DATA TRANSMISSION IN QUANTIZED CONSENSUS

Imtiaz Parvez

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

May 2013

APPROVED:

Yan Wan, Major Professor
Shengli Fu, Co-Major Professor
Kamesh Namuduri, Committee Member
Murali Varanasi, Chair of the Department of Electrical Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Mark Wardell, Dean of the Toulouse Graduate School

Parvez, Imtiaz. *Data Transmission in Quantized Consensus.* Master of Science (Electrical Engineering), May 2013, 48 pp., 4 tables, 18 illustrations, bibliography, 34 titles.

In the world of networked system, average consensus is an important dimension of co-ordinate control and cooperation. Since the communication medium is digital, real value cannot be transmitted and we need to perform quantization before data transmission. But for the quantization, error is introduced in exact value and initial average is lost. Based on this limitation, my 16 bit quantization method (sending MSB in 1-4 cycle and MSB+LSB in 5th cycle) reduces error significantly and preserves initial average. Besides, it works on all types of graphs (star, complete, ring, random geometric graph). My other algorithm, distributing averaging algorithm (PQDA) with probabilistic quantization also works on random geometric graph, star, ring and slow co-herency graph. It shows significant reduced error and attain strict consensus.

ACKNOWLEDGMENTS

At first, I would like to thank God. I would like to express my sincere gratitude to my advisor, Dr.Yan Wan for her guidance and support. As an advisor, she always motivated me with her immense knowledge and made me enthusiastic to make my research successful. She is a great teacher. Her classes are always fun and the way she presents a topic is outstanding and her exams are always challenging and motivating. I am also grateful to my co-advisor Dr.Shengli Fu. He not only advised me in my educational life but also guided me to set goals in my professional life.

Besides my advisor, I would like to thank Dr.Kamesh Namuduri for his support as committee member of my thesis. I am so grateful for the support of my parents, brother, sister and my friends. Generally, I also appreciate all the people I know for helping me grow up.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The concepts of consensus and its formal study first originated in management science and statistics in 1960. Afterwards, the ideas of statistical consensus theory by Degroot [21, 22] reappeared two decades later in aggregation of information with uncertainty obtained from multiple sensors and medical experts. In computer science, it has been used for a long time and started new field distributed computing [22].

In a dynamic system or network, consensus means to reach common quantity of interest. A consensus algorithm is an interaction rule that specifies information exchange among agents.

Recently, extensive research on multi agent network systems have been going on with close ties to consensus. This research[26] includes consensus, collective behavior of flocks and swarms, sensor fusion, random networks, synchronization of coupled oscillators, algebraic connectivity of complex networks, asynchronous distributed algorithms, formation control for multirobot systems, optimization-based cooperative control, dynamic graphs, complexity of coordinated tasks and consensus based belief propagation in Bayesian networks etc.

In literature, different kinds of consensus algorithms are found, and each algorithm is appropriate for a specific purpose. Some algorithm is chosen based on time invariant/variant, memory/memoryless, noisy/deterministic system, strict/weak consensus, uniform quantization/dither quantization, single communication/multiple communication system etc. The main pivotal thing in reaching consensus is speed (i.e., how fast they reach consensus) and precision of consensus (i.e., how much near the node values at final stage). But power and bandwidth constraints limit communication between agents and restrict the limit of consensus speed. Besides that for digital communication, error is introduced in the value due to quantization and channel noise (white gussacian noise). The quantization noise is a function of quantization levels or steps. The smaller the quantization level, the less error is introduced during quantization. The power and bandwidth constraints also allow a specific number of quantization levels. So there is a tradeoff between both convergence speed and strictness of consensus, and quantization level. This motivates me to work on to find an appropriate quantization level in accordance with the power/bandwidth constraint that reduces error from the desired value and attains specific speed.

Moreover, most of the algorithms were implemented on a random geometric graph. An algorithm working on a specific graph may not provide desired results on other types of graphs. For this, I have implemented a specific algorithm to star graph, ring and slow co-herency graph and investigated their results.

My proposed gossip algorithm is based on the random gossip algorithm [13, 6, 18]. Besides that, I investigated distributed averaging with the dither algorithm [5, 32]. The significant contributions of my work are

1) The proposed algorithm saves a significant amount of energy.

