On Corrective Consensus: Converging to the Exact Average

Yin Chen† Roberto Tron* Andreas Terzis† Rene Vidal*
yinchen@cs.jhu.edu tron@cis.jhu.edu terzis@cs.jhu.edu rvidal@jhu.edu
Computer Science Department† Center for Imaging Science*
Johns Hopins University

Abstract

Consensus algorithms provide an elegant, distributed way for computing the average of a set of measurements across a sensor network. However, the convergence of the node estimates to the global average depends on the timely and reliable exchange of the measurements to neighboring sensors. These assumptions are violated in practice due to random packet losses, causing the estimated average to be biased. In this paper, we present and analyze a practical consensus protocol that overcomes these difficulties and assures convergence to the correct average. Simulation results show that the proposed *corrective consensus* requires ten times less overhead to reach the same level of accuracy as the one achieved by a variant of standard consensus that uses retransmissions to (partially) overcome the negative effects of packet losses. In networks with more severe packet loss rates, corrective consensus is more than forty times more accurate than standard consensus that uses retransmissions. More importantly, by continuing to execute the corrective consensus algorithm the estimation error can become *arbitrarily* small.

1 Introduction

Consider a network of low-power sensing nodes that form an ad-hoc wireless network, each measuring a quantity of interest. Furthermore, we are interested in computing the mean of the network's measurements. Collecting the measurements to a central location and disseminating the result to the whole network is often inefficient or infeasible. Averaging *consensus* algorithms [14] provide a fully distributed way to iteratively solve this task. While every node exchanges messages only with its directly connected neighbors, it can be shown that, under mild conditions, these algorithms converge to the correct global average ([24]).

Consensus algorithms form the basis for a large number of distributed algorithms (such as Distributed Hypothesis Testing [13], Distributed Maximum Likelihood Estimation [2, 19], and Distributed Kalman Filtering [11, 18]), and have been studied under a wide variety of conditions (such as networks with undirected or directed links, time varying topology [7] or noisy channels [20]).

Nevertheless, standard consensus is not robust to packet losses, common in low-power wireless networks. The problem is that traditional solutions require symmetric packet exchanges, i.e., if node j receives a packet from node i, then it is assumed that node i will receive a packet from node j. If this does not happen (due to packet losses), consensus algorithms will misbehave and not converge to the correct average. This adverse effect can be lessened (but not completely eliminated) through reliable transmission schemes which reduce the *effective* link loss. Doing so, however, significantly increases the communication and execution overhead of the algorithm, while as we show in Section 3 the estimated average can still be far from the correct value.

In Section 4 we introduce *corrective consensus*, a mechanism that eliminates the consensus error even in presence of asymmetric link losses. The core idea is to maintain at each node i additional variables ϕ_{ij} which represent the amount of change the node has made to its local state due to the updates from its neighbor j. By periodically exchanging these new variables between neighbors during *corrective iterations*, the nodes have a chance to update their local states and correct the errors due to packet losses. We prove that this scheme obtains almost sure convergence to the correct average by selecting an appropriate number of retransmissions in the standard and corrective iterations.

In Section 5 we use simulations to compare the convergence properties of corrective and standard consensus algorithms. The results show that in a network with 10 nodes and 80% packet loss, standard consensus may fail to converge to the correct average even with 50 retransmissions. On the other hand, corrective consensus reaches the same order of error at a speed that is more than 10 times faster. Continuing the execution of the corrective consensus reduces the error to zero.

2 Related Work

Consensus has been widely studied in a variety of contexts, including those we summarize in Section 3. In the paragraphs that follow we compare our work to the consensus variants that are most closely related to the one we consider.

Rajagopal and Wainwright investigated consensus averaging on graphs whose links are symmetric and polluted by quantization noise [15]. Specifically, they studied consensus algorithms based on damped updates and proved convergence to a Gaussian distribution whose variance depends on the Laplacian's eigenvalues. We consider networks that suffer from a more severe form of faults, where links can arbitrarily discard packets.

Kar and Moura studied average consensus with random topologies and noisy channels [7]. They proposed two algorithms: A-NC, which averages multiple runs of consensus with a fixed number of iterations, and A-ND, which modifies conventional consensus by forcing the weights to satisfy a persistence condition (slowly decaying to zero). When the channels are without noise, only A-ND assures almost-sure convergence to the correct average as our algorithm does. However, their assumption is that packet losses are symmetric and the modification on the weights increases the number of iterations needed to reach convergence.

Asymmetric packet losses have also been modeled with directed switching network topologies, for instance by Kingston and Beard [8] (which use standard consensus) and Li and Zhang [9] (which use an approach similar to A-ND). In all the cases, however, there is the critical assumption that the network is *balanced* (a precise definition will be given later).

Mehyar et al. derived asynchronous averaging algorithms in packet-switched networks [10]. They implemented and tested their algorithms on Planetlab, a global-scale distributed network overlaid on the public Internet. Their formulation also uses variables ϕ_{ij} to correct errors. Nevertheless, their algorithms assume a reliable bidirectional exchange of packets between any pair of nodes. Furthermore, Internet-connected PC class nodes do not have the resource constraints that wireless sensor nodes face.

In summary, the main problem with existing solutions is that they assume symmetric packet losses or balanced graphs, which is not a realistic assumption in wireless networks. Our work, on the other hand, guarantees almost-sure convergence without this restrictive assumption.

Control systems over wireless networks that drop packets have drawn much attention [6, 21]. Instead of achieving statistical optimality, our work aims to completely eliminate the error caused by packet drops in average consensus algorithms and enable their practical use in wireless networks.

3 Background

Consensus is a distributed iterative algorithm designed for networks of connected devices, such as networks of wireless sensor nodes deployed to measure physical quantities of interest (e.g., temperature). Each node i starts with its own measurement z_i and the consensus algorithm aims to compute $\overline{z} = \frac{1}{N}\sum_{i=1}^{N} z_i$, where N is the number of nodes in the network. In other words, consensus calculates the average of the initial measurements across the whole network. Rather than collecting all the z_i's at a central node to compute the average, each node i maintains a running local estimate of $\overline{z}$, denoted as $x_i(t)$ with t being the iteration counter. $x_i(t)$ is also known as the state value of node i. During each consensus iteration, nodes exchange their state values with their neighbors and each node updates its own state based on a weighted average of the state values it receives. It can then be shown that under certain conditions all $x_i(t)$ eventually converge to $\overline{z}$ [12, 14, 17]

We model the network as a directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Vertex set $\mathcal{V} = \{1, 2, \ldots, N\}$ represents the nodes. An ordered pair $(i, j) \in \mathcal{E}$ if node j can transmit packets containing its state value to node i directly. Reflecting the realities of wireless communication, we assume that link (i, j) may experience random packet losses. For ease of analysis, we further assume the packet losses are independent, and denote the probability of loss as $1 - p_{ij} \in [0, 1]$ for link (i, j). Said differently, p_{ij} is the Packet Reception Ratio (PRR) of the wireless link $j \to i$. We assume p_{ij} is time-invariant and $p_{ij} = p_{ji}$ for any (i, j) pair, i.e., the PRRs in both directions $(j \to i)$ and $(i \to j)$ are equal. Denote $N_i = \{j : p_{ij} > 0\}$ as the set of neighboring nodes of node i.

Nodes can leverage the broadcast nature of the wireless medium to broadcast their state values to their neighbors in every iteration of the consensus algorithm. Then, the value of p_{ij} determines the probability of receiving the broadcast packet on the $j \to i$ link. Nodes can perform multiple broadcasts in one iteration to increase the probability of delivering their updates. Then if a node transmits n broadcasts, the probability of successfully delivering at least one copy of the update on the $j \to i$ link is $\hat{p}_{ij} = 1 - (1 - p_{ij})^n$. Here the $\hat{p}_{ij}$ is the effective packet reception ratio on the $j \rightarrow i$ link.

One can define the network adjacency matrix $A(t)$ of graph $\mathcal{G}$, whereby $A_{ij}(t) = 1$ if node j successfully transmitted its state value to node i during the t-th iteration and is zero otherwise. Furthermore, by definition $A_{ii} = 0$ for all i's. The network adjacency matrix $A(t)$ is a random 0-1 matrix whose distribution does not depend on t. Also, we denote $d_i(t) = \sum_{j=1}^{N} A_{ij}(t)$ as the in-degree of node i in iteration t. Then, the degree matrix $D(t)$ is a diagonal matrix with $D_{ii}(t) = d_i(t)$. Finally, the Laplacian matrix of the graph $\mathcal{G}$ is

$$L(t) = D(t) - A(t) \tag{1}$$

The eigenvalues of $L(t)$ are within a disk centered at $\max_{i,t}(d_i(t)) + 0j$ on the complex plane with a radius of $\max_{i,t}(d_i(t))$, due to Gershgorin's theorem [5].

