhD thesis, Massachusetts Institute of Technology, 1984.

[5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[6] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 482–491, 2003.

[7] P.-A. Bliman and G. Ferrari-Trecate, "Average consensus problems in networks of agents with delayed communications," *Automatica*, vol. 44, no. 8, pp. 1985–1995, 2008.

[8] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 102–133, 2009.

[9] W. Ren, R. W. Beard, *et al.*, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[10] I. Matei, J. S. Baras, and C. Somarakis, "Convergence results for the linear consensus problem under markovian random graphs," *SIAM Journal on Control and Optimization*, vol. 51, no. 2, pp. 1574–1591, 2013.

[11] M. Huang and J. H. Manton, "Coordination and consensus of networked agents with noisy measurements: stochastic algorithms and asymptotic behavior," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 134–161, 2009.

[12] D. Bauso, L. Giarré, and R. Pesenti, "Consensus for networks with unknown but bounded disturbances," *SIAM Journal on Control and Optimization*, vol. 48, no. 3, pp. 1756–1770, 2009.

[13] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[14] N. H. Vaidya, C. N. Hadjicostis, and A. D. Domínguez-García, "Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity.," in *51st IEEE Conference on Decision and Control*, pp. 2761–2766, 2012.

[15] P. Frasca and J. M. Hendrickx, "Large network consensus is robust to packet losses and interferences," in *Proceedings of the European Control Conference (ECC)*, pp. 1782–1787, 2013.

[16] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 1753–1757, 2010.

[17] F. Iutzeler, P. Ciblat, and W. Hachem, "Analysis of sum-weight-like algorithms for averaging in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2802–2814, 2013.