2) Since in the proposed algorithm, data use 50% bandwidth in 80% time of data transmission, it saves bandwidth utilization.

3) In the proposed algorithm, the mean square error is reduced significantly.

4) The proposed algorithm works in major types of graph- complete, random geometric, star and ring graph.

5) The implementation of distributed averaging algorithm with probabilistic quantization is extended to other types of graphs- star graph, ring graph and slow coherency graph.

1.1. Overview of Thesis Content

In this thesis, I illustrate theoretical background, modified algorithm, experimental results and theoretical proof on different types of networks. The thesis is organized as follows. In chapter 2, I broadly introduce the background and current status of consensus. In chapter 3, I represent my new proposed algorithm- Average quantized consensus building by gossip algorithm using 16 bit quantization and efficient data transfer. In chapter 4, I show the result of implementation of distributed averaging with dithered quantization. Finally, in chapter 5, I conclude and recommend future work.

CHAPTER 2

BACKGROUND AND REVIEW OF THE AVERAGE CONSENSUS

This chapter provides a review on average consensus up to the cutting edge research.

2.1. Motivation

Networked systems have become a matter of interest and caught the attention of mathematicians, computer scientists and electrical engineers. The arrival of networked systems are not only a consequence of technological advances, but also it need to perform some complex tasks like rescue, health care and environmental monitoring and surveillance.[7] In the networked system, co-ordinate control and co-operation are the governing issue, and consensus is one of the most important dimensions of this co-ordinate control and co-operation.

Consensus means to reach agreement on some key information or common interest that enable them to cooperate in a co-ordinated fashion. The application of consensus has been extended in various sectors.

In multivehicle (unmanned vehicle) co-operative control, co-ordination is required for numerous vehicles. Besides that, for monitoring in wild fire, oil field, pipeline, hospital, tracking wildlife, patrolling the border, surveillance, reconnaissance etc, co-operative networking is the most reliable and efficient way. In this case, application of consensus algorithm based on information formation, rendezvous, attitude alignment, flocking etc need to be implemented.

Load balancing is a critical issue in parallel and distributed computation for efficient utilization of computation resources. The goal is to develop decentralized consensus based on algorithm to distribute work load among processors to speed up the calculation and efficient utilization of resources.

In the application [7] of wireless sensors like mobile target tracking, event detection, efficient TDMA[1] scheduling, sleep scheduling with very low duty cycle etc, global clock syn-

[1]Time Division Multiple Access

chronization is the most critical issue. In these cases, the nodes act in synchronized and coordinated fashion. In clock synchronization, consensus algorithm plays an important role in determining a common notion of time.

In wireless sensor networking, distributed estimation and tracking signal are the most important information colaborating processes. Recently the application of consensus has attracted the attention in this estimation calculation. The solution of the best linear unbiased estimator can be found by consensus algorithm. Besides that, consensus idea is also being used in more general distributed computation in computer vision, modeling information and decision dynamics in social networking (learning, opinion dynamics, rule of thumb etc) [7].

2.2. Definition of Consensus

In any system, consensus means to reach an agreement of some common interest. The information state which needs to be coordinated between agents may be in different forms like velocity, position, oscillation phase, decision variable etc. However, agreement does not mean all agents will have the same quantity. Acceptable variance among node values can be accepted based on requirement of strictness of consensus.

Let us consider, a network of N nodes specified by graph $\mathcal{G} = (\upsilon, \varepsilon)$ where υ is the set of vertices (nodes) and ε is the set of edges, which is subset of $\{\{i, j\} : \{i, j\} \in \upsilon, i \neq j\}$. The nodes are connected by links. Since the graph is connected, there is a path between any two nodes i and j. Therefore, a path in $\mathcal{G}$ consists in a sequence of vertices $(i = i_1, i_2, \ldots\ldots i_r = j)$ such that $\{i_j, i_{j+1}\} \in \mathcal{G}$ for every $j \in \{1, 2\ldots\ldots r - 1\}$. Each node is given a time dependent finite value $x_i(k)$ at time k such that $x_i[k] \in \mathbb{R}$. The update method of any node i is as like below- $x_i[k+1] = \sum w_{ij}[k]x_j[k]$ where w_{ij} is the weight factor.
Our goal is to reach common value (i.e. average $\bar{x}$)asymptotically. The average is defined based on initial values.