3.1 Standard Consensus

In the standard Laplacian-based consensus each node i updates its state x_i as

$$x_i(t) = \begin{cases} \sum_{j=1}^{N} W_{ij}(t-1)x_j(t-1) & t > 0 \\ z_i & t = 0 \end{cases} \tag{2}$$

Written in vector form, the whole network updates its state values vector as

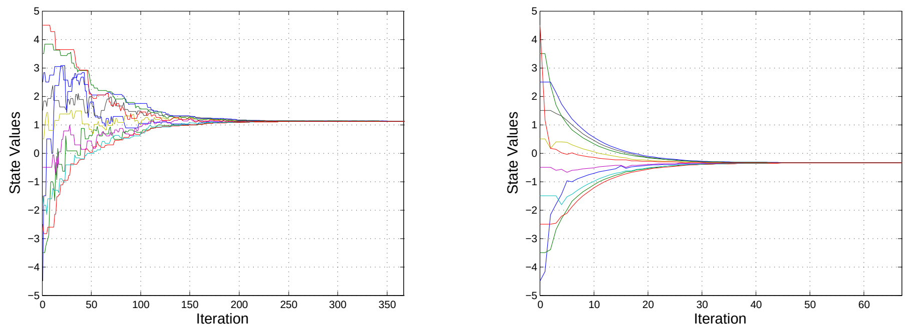
$$x(t) = \begin{cases} W(t-1)x(t-1) & t > 0 \\ z & t = 0 \end{cases} \tag{3}$$

where $x(t), z \in \mathbf{R}^{N\times 1}$ are column vectors and $W(t-1) \in \mathbf{R}^{N\times N}$ is the weight matrix used during the t-th iteration. The weight matrix $W(t)$ is defined as

$$W(t) = I - \epsilon L(t) \tag{4}$$

where I is the identity matrix of the same dimension as W and ϵ is a small positive constant. One important property of $W(t)$ is that $W(t)\mathbf{1} = \mathbf{1}$ for any t, where $\mathbf{1}$ is the column vector of all ones. If one selects $\epsilon < 1/\max_{i,t}(d_i(t))$, the eigenvalues of $W(t)$ will be distributed as $1 = \lambda_1 \geq |\lambda_2| \geq \cdots \geq |\lambda_N| \geq 0$.

Let $\mathbf{E}W$ be the expectation of $W(t)$ which does not depend on t. Note that when link qualities are symmetric (i.e., $p_{ij} = p_{ji}, \forall(i,j) \in \mathcal{E}$) $\mathbf{E}W$ is also symmetric and thus from $\mathbf{E}W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T\mathbf{E}W = \mathbf{1}^T$ also holds. It can be shown that if $\mathcal{G}$ is connected and $\epsilon < 1/\max_{i,t}(d_i(t))$, then $|\lambda_2(\mathbf{E}W)| < 1$ ([14]).



(a) Standard consensus without any retransmissions.

(b) Each node performs up to 20 retransmissions when necessary.

Figure 1: Running consensus on a 10-node ring topology where all links have PRR of 20% and the sum of all initial state values is equal to zero. In both cases, consensus converges to biased estimates.

This is an important property because it can also be shown that the update scheme shown in Euqation (3) reaches consensus, i.e., $\lim_{t\to\infty} x(t) = \alpha\mathbf{1}$, if $|\lambda_2(\mathbf{E}W)| < 1$ ([22]).

While $\alpha = \overline{z}$ when the weight matrix $W(t)$ is symmetric (as is the case for undirected graphs [25]), $W(t)$ is a random matrix and is not always *balanced*, i.e., $\exists\ t$ s.t. $\mathbf{1}^T W(t) \neq \mathbf{1}^T$. In this case, consensus will converge to a *biased* value that is not equal to the average of the nodes' initial states [3].

In the section that follows we show how unbalanced weight matrices, caused by asymmetric link losses, lead to biased estimates and quantify the magnitude of the resulting estimation error.

3.2 Effect of Packet Loss

As previously argued, Equation (3) does not converge to the average of initial measurements because $W(t)$ is not symmetric due to random packet loss. To see how this asymmetry leads to bias consider the case during the t-th iteration when node i receives the update from node j but the packet from i to j is lost. In other words, $A_{ij} = 1$ but $A_{ji} = 0$. Therefore, node i will update its state as $x_i(t+1) \leftarrow x_i(t) + \epsilon(x_j(t) - x_i(t))$[1] but node j will not

[1]We omit other neighbors to illustrate the effect of packet loss. A formal treatment will be given in Section 4.

change its state value. As a result, $x_i(t+1) + x_j(t+1) \neq x_i(t) + x_j(t)$.

On the other hand, if both packets are received $x_i(t+1) + x_j(t+1) = x_i(t) + \epsilon(x_j(t) - x_i(t)) + x_j(t) + \epsilon(x_i(t) - x_j(t)) = x_i(t) + x_j(t)$. Likewise, if both the packets are lost, the sum of states is also preserved because no changes are made. Keeping the sum of states equal across iterations ensures that α equals to the average of nodes' initial states. It is then clear that the error in the converged value α comes from the iterations in which node i and j take different actions, i.e., one updates whereas the other does not.

Acknowledgments and retransmissions can effectively tame packet losses and they should intuitively be able to help nodes take the same action. In the previous example where $A_{ij} = 1$ but $A_{ji} = 0$, if node i knows that the packet it sent to j was lost, then it can simply disregard j's packet to preserve the sum of the states. Or more actively, node i can try resending its packet and use acknowledgments to ensure its reception at node j.

Therefore, it seems plausible that consensus will converge to the exact average of the initial measurements if nodes employ an automatic repeat request (ARQ) scheme at each iteration of Equation (3). However, it is impossible for ARQ acknowledgments and retransmissions to eliminate these biases. This is due to the well-known Two Generals' Problem which proves it is impossible for two armies to coordinate their attack using messengers sent through a valley occupied by their common enemy [1, 4]. In short, node i and node j can not be 100% sure that they will take the same action, regardless of how many messages they are willing to exchange. If for once node i updates whereas node j does not, an error will be introduced and consensus will generally not converge to the actual average of initial states.

However, sufficient retransmissions can significantly reduce the probability of nodes taking different actions. Nevertheless, our simulation results show that the difference between α and $\overline{z}$ can still be quite large even with a high number of message exchanges. Figure 1 presents the results of running consensus on a 10-node ring topology in which the PRR of all links is 20%. It is evident from (a) that without retransmissions, errors can easily occur. The converged value is 1.12, whereas the average value of the initial states is 0. An ARQ scheme is employed in (b) with each node configured to perform up to 20 retransmissions if necessary. Therefore the effective PRR in this case is approximately 99% on every link. Nevertheless, the converged value has still significant bias. The converged value in this case is -0.34, while the average of the initial states is 0.

The results above indicate that retransmissions can indeed reduce biases in the converged value. However, real-life wireless networks have long links with low PRR ([26]) and thus require a large number of retransmissions to ameliorate the effects of packet losses. Doing so however increases the time necessary to complete each consensus iteration. For example, each iteration in Figure 1(b) takes at least 20 times more time than one iteration in Figure 1(a). Therefore, the total convergence time is significantly prolonged in (b), despite the fewer number of iterations. The speed of convergence is critical for some types of applications, such as controlling unmanned vehicles and camera surveillance network. Section 5 elaborates on the speed of convergence.

4 Corrective Consensus

Notice that Equation (2) can be written as

$$x_i(t+1) = x_i(t) + \sum_{j=1}^{N} W_{ij}(t)\big(x_j(t) - x_i(t)\big), \quad x_i(0) = z_i, \tag{5}$$

therefore we can define a new set of variables $\phi_{ij}(t)$ as

$$\phi_{ij}(t+1) = \phi_{ij}(t) + W_{ij}(t)\big(x_j(t) - x_i(t)\big), \quad \phi_{ij}(0) = 0. \tag{6}$$

It can be seen that on each node i, the auxiliary variable ϕ_{ij} represents the amount of change that node i has made to its state variable $x_i(t)$ due to neighbor j. Also, note that keeping ϕ_{ij} and updating it according to Equation (6) do not need any additional message exchange because the nodes already execute Equation (5) at every iteration.

If node i and j always take the same action, as explained in Section 3.2, then the changes they make should be symmetric, i.e., $\phi_{ij} = -\phi_{ji}$. Therefore, it is natural to define a new set of variables

$$\Delta_{ij}(t) = \phi_{ij}(t) + \phi_{ji}(t), \tag{7}$$

and $\Delta_{ij}(t)$ would represent the amount of bias (in the sum of the states) that has accumulated on both directions of the (i, j) link. Note that from Equation (5) and (6) we have the interesting property that

$$x_i(t) = x_i(0) + \sum_{j \in N_i} \phi_{ij}(t), \tag{8}$$

from which it follows that

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Delta_{ij}(t) = \mathbf{1}^T x(t) - \mathbf{1}^T x(0). \tag{9}$$

Therefore, one straightforward approach for eliminating the consensus error is to first run consensus according to Equation (5) until $x = \alpha\mathbf{1}$ and then remove the bias by

$$\alpha \leftarrow \alpha - \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Delta_{ij}. \tag{10}$$

Unfortunately, Equation (10) requires each node to know $\frac{1}{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\Delta_{ij}$, which is a consensus problem by itself. Note that $\phi_{ij}(t)$ is locally stored at node i, and calculating $\Delta_{ij}(t)$ requires nodes i and j to exchange $\phi_{ij}(t)$ and $\phi_{ji}(t)$.

Instead of removing the bias in one step, which is unrealistic, nodes should reduce the error in a distributed and iterative manner: each node i corrects its own state value. Specifically, node i collects ϕ_{ji} ($j \in N_i$) from its neighbors to calculate the Δ_{ij}'s. Then node i adjusts its state variable $x_i(t)$ using the Δ_{ij}'s, thereby accounting for the errors accumulated on its direct (1-hop) links. After this correction, nodes resume the standard consensus shown in (5) while periodically performing the corrective step described above.

In summary, there are two types of iterations in corrective consensus: *Standard* and *Corrective* iterations.

(1) During a *standard iteration*, nodes exchange state values in a best-effort manner and update the values in accordance with Equation (5). Each node performs up to n transmissions to deliver its state variables (i.e., the node performs up to $n-1$ retransmissions). In addition, each node also updates the ϕ_{ij}'s according to Equation (6).

(2) During a *corrective iteration*, nodes exchange ϕ_{ij}'s to calculate the Δ_{ij}'s and use them to adjust their state variables x_i's and auxiliary variables ϕ_{ij}'s. Each node attempts up to m transmissions to deliver the ϕ_{ij}'s.

Corrective consensus starts with the standard iterations and after every k consecutive standard iterations, one corrective iteration takes place as follows

$$x_i(k+1) = x_i(k) - \sum_{j=1}^{N} \Delta_{ij}(k)/2 \tag{11}$$

$$\phi_{ij}(k+1) = \phi_{ij}(k) - \Delta_{ij}(k)/2 \tag{12}$$

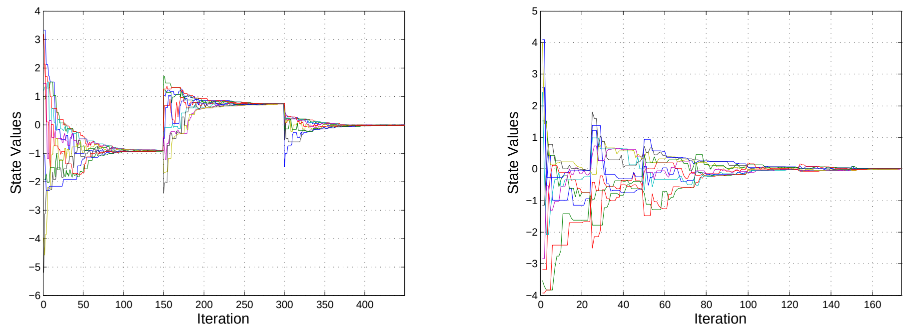
Overall, corrective consensus iterates as follows

$$x_i(t+1) = \begin{cases} x_i(t) - \dfrac{1}{2}\displaystyle\sum_{j=1}^{N} \Delta_{ij}(t) & \text{if } t = u(k+1)-1 \\ x_i(t) + \displaystyle\sum_{j=1}^{N} W_{ij}(t)\big(x_j(t) - x_i(t)\big) & \text{otherwise} \end{cases}$$
$$\phi_{ij}(t+1) = \begin{cases} \phi_{ij}(t) - \dfrac{1}{2}\Delta_{ij}(t) & \text{if } t = u(k+1)-1 \\ \phi_{ij}(t) + W_{ij}(t)(x_j(t) - x_i(t)) & \text{otherwise} \end{cases} \tag{13}$$

where $u = 1, 2, 3, \cdots$ and represents the u-th corrective iteration. Note that there are k standard iterations executed in between every two corrective iterations, therefore the u-th corrective iteration corresponds to the $u(k+1)$-th iteration in total. As a result it is straightforward to define a shorthand $\hat{u} = u(k+1)$, which translates the index of corrective iterations into the global index of all iterations, and will be used later on in Section 4.3.

Notice that similar to the x_i's, the ϕ_{ij}'s can be delivered asymmetrically (despite the m transmissions) due to packet drops. Nevertheless, for ease of exposition we assume for now that the ϕ_{ij}'s are reliably delivered and remove this assumption in Section 4.3. It follows from this assumption that $\Delta_{ij}(u(k+1)) = 0$, $\forall i, j, u$.

Figure 2 shows two examples of the corrective consensus algorithm, in a 10-node ring topology for which all links have PRR of 20%. The average of the initial state values is zero. 2(a) uses $k = 149$ while 2(b) uses $k = 24$. In 2(a) the states converge to a biased value during the k standard iterations and the biases are then eliminated during each corrective iteration. After two corrective iterations, the state values converge to the correct average of the initial states. In 2(b) k is not large enough for the states to converge within each k standard iterations. Therefore, the state value curves appear to be much smoother. It is also evident from Figure 2 that selecting an overly large k can lead to unnecessary iterations, which waste resources and delay convergence.



(a) $k = 149$. (b) $k = 24$.

Figure 2: Two examples of corrective consensus. (a) When k is large, state values almost converge before the corrective iterations. After two corrective iterations, state values reach the average of the initial measurements. (b) When k is small variations across state values are sizable when corrective iterations take place. In both cases corrective consensus converges to the correct average value.

4.1 Basic Properties

Let us consider the first corrective iteration

$$
\begin{aligned}
x(k+1) &= x(k) - \frac{1}{2}\begin{bmatrix} \sum_{j=1}^{N} \Delta_{1j}(k) \\ \vdots \\ \sum_{j=1}^{N} \Delta_{Nj}(k) \end{bmatrix} \\
&= x(0) + \begin{bmatrix} \sum_{j=1}^{N} \phi_{1j}(k) \\ \vdots \\ \sum_{j=1}^{N} \phi_{Nj}(k) \end{bmatrix} - \frac{1}{2}\begin{bmatrix} \sum_{j=1}^{N} \phi_{1j}(k) + \phi_{j1}(k) \\ \vdots \\ \sum_{j=1}^{N} \phi_{Nj}(k) + \phi_{jN}(k) \end{bmatrix} \\
&= x(0) + \frac{1}{2}\begin{bmatrix} \sum_{j=1}^{N} \phi_{1j}(k) - \phi_{j1}(k) \\ \vdots \\ \sum_{j=1}^{N} \phi_{Nj}(k) - \phi_{jN}(k) \end{bmatrix} \\
&= x(0) + \frac{1}{2}\sum_{j=1}^{N}\sum_{s=0}^{k-1}\begin{bmatrix} (W_{1j}(s) + W_{j1}(s))(x_j(s) - x_1(s)) \\ \vdots \\ (W_{Nj}(s) + W_{jN}(s))(x_j(s) - x_N(s)) \end{bmatrix}
\end{aligned}
$$

$$= x(0) + \frac{1}{2}\sum_{s=0}^{k-1}\Big[W(s) + W^T(s) - \mathbf{diag}\Big(\big(W(s) + W^T(s)\big)\mathbf{1}\Big)\Big]M(s)x(0) \tag{14}$$

where we define

$$M(s) = \begin{cases} \prod_{t=0}^{s-1} W(t) & s \geq 1 \\ I & s = 0 \end{cases} \tag{15}$$

and note that $M(s)x(0) = x(s)$.

Let us define

$$P(u) = I + \frac{1}{2}\sum_{s=u(k+1)}^{(u+1)(k+1)-2}\Big[W(s) + W^T(s) - \mathbf{diag}\Big(\big(W(s) + W^T(s)\big)\mathbf{1}\Big)\Big]M(s) \tag{16}$$

and it follows from $\Delta_{ij}(u(k+1)) = 0$ that

$$x\big((k+1)(u+1)\big) = P(u)x\big((k+1)u\big) \tag{17}$$

Theorem 1. *$P(u)$ has the following properties*

1. $P(u)\mathbf{1} = \mathbf{1}$.
2. $\mathbf{1}^T P(u) = \mathbf{1}^T$.
3. $\mathbf{E}P(u) = (\mathbf{E}W)^k$.

Proof. 1) and 2) are easily verifiable. To show 3), note that $W(s)$ and $M(s)$ are independent, $\mathbf{E}W(s) = \mathbf{E}W^T(s) = \mathbf{E}W$ and $\mathbf{E}W\mathbf{1} = \mathbf{1}$. □

As a direct result, we have

$$\mathbf{E}x\big((k+1)u\big) = (\mathbf{E}W)^{ku}x(0) \tag{18}$$

which indicates that the expectation of the state values after each corrective iteration will converge to $\mathbf{1}\overline{z}$ because $\mathbf{E}W$ is symmetric and $|\lambda_2(\mathbf{E}W)| < 1$ (cf. §3.1).

4.2 Convergence Analysis

We start with the observation that if Equation (13) converges to a single point, then the converged point has to satisfy two conditions: $x = \overline{z}\mathbf{1}$ and $\Delta_{ij} = 0, \forall i, j$. As a result, in what follows we only need to show that Equation (13) converges, which is equivalent to proving $x(t) - \mathbf{1}\mathbf{1}^T x(t)/N \to \mathbf{0}$.

Define

$$\tilde{x}(t) = x(t) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t) \tag{19}$$

and our goal is to show that the L2-norm $\|\tilde{x}(t)\| \to 0$.

Theorem 2. *After k consecutive standard iterations, we have*

$$\mathbf{E}\big\|\tilde{x}\big((k+1)u-1\big)\big\| \le \overline{\lambda_2}^k \mathbf{E}\big\|\tilde{x}\big((k+1)(u-1)\big)\big\|$$

where $\overline{\lambda_2} = \mathbf{E}\big(|\lambda_2(W(t))|\big)$.

Proof. $\forall t \neq (k+1)u-1$, $u = 1, 2, \cdots$, we have

$$x(t+1) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t) = W(t)\big(x(t) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t)\big) \tag{20}$$

because $W(t)\mathbf{1} = \mathbf{1}$, $\forall t \ge 0$. In addition, we have

$$\begin{aligned} \|\tilde{x}(t+1)\| &= \|x(t+1) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t+1)\| \\ &\le \|x(t+1) - \frac{1}{N}\mathbf{1}\mathbf{1}^T x(t)\| \le |\lambda_2(W(t))|\|\tilde{x}(t)\| \end{aligned} \tag{21}$$

Take the expectation and note that $W(t)$ and $\tilde{x}(t)$ are independent, we have

$$\mathbf{E}\|\tilde{x}(t+1)\| \le \mathbf{E}\big(|\lambda_2(W(t))|\big)\mathbf{E}\|\tilde{x}(t)\| = \overline{\lambda_2}\mathbf{E}\|\tilde{x}(t)\| \tag{22}$$

For k standard iterations, we have

$$\mathbf{E}\big\|\tilde{x}\big((k+1)u-1\big)\big\| \le \overline{\lambda_2}^k \mathbf{E}\big\|\tilde{x}\big((k+1)(u-1)\big)\big\| \tag{23}$$

□

Theorem 2 shows that $\|\tilde{x}(t)\|$ decreases during the standard iterations. However, it is easy to verify that $\|\tilde{x}(t)\|$ might increase in a corrective iteration. Therefore, we will focus on the state values immediately after each corrective iteration. Denote $y(u) = x\big((k+1)u\big)$, then we are interested in the sequence of $\|\tilde{y}(u)\|$. Here we also denote $\tilde{y}(u) = y(u) - \frac{1}{N}\mathbf{1}\mathbf{1}^T y(u)$.