$$\bar{x} = (1/N) \sum x_i[0]$$

A continuous time model can be summarized as

$$\dot{x}_i(t) = -\sum_{j \in S_i(t)} W_{ij}(x_i(t) - x_j(t)) \tag{1}$$

where $S_i(t)$ is the set of information to which i node's information belongs to at time t, and W_{ij} is the time varying weighting factor. The information of each agent is driven to the value of the neighboring agent and tends to reach on consensus.

In the matrix form, the equation (1) can be written as

$\dot{x} = -Lx$

where L is the laplacian matrix of the graph and $x = [x_1, x_2, ..., x_n]^T$.
Consequently, a discrete time model can be summarized as

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in S[k]} W_{ij}x_j[k] \tag{2}$$

where $\sum_{j \in S_i[k] \cup \{i\}} w_{ij} = 1$ and $w_{ij} > 0$ for $j \in S_i[k] \cup \{i\}$.

In the matrix form, the algorithm can be written assuming $w_{ii} = 0$.

$x[k+1] = wx[k]$ where w is the stochastic matrix with positive diagonal entry.

So in the discrete time model, each agent updates its value with a weighted sum of current value of neighbor nodes and its own. Though each node is aligning to the same value, each node may not have value to exchange. Consensus is said to be achieved for all agents if and only if $\| x_i - x_j \| \to 0$ as $k \to \infty$,$\forall i \neq j$.

In digital communication, continuous real value cannot be sent. For this, discretization of real value at discrete time is required, and consequently quantization is an integral process in this regard.

2.3. Background of Quantization and Consensus Parameters

Before going through the current status on consensus algorithm, I discuss the background of quantization and some consensus parameters.

2.3.1. Quantization

In the application of sensor network, the agents or nodes communicate among themselves via digital communication. However power and bandwidth constraints limit digital channels. This forces quantization on node values. Our focus is on a more realistic and practical digital channel, which is finite rate digital links. Since in digital communication, the nodes exchange only symbolic data in a finite alphabet, the nodes cannot know about the exact value of other nodes but rather some estimated values. We assume that before exchange of the data, the nodes do quantization. For the quantization, the error is introduced in real value and mean square error is observed in the simulation result. The goal is to reduce the error and speed up the convergence rate and keep quantization resolution as small as possible.

In the literature, mainly two kinds of quantization method are found- the deterministic uniform quantization and probabilistic (dithered) quantization.

2.3.2. Uniform Quantization

In uniform quantization [14], the continuous or discrete value is quantized to upper, lower or mid-value of the specific quantization level.

Suppose that the scalar value $x_i \in \mathbb{R}$ is bounded to finite interval $[U, -U]$. Further suppose that we wish to obtain a quantized message with length l bits, where l is application dependent. We, therefore, have $L = 2^l$ quantization points given by the set $\tau = \{\tau_1, \tau_2 ...., \tau_L\}$. This points are uniformly spaced such that $\Delta = (\tau_{j+1} - \tau_j)$ for $j \in \{1, 2.....L-1\}$. It follows that $\Delta = [\frac{2U}{2^l - 1}]$. Now suppose $x_i \in [\tau_{j-1}, \tau_{j+1}]$, then $q(x_i) = \tau_i$. So, here if the node value falls between a quantization interval, mid-value is taken for communication.

if q_i be the quantization set for value x and $\Delta \in \mathbb{N}$, the value will be in the set q_i such that $|q_i| < \delta\Delta$ where $\delta \to 0$.