Theorem 3.

$$\mathbf{E}\|\tilde{y}(u)\| \leq \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N}\frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}} \right)^u \|\tilde{y}(0)\|$$

where $p = \underset{r\in\{\hat{p}_{ij}\}}{\arg\min}|r-0.5|$*, and we denote* $\tilde{p} = 2(p-p^2)$ *as the shorthand. Note that* $0 \leq \tilde{p} \leq 0.5$*.*

Proof. First, we need to define two sets of variables

$$\varphi_{ij}(t) = W_{ij}(t)\big(x_j(t) - x_i(t)\big) = W_{ij}(t)\big(\tilde{x}_j(t) - \tilde{x}_i(t)\big) \tag{24}$$

$$\delta_{ij}(t) = \varphi_{ij}(t) + \varphi_{ji}(t) \tag{25}$$

and note that $\mathbf{E}\delta_{ij}(t) = 0$, $\forall t, i, j$, which is due to the assumption that $p_{ij} = p_{ji}$. Next we can write the correction iterations as follows

$$x(k+1) = x(k) - \frac{1}{2}\sum_{s=0}^{k-1}\begin{bmatrix} \sum_{i=1}^N \delta_{i1}(s) \\ \vdots \\ \sum_{i=1}^N \delta_{iN}(s) \end{bmatrix} \tag{26}$$

and left multiply by $(I - \mathbf{1}\mathbf{1}^T/N)$, we will arrive at

$$\tilde{x}(k+1) = \tilde{x}(k) - \frac{1}{2}\sum_{s=0}^{k-1}\begin{bmatrix} \displaystyle\sum_{i=1}^N \delta_{i1}(s) - \frac{1}{N}\sum_{\substack{0\leq j\leq N\\ 0\leq i\leq N}} \delta_{ij}(s) \\ \vdots \\ \displaystyle\sum_{i=1}^N \delta_{iN}(s) - \frac{1}{N}\sum_{\substack{0\leq j\leq N\\ 0\leq i\leq N}} \delta_{ij}(s) \end{bmatrix} \tag{27}$$

Take the L2-norm on Equation (27), we get

$$\begin{aligned}
\|\tilde{x}(k+1)\| &\leq \|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\left\| \begin{matrix} \sum_{i=1}^N \delta_{i1}(s) - \frac{1}{N}\sum_{j=1}^N\sum_{i=1}^N \delta_{ij}(s) \\ \vdots \\ \sum_{i=1}^N \delta_{iN}(s) - \frac{1}{N}\sum_{j=1}^N\sum_{i=1}^N \delta_{ij}(s) \end{matrix} \right\| \\
&= \|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\left\{ \sum_{j=1}^N \Big[\sum_{i=1}^N \delta_{ij}(s) - \frac{1}{N}\sum_{j=1}^N\sum_{i=1}^N \delta_{ij}(s)\Big]^2 \right\}^{\frac{1}{2}}
\end{aligned}$$

$$
\begin{aligned}
&= \|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\sqrt{\sum_{j=1}^{N}\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2 - \frac{1}{N}\Big(\sum_{j=1}^{N}\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2} \\
&\le \|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\sqrt{\sum_{j=1}^{N}\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2}
\end{aligned} \tag{28}
$$

Take the expectation on both sides of Equation (28), we have

$$
\mathbf{E}\|\tilde{x}(k+1)\| \le \mathbf{E}\|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\mathbf{E}\sqrt{\sum_{j=1}^{N}\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2} \tag{29}
$$

Now, we would like to give an upper bound on the second term of the right-hand side of Equation (29). First, we will compute the upper bound on $\mathbf{E}\big(\sum_{i=1}^{N}\delta_{ij}(s)\big)^2$. Let us start with the conditional expectation which we write as follows

$$
\begin{aligned}
&\mathbf{E}\Big[\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2\Big|\tilde{x}(s)\Big] = \mathbf{E}\Big[\sum_{i=1}^{N}\delta_{ij}^2(s) + 2\sum_{i=1}^{N-1}\sum_{l=i+1}^{N}\delta_{ij}(s)\delta_{lj}(s)\Big|\tilde{x}(s)\Big] \\
&= \mathbf{E}\Big[\sum_{i=1}^{N}\delta_{ij}^2(s)\Big|\tilde{x}(s)\Big] = \sum_{i=1}^{N}\mathbf{E}\Big[\delta_{ij}^2(s)\Big|\tilde{x}(s)\Big] \\
&= \sum_{i=1}^{N}\mathbf{E}\left[(W_{ij}(s) - W_{ji}(s))^2\right](\tilde{x}_j(s) - \tilde{x}_i(s))^2 \\
&= \sum_{i=1}^{N}2(\hat{p}_{ij} - \hat{p}_{ij}^2)\epsilon^2(\tilde{x}_j(s) - \tilde{x}_i(s))^2 \\
&\le \sum_{i=1}^{N}2(p - p^2)\epsilon^2(\tilde{x}_j(s) - \tilde{x}_i(s))^2 \\
&= 2(p - p^2)\epsilon^2\Big(\sum_{i=1}^{N}\tilde{x}_j^2(s) + \sum_{i=1}^{N}\tilde{x}_i^2(s) - 2\sum_{i=1}^{N}\tilde{x}_j(s)\tilde{x}_i(s)\Big) \\
&= 2(p - p^2)\epsilon^2\Big(N\tilde{x}_j^2(s) + \|\tilde{x}(s)\|^2\Big) = \tilde{p}\epsilon^2\Big(N\tilde{x}_j^2(s) + \|\tilde{x}(s)\|^2\Big)
\end{aligned} \tag{30}
$$

Recall that $\hat{p}_{ij} = 1 - (1 - p_{ij})^n$, where n is the number of transmissions in each standard iteration. In Equation (30) we define $p = \arg\min_{r \in \{\hat{p}_{ij}\}} |r - 0.5|$, and we use a shorthand $\tilde{p} = 2(p - p^2)$. Note that $0 \leq \tilde{p} \leq 0.5$ and $\tilde{p}$ reaches the maximum when $p = 0.5$.

With the results from Equation (30), we can now write the conditional expectation for the sum of the N terms

$$\begin{aligned} \mathbf{E}\Big[\sum_{j=1}^{N}\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2\Big|\tilde{x}(s)\Big] &= \sum_{j=1}^{N}\mathbf{E}\Big[\Big(\sum_{i=1}^{N}\delta_{ij}(s)\Big)^2\Big|\tilde{x}(s)\Big] \\ &\leq \sum_{j=1}^{N}\epsilon^2\tilde{p}\left(N\tilde{x}_j^2(s) + \|\tilde{x}(s)\|^2\right) \\ &= 2N\epsilon^2\tilde{p}\|\tilde{x}(s)\|^2 \end{aligned} \tag{31}$$

and finally the upper bound for the second term on the right-hand side of Equation (29) can be given as follows

$$\begin{aligned} \mathbf{E}\sqrt{\sum_{j=1}^{N}\big(\sum_{i=1}^{N}\delta_{ij}(s)\big)^2} &= \mathbf{E}\Big[\mathbf{E}\Big[\sqrt{\sum_{j=1}^{N}\big(\sum_{i=1}^{N}\delta_{ij}(s)\big)^2}\Big|\tilde{x}(s)\Big]\Big] \\ &\leq \mathbf{E}\Big[\sqrt{\mathbf{E}\Big[\sum_{j=1}^{N}\big(\sum_{i=1}^{N}\delta_{ij}(s)\big)^2\Big|\tilde{x}(s)\Big]}\Big] \\ &\leq \mathbf{E}\Big[\sqrt{2\tilde{p}N\epsilon^2\|\tilde{x}(s)\|^2}\Big] = \epsilon\sqrt{2\tilde{p}N}\mathbf{E}\|\tilde{x}(s)\| \end{aligned} \tag{32}$$

where the second line follows from Jensen's inequality.

Now, plug Equation (32) into Equation (29) and we will get

$$\begin{aligned} \mathbf{E}\|\tilde{x}(k+1)\| &\leq \mathbf{E}\|\tilde{x}(k)\| + \frac{1}{2}\sum_{s=0}^{k-1}\epsilon\sqrt{2\tilde{p}N}\mathbf{E}\|\tilde{x}(s)\| \\ &= \mathbf{E}\|\tilde{x}(k)\| + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N}\sum_{s=0}^{k-1}\mathbf{E}\|\tilde{x}(s)\| \\ &\leq \overline{\lambda_2}^k\|\tilde{x}(0)\| + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N}\sum_{s=0}^{k-1}\overline{\lambda_2}^s\|\tilde{x}(0)\| \end{aligned}$$

$$
\begin{aligned}
&= \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N} \sum_{s=0}^{k-1} \overline{\lambda_2}^s \right) \|\tilde{x}(0)\| \\
&= \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N} \frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}} \right) \|\tilde{x}(0)\|
\end{aligned} \tag{33}
$$

Assuming ϕ_{ij}'s can be reliably exchanged, we have $\Delta_{ij}((k+1)u) = 0$, for $u = 1, 2, 3, \cdots$. Therefore, we can extend Equation (33) to

$$
\mathbf{E}\|\tilde{x}((k+1)u)\| \leq \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N} \frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}} \right)^u \|\tilde{x}(0)\| \tag{34}
$$

i.e.,

$$
\mathbf{E}\|\tilde{y}(u)\| \leq \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N} \frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}} \right)^u \|\tilde{y}(0)\| \tag{35}
$$

□

Define $c = \left(\overline{\lambda_2}^k + \frac{1}{2}\epsilon\sqrt{2\tilde{p}N} \frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}} \right)$, and it can be seen from Theorem 3 that c is the critical value that determines the speed of convergence.

Theorem 4. $\|\tilde{y}(u)\|$ *almost surely converges to zero if* $c < 1$.

Proof. By Markov's inequality, we have $\forall \delta > 0$

$$
\begin{aligned}
&\lim_{t\to\infty} \sum_{i=t}^{\infty} P(\|\tilde{y}(i)\| > \delta) \leq \lim_{t\to\infty} \sum_{i=t}^{\infty} \frac{\mathbf{E}\|\tilde{y}(i)\|}{\delta} \\
&\leq \lim_{t\to\infty} \sum_{i=t}^{\infty} \frac{\|\tilde{y}(0)\| c^i}{\delta} = \frac{\|\tilde{y}(0)\|}{\delta(1-c)} \lim_{t\to\infty} c^t = 0
\end{aligned}
$$

□

Now, the question is whether c will be smaller than 1. Notice that the value of c is collectively controlled by ϵ, $\overline{\lambda_2}$, $\tilde{p}$ and k. Here $\tilde{p}$ depends on the $\hat{p}_{ij}$'s only. $\overline{\lambda_2}$ depends on ϵ, the expected topology $\mathbf{E}W$ and the $\hat{p}_{ij}$'s.

Theorem 5. *Critical value c can always be made less than one by employing retransmissions during standard iterations.*

Proof. First, $\overline{\lambda_2} < 1$ if $\mathcal{G}$ is connected. As a result, $\overline{\lambda_2}^k$ goes to 0 as k increases, whereas $\frac{1-\overline{\lambda_2}^k}{1-\overline{\lambda_2}}$ in the second term is upper bounded by $\frac{1}{1-\overline{\lambda_2}}$. Second, $\tilde{p}$ can be made arbitrarily close to 0 by increasing n, which is the number of transmissions in each standard iteration. □

4.3 Unreliable Exchange of ϕ_{ij}'s

In this section we consider the case that ϕ_{ij}'s might be lost, i.e., $\Delta_{ij}\big((k+1)u\big)$ may be nonzero for $u = 1, 2, 3, \cdots$. As a consequence, the nonzero Δ_{ij} values will propagate to the standard iterations that follow. Therefore, the results obtained from Theorem 3 are not directly applicable.

First, we need to define a set of variables indicating the reception status of ϕ_{ij}:

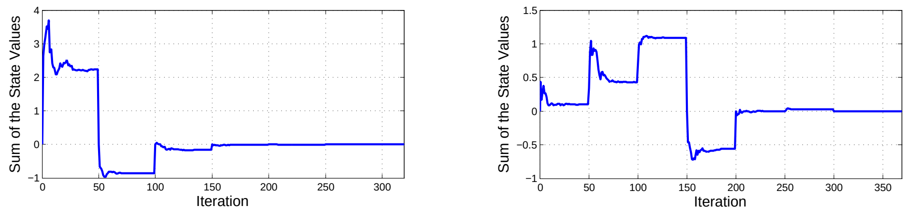
$$v_{ij}(u) = \begin{cases} 1 & \text{if node } i \text{ receives } \phi_{ji} \text{ from node } j \\ & \text{at the } u\text{-th corrective iteration} \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

We note here that $v_{ij}(u)$ is a random variable whose distribution depends on p_{ij} and the number of transmissions. For example, $v_{ij}(u)$ is always 1 if $p_{ij} = 1$, because node i receives every packet from j. We denote $q_{ij} = P(v_{ij}(u) = 1)$, $\forall u$. $q_{ij} = 1 - (1 - p_{ij})^m$, where m is the maximum number of transmissions allowed in each corrective iteration.

Next, we can rewrite the corrective consensus scheme as

$$\begin{aligned} x_i(t+1) &= \begin{cases} x_i(t) - \dfrac{1}{2}\displaystyle\sum_{j=1}^{N} \Delta_{ij}(t) v_{ij}(u) & t = u(k+1) - 1 \\ \sum_{j=1}^{N} W_{ij}(t) x_j(t) & \text{otherwise} \end{cases} \\ \phi_{ij}(t+1) &= \begin{cases} \phi_{ij}(t) - \dfrac{1}{2}\Delta_{ij}(t) v_{ij}(u) & t = u(k+1) - 1 \\ \phi_{ij}(t) + W_{ij}(t)(x_j(t) - x_i(t)) & \text{otherwise} \end{cases} \end{aligned} \tag{37}$$

where $u = 1, 2, 3, \cdots$ and represents the u-th corrective iteration.



(a) ϕ_{ij}'s are reliably delivered in corrective iterations.

(b) ϕ_{ij}'s can be lost in corrective iterations.

Figure 3: Sum of the state values when running corrective consensus in a 10-node ring topology in which all links have PRR of 80%. $k = 49$ **in both cases. (a) The sum of the state values is zero during each corrective iteration. When the** ϕ_{ij}**'s are not reliably exchanged in (b) the sum is nonzero even during the corrective iterations. However, both cases eventually converge to the correct value.**

Figure 3 plots the sum of the state values when running corrective consensus, with the sum of the initial measurements being zero. (a) assumes the ϕ_{ij}'s can always be delivered and therefore the sum is always zero during the corrective iterations. On the other hand, ϕ_{ij}'s can be lost in (b) and as a result each corrective iteration is not always able to reduce all the bias. In both cases the sums eventually converge to the correct value.

Theorem 6. *Define* $q = \min\{q_{ij} : q_{ij} > 0\}$. *If there exists* b *such that* $c \le b < 1$ *and* $\sqrt{(1-q)(1-q/2)} \le \frac{b(b-c)}{2b-c}$, *then we have* $\mathbf{E}\|\tilde{x}((k+1)u)\| \le cb^{u-1}\|\tilde{x}(0)\|$ *for* $u \ge 1$.

Proof. Let us start with the first corrective iteration, which can be written as

$$\phi_{ij}(k+1) = \phi_{ij}(k) - \frac{1}{2}v_{ij}(1)\big(\phi_{ij}(k) + \phi_{ji}(k)\big) \tag{38}$$

From Equation (38), we can see that if $v_{ij}(1) = 0$, it follows that $\phi_{ij}(k+1) = \phi_{ij}(k)$, i.e., node i does not change ϕ_{ij} because it does not receive ϕ_{ji} and as a result can not compute Δ_{ij}. On the other hand, node i will correct the ϕ_{ij} (and its state value) when $v_{ij}(1) = 1$, as is the case in Section 4.2. Along the same line, we can extend Equation (38) to each corrective iteration as

$$
\begin{aligned}
\Delta_{ij}(\hat{u}) &= \begin{cases} 0 & \text{if } v_{ij}(u) = v_{ji}(u) = 1 \\ \frac{1}{2}\Delta_{ij}(\hat{u}-1) & \text{if } v_{ij}(u) + v_{ji}(u) = 1 \\ \Delta_{ij}(\hat{u}-1) & \text{if } v_{ij}(u) = v_{ji}(u) = 0 \end{cases} \\
&= \Big(1 - \frac{1}{2}\big(v_{ij}(u) + v_{ji}(u)\big)\Big)\Delta_{ij}(\hat{u}-1)
\end{aligned} \tag{39}
$$

where we use $\hat{u} = (k+1)u$ as a shorthand.