2.3.3. Probabilistic Quantization

Suppose that the scalar value $x_i \in \mathbb{R}$ is bounded to finite interval $[U, -U]$. Further suppose that we wish to obtain a quantized message with length l bits, where l is application dependent. We, therefore, have $L = 2^l$ quantization points given by the set $\tau = \{\tau_1, \tau_2 ...., \tau_L\}$. This points are uniformly spaced such that $\Delta = (\tau_{j+1} - \tau_j)$ for $j \in \{1, 2.....L-1\}$. It follows that $\Delta = [\frac{2U}{2^l-1}]$. In the probabilistic quantization method, at time k, $x_i[k]$ is quantized by Bernoulli outcome

$P(q_i = \tau_{j+1}) = r$ and $P(q_i = \tau_j) = 1 - r$ where $r = \frac{x_i - \tau_j}{\Delta}$ and q_i is the quantized value of $x_i[k]$.

In the algorithm, using rounding or nearest integer approximation, finite or infinite level quantization, it is assumed that the initial average is kept constant through the averaging process. But in a practical case, some value is lost from exact value while doing quantization. So the nodes may not have reached strict consensus (i.e. they may not have exact value). Therefore, there will be average disagreement $(\phi(t) \neq 0)$ [9] and drift $(d(t) \neq 0)$[9]. But this may be justified for its convergence speed. It has great application where there is power and communication bandwidth constraint and near consensus is acceptable.

Drift : If N agents of a network have values $[x_1[k], x_2[k], x_3[k]............., x_n[k]]$ at time k, then drift [9] can be defined for $k \to \infty$ as $d[k] = 1/N \sum_i x_i[k] - 1/N \sum_i x_[k]$.

Average Disagreement : For N number of nodes with values $[x_1[k], x_2[k], x_3[k]............., x_n[k]]$, average disagreement [9] for time k can be defined as

$\phi[k] = [\frac{\sum_{ij} w_{ij}(x_i - x_j)^2}{(N\Delta)}]^{1/2}$

Where w = weight matrix and Δ = graph degree.

2.4. Review of Consensus Research

Research on consensus can be divided into two parts- (1) consensus based on non-quantized(real number transfer) communication (2) consensus based on quantized communication

2.4.1. Non-quantized Consensus

In the field of wireless sensor networking, extensive research has been done focusing on coordinated control and estimation [17]. In this area, the most crucial issue is coordinated and de-centralized consensus, which has been addressed by many researchers. Most of the research has been done based on modeling the communication between agents by a graph and assuming the communication medium is ideal (i.e. non-quantized real number is transferred between nodes). In this regard, much research has been conducted focusing on the design of effective averaging algorithm with slight modified graph. So, early research on consensus was based on non-quantized communication.

Early work on consensus was initiated by researchers lead by Boyd and Xiao [29], who introduced fast distributing linear averaging (FDLA), which can be summarized by

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in N} w_{ij}x_j[k]$$

In matrix form $x[k+1] = wx[k]$

They showed that the necessary condition for consensus is $1^T w = 1^T$, $w1 = 1$ and spectral radius, $\rho(w - (1/N)(11^T)) < 1$.

The authors proposed a heuristics weight matrix based on the Laplacian matrix as well as constant edge weight and local degree weights for fast convergence.

Afterwards, researchers have worked on average algorithm focusing on communication graph (i.e. constructing the weight matrix). Avrachenkov et al.[1] introduced the neighborhood algorithm for clustering structure topology of network. Each node i sets the weights for the link (i, j) depending on the similarity between its neighborhood set of the node i and the neighborhood of the node j. In order to quantify the similarity, the Jaccard index [1] has been utilized. For any neighbor set $N[i]$ for i and neighbor set $N[j]$ for j, Jaccard index [1] is defined as

$$J[N[i], N[j]] = \frac{|N[i] \cap N[j]|}{|N[i] \cup N[j]|}$$

The smaller the value of J, the more difference between node $N[i]$ and $N[j]$ and consequently larger weight is given and vice-versa. The main contribution of neighborhood algorithm is the convergence rate.

Sardellitti et al. [27] utilized homogenization mechanisms involving advection-diffusion process in the link between nodes and tried to find out the time varying weight factor for asymmetric graphs governing the interaction between sensors.

Let us consider the general form for N-dimensional system

$$x[k+1] = w[k]x[k]$$

where $w[k] = I - \epsilon L(k)$

and $L[k] = \gamma c[k]/a_c + (1-\gamma)D_f/a_d$

Here, matrix $c[k]$ and D_f account for the advection and diffusion mechanisms respectively, and ϵ, a_c, γ and a_d are the co-efficients of the first order partial differential equation of the system.