Equation (39) gives the dynamics of the nonzero sums of $\phi_{ij}(\hat{u})$ propagating to the subsequent iterations. Therefore, we can rewrite Equation (39) as follows

$$
\Delta_{ij}(\hat{u}) = \sum_{r=1}^{u} \sum_{s=(k+1)(r-1)}^{(k+1)r-2} \delta_{ij}(s)\Lambda_{ij}(u,r) \tag{40}
$$

for which we define a new set of variables

$$
\Lambda_{ij}(u,r) = \prod_{l=r}^{u}\Big(1 - \frac{1}{2}\big(v_{ij}(l) + v_{ji}(l)\big)\Big). \tag{41}
$$

Note that in Equation (40) $\delta_{ij}(s)$ represents the amount of bias accumulated on link (i,j) at the s-th iteration. Therefore, it can be seen that $\Lambda_{ij}(u,r)$ is the decay coefficient (considered at the u-th corrective iteration) for biases accumulated between the $(r-1)$-th and the r-th corrective iterations

With the notations from Equation (40), we can rewrite Equation (26) as

$$
\begin{aligned}
x(\hat{u}) &= x(\hat{u}-1) - \frac{1}{2}\begin{bmatrix} \sum_{i=1}^{N}\Delta_{i1}(\hat{u}-1)v_{i1}(u) \\ \vdots \\ \sum_{i=1}^{N}\Delta_{iN}(\hat{u}-1)v_{iN}(u) \end{bmatrix} \\
&= x(\hat{u}-1) - \frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\begin{bmatrix} \sum_{i=1}^{N}\delta_{i1}(s)v_{i1}(u) \\ \vdots \\ \sum_{i=1}^{N}\delta_{iN}(s)v_{iN}(u) \end{bmatrix} \\
&\qquad - \frac{1}{2}\begin{bmatrix} \sum_{i=1}^{N}\Delta_{i1}(\hat{u}-k-1)v_{i1}(u) \\ \vdots \\ \sum_{i=1}^{N}\Delta_{iN}(\hat{u}-k-1)v_{iN}(u) \end{bmatrix}
\end{aligned}
$$

$$
= x(\hat{u}-1) - \frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\begin{bmatrix}\sum_{i=1}^{N}\delta_{i1}(s)v_{i1}(u)\\ \vdots \\ \sum_{i=1}^{N}\delta_{iN}(s)v_{iN}(u)\end{bmatrix}
- \frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\begin{bmatrix}\sum_{i=1}^{N}\delta_{i1}(s)v_{i1}(u)\Lambda_{i1}(u-1,r)\\ \vdots \\ \sum_{i=1}^{N}\delta_{iN}(s)v_{iN}(u)\Lambda_{iN}(u-1,r)\end{bmatrix} \quad (42)
$$

and left multiply by $(I - \mathbf{1}\mathbf{1}^T/N)$, we have

$$
\tilde{x}(\hat{u}) = \tilde{x}(\hat{u}-1) - \frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}(I-\mathbf{1}\mathbf{1}^T/N)\begin{bmatrix}\sum_{i=1}^{N}\delta_{i1}(s)v_{i1}(u)\\ \vdots \\ \sum_{i=1}^{N}\delta_{iN}(s)v_{iN}(u)\end{bmatrix}
- \frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}(I-\mathbf{1}\mathbf{1}^T/N)\cdot\begin{bmatrix}\sum_{i=1}^{N}\delta_{i1}(s)v_{i1}(u)\Lambda_{i1}(u-1,r)\\ \vdots \\ \sum_{i=1}^{N}\delta_{iN}(s)v_{iN}(u)\Lambda_{iN}(u-1,r)\end{bmatrix} \quad (43)
$$

Take the L2-norm on both sides of Equation (43), we have

$$
\|\tilde{x}(\hat{u})\| \le \|\tilde{x}(\hat{u}-1)\| + \frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u) - \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Big]^2\Bigg\}^{\frac{1}{2}}
$$

$$
\begin{aligned}
&+\frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\\
&\quad-\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2\Bigg\}^{\frac{1}{2}}\\
\le{}&\|\tilde{x}(\hat{u}-1)\|+\frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Big]^2\Bigg\}^{\frac{1}{2}}\\
&+\frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2\Bigg\}^{\frac{1}{2}}
\end{aligned}
\tag{44}
$$

Notice that $v_{ij}^2(u) \le 1$, $\forall u$, and that $\mathbf{E}\delta_{ij}(t) = 0$, $\forall t, i, j$, we have

$$
\mathbf{E}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Big]^2 \le \mathbf{E}\Big[\sum_{i=1}^{N}\delta_{ij}(s)\Big]^2 \tag{45}
$$

Combining this inequality and the results from Equation (33), we can take the expectation of the first two terms on the right-hand side of Equation (44) and get the following

$$
\begin{aligned}
&\mathbf{E}\|\tilde{x}(\hat{u}-1)\|+\mathbf{E}\left[\frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Big]^2\Bigg\}^{\frac{1}{2}}\right]\\
&\le \mathbf{E}\|\tilde{x}(\hat{u}-1)\|+\mathbf{E}\left[\frac{1}{2}\sum_{s=\hat{u}-k-1}^{\hat{u}-1}\Bigg\{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)\Big]^2\Bigg\}^{\frac{1}{2}}\right]\\
&\le c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\|
\end{aligned}
\tag{46}
$$

Now we can plug Equation (46) back into Equation (44) and will get

$$
\begin{aligned}
\mathbf{E}\|\tilde{x}(\hat{u})\| \le{}& c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\|\\
&+\frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\mathbf{E}\sqrt{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2}
\end{aligned}
\tag{47}
$$

Note that $v_{ij}(u)$, $\Lambda_{ij}(u-1,r)$ and $\delta_{ij}(s)$ are independent random variables for all i, j, u and for $r < u$ and $s \le (k+1)r-2$. Therefore, it can be shown that $\mathbf{E}(v_{ij}^2(u)) = q_{ij}$, and $\mathbf{E}(\Lambda_{ij}^2(u-1,r)) = [(1-q_{ij})(1-q_{ij}/2)]^{u-r}$. With these results we can do the calculation

$$\begin{aligned}
&\mathbf{E}\sqrt{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2} \\
&= \mathbf{E}\left[\mathbf{E}\Big[\sqrt{\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2}\Big|\tilde{x}(s)\Big]\right] \\
&\leq \mathbf{E}\left[\sqrt{\mathbf{E}\Big[\sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\delta_{ij}(s)v_{ij}(u)\Lambda_{ij}(u-1,r)\Big]^2\Big|\tilde{x}(s)\Big]}\right] \\
&= \mathbf{E}\left[\sqrt{\mathbf{E}\Big[\sum_{j=1}^{N}\sum_{i=1}^{N}\delta_{ij}^2(s)v_{ij}^2(u)\Lambda_{ij}^2(u-1,r)\Big|\tilde{x}(s)\Big]}\right] \\
&= \mathbf{E}\left[\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{N}\mathbf{E}\left[\delta_{ij}^2(s)\Big|\tilde{x}(s)\right]q_{ij}\big((1-q_{ij})(1-q_{ij}/2)\big)^{u-r}}\right] \\
&\leq \sqrt{\hat{q}}\big((1-\hat{q})(1-\hat{q}/2)\big)^{\frac{u-r}{2}}\mathbf{E}\left[\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{N}\mathbf{E}\left[\delta_{ij}^2(s)\Big|\tilde{x}(s)\right]}\right] \\
&\leq \sqrt{\hat{q}}\big((1-\hat{q})(1-\hat{q}/2)\big)^{\frac{u-r}{2}}\sqrt{2\tilde{p}N}\epsilon\mathbf{E}\|\tilde{x}(s)\| \\
&\leq \big((1-\hat{q})(1-\hat{q}/2)\big)^{\frac{u-r}{2}}\sqrt{2\tilde{p}N}\epsilon\mathbf{E}\|\tilde{x}(s)\| \\
&\leq \big((1-q)(1-q/2)\big)^{\frac{u-r}{2}}\sqrt{2\tilde{p}N}\epsilon\mathbf{E}\|\tilde{x}(s)\|
\end{aligned} \tag{48}$$

where $\hat{q} = \underset{s\in\{q_{ij}\}}{\arg\max}\ s\big((1-s)(1-s/2)\big)^{u-r}$ and $q = \underset{s\in\{q_{ij}\},s>0}{\arg\min}\ s$. We denote $\tilde{q} = \sqrt{(1-q)(1-q/2)}$ as a shorthand and plug Equation (48) into Equation (47), which gives us the following equation

$$\begin{aligned}
\mathbf{E}\|\tilde{x}(\hat{u})\| &\leq c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\| + \frac{1}{2}\sum_{r=1}^{u-1}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\tilde{q}^{u-r}\sqrt{2\tilde{p}N}\epsilon\mathbf{E}\|\tilde{x}(s)\| \\
&= c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\| + \sum_{r=1}^{u-1}\tilde{q}^{u-r}\sum_{s=(k+1)(r-1)}^{(k+1)r-2}\frac{1}{2}\sqrt{2\tilde{p}N}\epsilon\mathbf{E}\|\tilde{x}(s)\|
\end{aligned}$$

$$
\begin{aligned}
&\leq c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\| + \sum_{r=1}^{u-1} \tilde{q}^{u-r} \frac{1}{2}\sqrt{2\tilde{p}N\epsilon} \sum_{s=0}^{k-1} \overline{\lambda_2}^s \mathbf{E}\|\tilde{x}((k+1)(r-1))\| \\
&= c\mathbf{E}\|\tilde{x}(\hat{u}-k-1)\| + \sum_{r=1}^{u-1} \tilde{q}^{u-r}(c-\overline{\lambda_2}^k)\mathbf{E}\|\tilde{x}((k+1)(r-1))\|
\end{aligned}
\tag{49}
$$

We observe that there is a recursion pattern in the last line of Equation (49), which motivates us to go with mathematical induction to derive a upper bound for $\mathbf{E}\|\tilde{x}(\hat{u})\|$. Recall that $\hat{u} = (k+1)u$, in what follows we will prove the statement that $\mathbf{E}\|\tilde{x}((k+1)u)\| \leq cb^{u-1}\|\tilde{x}(0)\|$ holds for any $u \geq 1$.

First, we need to show that the statement holds for the base case (i.e., when $u = 1$). In this case, $\hat{u} = k+1$, and it follows from the results in Equation (46) that $\mathbf{E}\|\tilde{x}(k+1)\| \leq c\|\tilde{x}(0)\| = cb^{u-1}\|\tilde{x}(0)\|$. Note that $\Delta_{ij}(0) \equiv 0$, and therefore it is very similar to the case where ϕ_{ij}'s are never lost.

Next we need to show the inductive step that if $\mathbf{E}\|\tilde{x}((k+1)u)\| \leq cb^{u-1}\|\tilde{x}(0)\|$ holds for $1 \leq u \leq U-1$, where U is some integer, then it can be extended to the case where $u = U$, i.e., $\mathbf{E}\|\tilde{x}((k+1)U)\| \leq cb^{U-1}\|\tilde{x}(0)\|$.