The algorithm induces a substantial increase in speed of convergence compared to the time invariant symmetric algorithm.

From the perspective of enhancing the weight matrix, Yuan et al. introduced the primal-dual sub gradient method [34] in which the goal is to minimize the sum of each agent's local convex objective function, subject to the global convex inequality constraints and a convex state constraints over network. For any distributed multiagent optimization problem, let us consider the objective function with constant a, b and eigen value λ and which is the sum of n quadratic functions

minimum $\sum_{i=1}^{n}[a_i||x||^2 + (b_i)^T x + (b_i)^T x]$

subject to $x^T x << 1$,

$$x \in R^m \tag{3}$$

The Lagragian function $L : R^m \times R_+ \to R$ associate with equivalence of (3) in which the inequality constraint is replaced by $n(x^T x - 1) \leq 0$, is given by

$$L(x, \lambda) = \sum_{i=1}^{n} [(a_i ||x||^2 + (b_i)^T x) + \lambda n(x^T x - 1)]$$

The weight matrix w is updated based on sub gradient and super gradient with respect to x and λ of Lagragian function L.

2.4.2. Quantized Consensus

From a practical point of view, in wireless communication real numbers cannot be transferred and quantization on real number needs to be performed. From this, consensus based on quantized communication originated, and research regarding this is moving forward.

In literature, mainly two types of algorithm are found- distributed averaging algorithm [3, 31, 2, 15, 19] and averaging with gossip algorithm [13, 18]. In the distributed averaging algorithm, each agent updates its state with a weighted sum of values from the neighboring nodes and its own value. The distributed algorithm has been summarized in the equation (1).

Paulo et al.[13, 6] has introduced new types of algorithm called random gossip algorithm. In this algorithm, two nodes are randomly selected and average is done quantizing the node value assuming that each node has integer value.

At each time, if an edge $\{i, j\} \in S$ is selected randomly with probability $P_{i,j}$ such that $\sum_{i,j \in S} P_{i,j} = 1$, then for probability matrix, $P \in \mathbb{P}^{N \times N}$

$$P_{ij} = P_{ji} = \begin{cases} P_{(i,j)} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where P_{ij} and P_{ji} are the probability of selecting edges (i, j) and (j, i) respectively.

The randomly chosen nodes adjourn their states by below formula [13, 6]:

$$x_i[k+1] = x_i[k] - \alpha q[x_i[k]] + \alpha q[x_j[k]] \tag{4}$$

$$x_j[k+1] = x_j[k] - \alpha q[x_j[k]] + \alpha q[x_i[k]] \tag{5}$$

where $q[x[k]]$ =quantized value of $x[k]$ and α is a constant.

They showed that the algorithm preserves average of the initial node values and approximately reaches to consensus. They also provide the bounds for convergence time for some specific network topologies. This algorithm can have great applications where there is power constraint, and near consensus is acceptable. But the main drawback of the algorithm is that the nodes may not have the same value after reaching the consensus.

Almost all the consensus building algorithms were based on single layer network. Y. Wan et al.[28] considered the multi group multi layer network for consensus building algorithm. They showed that for bipartite network i.e. network consists of controlling virtual fusion centers and general agents, the convergence depends on the routing structure. This can also be extended for more than two layers. For the desired convergence time and specific number of agents, we can design required number of virtual fusion centers [28].

Though, most of the consensus algorithms have been derived based on the memoryless system, yet some researchers have proposed algorithms with memory. Roy et al. has introduced linear distributing averaging algorithm [24] with memory. Let us consider, n agents having internal states $x_i[k]$ and storage state $z_i[k]$ where $i = [1, 2, 3.................n]$. The update method of the proposed algorithm [24] is

$x_i[k+1] = x_i[k] + q_i y_i[k] + r_i z_i[k] + s_i b_i$

$z_i[k+1] = t_i z_i[k] + v_i y_i[k]$

$wx = b$

Here q_i,r_i,s_i,t_i and v_i are the constant of the node i .