With the assumption that $\mathbf{E}\|\tilde{x}((k+1)u)\| \leq cb^{u-1}\|\tilde{x}(0)\|$ holds for $1 \leq u \leq U-1$, we can rewrite Equation (49) by substituting u with U

$$
\begin{aligned}
&\mathbf{E}\|\tilde{x}(\hat{U})\| \leq c\mathbf{E}\|\tilde{x}(\hat{U}-k-1)\| + \sum_{r=1}^{U-1} \tilde{q}^{U-r}(c-\overline{\lambda_2}^k)\mathbf{E}\|\tilde{x}((k+1)(r-1))\| \\
&= c\mathbf{E}\|\tilde{x}(\hat{U}-k-1)\| + (c-\overline{\lambda_2}^k)\tilde{q}^{U-1}\|\tilde{x}(0)\| \\
&\qquad\qquad + (c-\overline{\lambda_2}^k)\sum_{r=2}^{U-1} \tilde{q}^{U-r}\mathbf{E}\|\tilde{x}((k+1)(r-1))\| \\
&\leq c^2 b^{U-2}\|\tilde{x}(0)\| + (c-\overline{\lambda_2}^k)\tilde{q}^{U-1}\|\tilde{x}(0)\| + (c-\overline{\lambda_2}^k)\sum_{r=2}^{U-1} \tilde{q}^{U-r} c b^{r-2}\|\tilde{x}(0)\| \\
&= \|\tilde{x}(0)\|\Big\{c^2 b^{U-2} + (c-\overline{\lambda_2}^k)\tilde{q}^{U-1} + (c-\overline{\lambda_2}^k)\sum_{r=2}^{U-1} \tilde{q}^{U-r} c b^{r-2}\Big\} \\
&< \|\tilde{x}(0)\|\Big\{c^2 b^{U-2} + c\tilde{q}^{U-1} + c\sum_{r=2}^{U-1} \tilde{q}^{U-r} c b^{r-2}\Big\}
\end{aligned}
$$

$$
\begin{aligned}
&= cb^{U-1}\|\tilde{x}(0)\|\Big\{cb^{-1} + \left(\frac{\tilde{q}}{b}\right)^{U-1} + cb^{-1}\sum_{r=2}^{U-1}\left(\frac{\tilde{q}}{b}\right)^{U-r}\Big\} \\
&\le cb^{U-1}\|\tilde{x}(0)\|\Big\{cb^{-1} + \sum_{r=1}^{U-1}\left(\frac{\tilde{q}}{b}\right)^{U-r}\Big\} \\
&< cb^{U-1}\|\tilde{x}(0)\|\Big\{cb^{-1} + \frac{\tilde{q}/b}{1-\tilde{q}/b}\Big\}
\end{aligned} \tag{50}
$$

Observe from the last line of Equation (50) that it suffices to show that $\Big\{cb^{-1} + \frac{\tilde{q}/b}{1-\tilde{q}/b}\Big\}$ is not greater than 1, which is equivalent to the following

$$
\begin{aligned}
& cb^{-1} + \frac{\tilde{q}/b}{1-\tilde{q}/b} \le 1 \Leftrightarrow cb^{-1} + \frac{\tilde{q}}{b-\tilde{q}} \le 1 \\
& \Leftrightarrow (2b-c)\tilde{q} \le b(b-c) \Leftrightarrow \tilde{q} \le \frac{b(b-c)}{2b-c} \\
& \Leftrightarrow \sqrt{(1-q)(1-q/2)} \le \frac{b(b-c)}{2b-c}
\end{aligned} \tag{51}
$$

As a result, as long as we tune the number of transmissions (i.e., m) in the corrective iterations to control the value of q such that the inequality in Equation (51) is satisfied, we will have $\mathbf{E}\tilde{x}((k+1)U) \le cb^{U-1}$.

By mathematical induction, we have proved that $\mathbf{E}\|\tilde{x}((k+1)u)\| \le cb^{u-1}\|\tilde{x}(0)\|$, $\forall u \ge 1$, under the condition that there exists b such that $c \le b < 1$ and $\sqrt{(1-q)(1-q/2)} \le \frac{b(b-c)}{2b-c}$.

□

Theorem 7. *If* $\mathbf{E}\|\tilde{x}((k+1)u)\| \le cb^{u-1}\|\tilde{x}(0)\|$ *with* $c \le b < 1$*, then* $\|\tilde{x}((k+1)u)\|$ *almost surely converges to zero.*

Proof. Proof is similar to the one used in Theorem 4. □

Theorem 8. *By tuning the number of retransmissions used during the corrective iterations, it is always possible to satisfy* $\sqrt{(1-q)(1-q/2)} \le \frac{b(b-c)}{2b-c}$.

Proof. Define $p = \underset{r\in\{p_{ij}\},0<r<1}{\arg\min}\ r$, then $q = 1-(1-p)^m$. By increasing m, one can make $\sqrt{(1-q)(1-q/2)}$ arbitrarily close to 0. In other words, $\forall b$ such that $c < b < 1$, $\exists\ m$ such that $\sqrt{(1-q)(1-q/2)} \le \frac{b(b-c)}{2b-c}$. □

Notice that when $q = 1$ (i.e., the ϕ_{ij}'s are never dropped), we can choose b such that $b = c$, and the results derived in this section naturally degenerate to the ones presented in Section 4.2.

We note that while the convergence conditions derived in this section are sufficient it is not clear whether they are also necessary. Nevertheless, the simulation results from the next section suggest that the corrective consensus algorithms always converge to the correct value when $\mathcal{G}$ is connected. Furthermore, the corrective consensus algorithms in the next section are all configured with $n = 1$, meaning that the nodes simply broadcast their state values once in each standard iteration.

5 Evaluation

In what follows, we first define *convergence* and then evaluate the performance of the proposed corrective consensus algorithm using convergence error and speed as the evaluation metrics.

For the standard consensus, we declare that the algorithm converges during the t-th iteration if the non-increasing quantity $\|\tilde{x}(t)\|$ is less than a threshold κ_1. For the corrective consensus, however, this quantity could increase during a corrective iteration. Therefore, we add another condition: $|\frac{1}{N}\mathbf{1}^T x(t) - \overline{z}| < \kappa_2$, where κ_2 is another predefined threshold. Then, corrective consensus reaches convergence at the t-th iteration if both conditions are satisfied.

5.1 Convergence Error

Let the $\hat{t}$-th iteration be the first iteration that the convergence condition(s) are met. Then, our first metric is the convergence error, defined as $e = |\frac{1}{N}\mathbf{1}^T x(\hat{t}) - \overline{z}|$.

Figure 4 compares the convergence errors for different consensus algorithms that ran in a 10-node ring topology with PRR=20% for every link. The initial states are within $[-10, 10]$ and $\overline{z} = 0$, $\kappa_1 = \kappa_2 = 0.001$. As described in Section 3.2, increasing the number of transmissions n, reduces the error of the standard consensus algorithm. When $n = 50$ the standard consensus has comparable error values to the corrective consensus, with the average error being in the order of 10^{-5} for both algorithms. We note that unlike standard consensus which does not guarantee convergence to $\overline{z}$, the

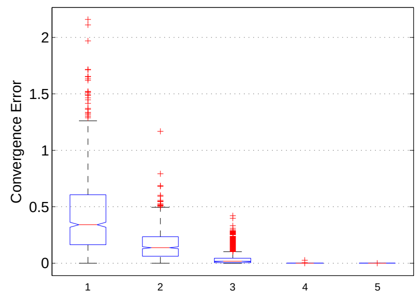


Figure 4: Box plots of the convergence errors for standard consensus and corrective consensus. Columns 1-4 are standard consensus with $n = 1, 10, 20, 50$ transmissions per iteration, respectively; column 5 is corrective consensus with $n = 1$, $m = 10$ and $k = 25$.

convergence error of corrective consensus will reach zero, if not stopped by the convergence conditions.

5.2 Convergence Speed

Convergence speed can be measured in terms of number of iterations. Generally, fewer iterations translate to shorter time. Nevertheless, as mentioned in Section 3.2, the duration of each iteration varies based on the number of retransmissions performed during each iteration.

In standard consensus, a node only needs to transmit its state value to its neighbors. Due to the nature of wireless communication, the most efficient approach is to broadcast the state value as a packet, allowing all neighbors a chance to receive the update. However, when nodes start to employ the retransmission scheme, broadcasting leads to the ack implosion problem (i.e., the neighbors' acknowledgments can collide at the origin). Not requiring neighbors to send acknowledgments removes this problem. In that case however a node has to blindly broadcast n times in every iteration, regardless of whether the neighbors have received the update. The problem becomes worse when n is large, which is likely the case in real life wireless networks due to the existence of long links with low PRR [26]. Therefore, we decide that when retransmissions are enabled, a node unicasts its state value to each of its neighbors. In turn, every neighbor replies with an acknowledgement

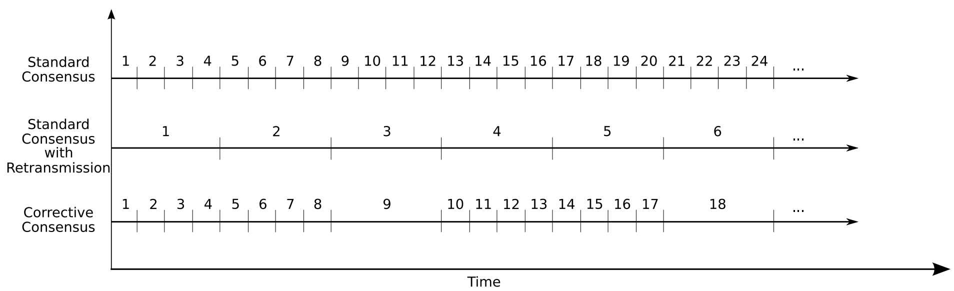


Figure 5: Illustration of the varying duration for different types of iterations. The corrective consensus is configured to broadcast once in each standard iteration, and $k = 8$.

packet upon the reception of the state value. The node attempts up to n times to deliver its state value to a neighbor. Also due to the broadcast nature of wireless communication, the other neighbors can overhear the unicast packet and take it as the state value. Therefore, each node actually has in average more than n opportunities of receiving the packet.

Without loss of generality, we assume that the duration of one iteration is $\tau = n\alpha\beta$ for $n > 1$. Here n is the number of transmissions per iteration, α reflects the network density (i.e., the average number of neighbors) and β represents the transmission latency.

In the corrective iteration of the corrective consensus, each node needs to send the ϕ_{ij}'s to its neighbors. Because generally ϕ_{ij}'s are not equal, the node needs to send different packets to the neighbors, and it will do so by unicast. Therefore, the time of one iteration is $\tau = m\alpha\beta$ in this case, with m being the number of allowed transmissions to deliver each ϕ_{ij}. In practice, the node could potentially put all the ϕ_{ij} in one packet and broadcast, if network density is low. Nevertheless, we do not consider this optimization in the above analysis as it belongs to the implementation details.

Figure 5 presents an example of the duration for different types of iterations. The standard consensus can perform significantly more iterations within the same amount of time. When retransmissions are enabled, each iteration becomes significantly longer. Consequently, standard consensus with retransmissions completes a fraction of iterations during the same amount of time. Corrective consensus amortizes the overhead related to retransmissions by performing them once every k standard iterations. Note that in each

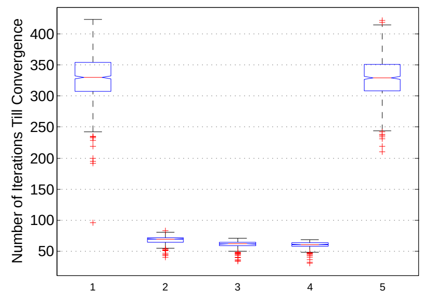


Figure 6: Number of iterations until convergence. Columns 1-5 are the same as Figure 4.

of the k standard iterations, the corrective consensus simply broadcasts the state value once.

We note that the ratio between iterations with and without retransmissions can in real life by far exceed the 4:1 ratio shown in Figure 5, depending on the implementation and network conditions.

Figure 6 presents the number of iterations necessary for the different consensus algorithms to converge, while Figure 7 shows the actual amount of time necessary for the algorithms to converge. The network configurations and parameters are identical to those in Figure 4. The results from Figure 6 are somewhat misleading, suggesting that standard consensus with retransmissions requires fewer iterations to converge. However, it is evident from Fig.7 that the actual convergence time for standard consensus with retransmissions is significantly longer than the time for corrective consensus, confirming the relative algorithm execution times from Fig.5.

Combining the results from Figure 4, we know that the standard consensus requires $n = 50$ transmissions to achieve the same error level with corrective consensus. Therefore, one can see from Figure 7 that achieving the same error level by retransmissions increases the execution time by more than tenfold. Last but not least, by comparing columns 1 and 5 in Figure 7, one can see that the overhead introduced by the corrective iterations is marginal compared to that of using even modest number of retransmissions.

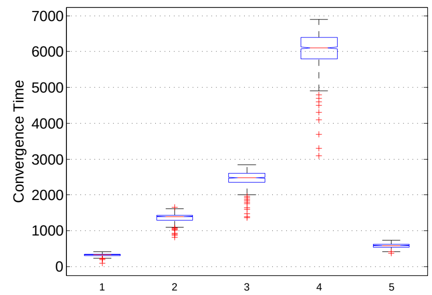


Figure 7: Convergence time. x-axis is proportional to the actual time. Columns 1-5 are the same as Figure 4.

	$n = 1$	$n = 10$	$n = 20$	$n = 50$	$n = 100$	corrective
error	0.1153	0.0407	0.0386	0.0222	0.0208	4.6e-4

Table 1: Convergence error in a 10-node random topology in which long links with low PRR exist.

5.3 Random Topology

Table 1 lists the performance of various consensus algorithms in a topology formed by randomly placing 10 nodes within a rectangular area. The PRR between each pair of nodes is determined by the Log-normal path loss model [16] with parameters experimentally derived from an environmental monitoring sensor network deployed in a forest, and the PRR-SNR curve of the 802.15.4 compliant CC2420 radio [23]. In this topology there exist long links that have low PRR. Specifically, several links have PRR below 10%, and the lowest is 3.4%. As a consequence, doing retransmissions alone is hard to eliminate the convergence error. Nevertheless, corrective consensus is able to reduce the error to a very low level. Once again we note that the error is due to the use of the κ_1 and κ_2 thresholds and the error eventually reduces to zero as the algorithm continues to execute.

6 Conclusion

Consensus algorithms constitute a valuable theoretical tool for computing scalar averages across networks of interconnected devices. Unfortunately, existing solutions are impractical when applied to wireless networks that naturally exhibit asymmetric packet losses [26].

In this paper we present a novel *corrective consensus* algorithm that enables the practical use of consensus in real-life sensor networks. Through the addition of new variables at each node and new corrective iterations, we prove that the proposed method converges almost surely to the correct average despite random and asymmetric link losses. Furthermore, we compare the performance of corrective and standard consensus algorithms both in terms of accuracy of the results and convergence speed.

Selecting the optimal interval k for corrective iterations is part of our future work. It is also interesting to investigate the behavior of corrective consensus when the $p_{ij} = p_{ji}$ assumption is removed.

References

[1] E. A. Akkoyunlu, K. Ekanadham, and R. V. Huber. Some constraints and tradeoffs in the design of network communications. In *SOSP '75: Proceedings of the fifth ACM symposium on Operating systems principles*, pages 67–74, New York, NY, USA, 1975. ACM.

[2] S. Barbarossa and G. Scutari. Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems. *Signal Processing, IEEE Transactions on*, 55(7):3456–3470, 2007.

[3] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM J. Control Optim.*, 48(1):102–133, 2009.

[4] J. Gray. Notes on data base operating systems. In *Operating Systems, An Advanced Course*, pages 393–481, London, UK, 1978. Springer-Verlag.

[5] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge Univ. Press, 1987.

[6] Z. Jin, V. Gupta, and R. M. Murray. State estimation over packet dropping networks using multiple description coding. *Automatica*, 42(9):1441–1452, 2006.

[7] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. *Trans. Sig. Proc.*, 57(1):355–369, 2009.

[8] D. Kingston and R. Beard. Discrete-time average-consensus under switching network topologies. In *American Control Conference*, page 6 pp., june 2006.

[9] T. Li and J.-F. Zhang. Consensus conditions of multi-agent systems with time-varying topologies and stochastic communication noises. *Automatic Control, IEEE Transactions on*, 55(9), 2010.

[10] M. Mehyar, D. Spanos, J. Pongsajapan, S. Low, and R. Murray. Asynchronous distributed averaging on communication networks. *IEEE/ACM Transactions on Networking*, 15(3):512–520, 2007.

[11] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5492–5498, Dec. 2007.

[12] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan. 2007.

[13] R. Olfati-saber, E. Franco, E. Frazzoli, and J. S. Shamma. Belief consensus and distributed hypothesis testing in sensor networks. In *Network Embedded Sensing and Control. (Proceedings of NESC'05 Worskhop), volume 331 of Lecture Notes in Control and Information Sciences*, pages 169–182. Springer Verlag, 2006.

[14] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, Sept. 2004.

[15] R. Rajagopal and M. J. Wainwright. Network-based consensus averaging with general noisy channels. Technical report, Department of Statistics, University of California, Berkeley, May 2008.

[16] T. S. Rappaport. *Wireless Communications: Principles & Practices.* Prentice Hall, 1996.

[17] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. In *IEEE Control Systems Magazine*, 2007.

[18] I. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro. Consensus in ad hoc wsns with noisy links—part ii: Distributed estimation and smoothing of random signals. *Signal Processing, IEEE Transactions on*, 56(4):1650–1666, April 2008.

[19] I. Schizas, A. Ribeiro, and G. Giannakis. Consensus in ad hoc wsns with noisy links—part i: Distributed estimation of deterministic signals. *Signal Processing, IEEE Transactions on*, 56(1):350–364, Jan. 2008.

[20] U. Schmid and C. Fetzer. Randomized asynchronous consensus with imperfect communications. *Reliable Distributed Systems, IEEE Symposium on*, 0:361, 2003.

[21] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49:1453–1464, 2004.

[22] A. Tahbaz-Salehi and A. Jadbabaie. On consensus over random networks. In *44th Annu. Allerton Conf. Commun., Contr. Comput.*, pages 1315–1321, 2006.

[23] Texas Instruments. CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Available at Available at `http://www.ti.com/lit/gpn/cc2420`, 2006.

[24] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 5, pages 4997–5002 Vol.5, Dec. 2003.

[25] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 9, Piscataway, NJ, USA, 2005. IEEE Press.

[26] J. Zhao and R. Govindan. Understanding Packet Delivery Performance In Dense Wireless Sensor Networks. In *Proceedings of the ACM Sensys*, Nov. 2003.