$w \in \mathbb{R}^{n \times n}$= the weighting square matrix

$b \in \mathbb{R}$ = some vector and

statistics $y_i = g_i^T x[k]$.

It was shown for the use of memory, the network reaches to consensus faster than the algorithm without memory.

As shown [3, 23, 15, 2], most of the research on consensus have been done based on simple communication system- single encoder and decoder. For integer approximation, single or finite bit quantizer along with single encoder and decoder is sufficient to exchange values between agents. In this regard, a distributed protocol[2] is designed with error compensation. The protocol [2] is characterized by three parameters- the control gain, the scaling function and the number of quantization levels. For a connected graph and fixed quantization levels, the control gain and scaling function can be chosen in a way that consensus can be achieved asymptotically. In fact, for the appropriate selection of these 3 control parameters, can lead to consensus with single bit quantizer.

Mehmet et al.[33] introduced an algorithm in which new technique of coding and decoding has been used. In coding process, previous exchanged data and current side information have been co-related to reduce mean square error up to the certain precision level. The author proposed a practical scalar quantizer based on prediction and nested lattice wyner-ziv encoding scheme and showed the detailed mathematical framework. For the suboptimal co-relation between previous data and current data, the MSE(mean square error) is reduced significantly and it also reduces the dependency of convergence speed on precision level(quantization level).

Some researchers have emphasized on transmission noise or quantization noise. In the early paper [30, 23, 33, 19, 11, 20, 25], the averaging algorithm was considered on noiseless communication medium. The averaging algorithm [16, 13, 8, 9] considers the communication medium among nodes non-perfect and affected by some error which can be either a random additive noise or produced by quantization process. The algorithm can be summarized as

$$\dot{x}_i(t) = -\sum_{j \in S_i(t)} w_{ij}(x_i(t) - x_j(t)) + n_i(t) \tag{6}$$

$$x_i[k+1] = w_{ii}x_i[k] + \sum_j w_{ij}x_j[k] + n_i[k] \tag{7}$$

where $n_i[t]$ is the white noise with zero mean and co-variance $\mathbb{E}[n(t)n(t)^T] = \mathbb{Z}$.
An approach to analyze quantized noise as white noise was not successful. So new agreement algorithm which preserves the initial average state was introduced, and it showed good performance and scalability with node numbers for noisy communication medium.

Consensus can be divided into two groups based on partially quantized [13] and globally quantized consensus [7]. Let us consider, two nodes i and j with value x_i and x_j respectively. The nodes update their values after global quantization [7] by

$$x_i[k+1] = 1/2[q(x_i[k]) + q(x_j[k])] \tag{8}$$

$$x_j[k+1] = 1/2[q(x_j[k]) + q(x_i[k])] \tag{9}$$

In this case, the average of the previous step is not saved and for that, significant drift [9] is observed. To overcome this problem, partially quantized algorithm [13] is deployed with non quantized previous node value.

$$x_i[k+1] = x_i[k] - \alpha q[x_i[k]] + \alpha q[x_j[k]] \tag{10}$$

$$x_j[k+1] = x_j[k] - \alpha q[x_j[k]] + \alpha q[x_i[k]] \tag{11}$$

From simulation result, it is observed that the drift is comparatively less in partially quantized algorithm than that of globally quantized algorithm.

Based on above discussion, the literature on consensus can be summarized as like below

Algorithom Type	Reference	Strategy	Quantization method	Drift (Differenc	Disagreement(dif	Consensus Type	Remarks
Distributed average Algorithm with least mean square deviation	[17]		no quantization	d(k)=0	has disagreement	Strict consensus	
Quantized consenus	[9]		Rounding	d(k)≠0	has disagreement	Near consensus	
Average consensus Algorithm with transmission noise or quantization	[18]	Drift can be minimize	Uniform Quantization	d(k)≠0	has disagreement	Near consensus	
Distributing Averaging Algorithm with quantization effects	[41]		uniform/probabilitic	d(k)≠0	has disagreement	Near consensus	Error depends on Quantization level
Algorithm with differential nested lattice coding	[16]		uniform/probabilitic	d(k)⟶0	MSE is reduced significantly	Strict consensus	Due to memory and battery life ,may not be feasible
Distributed average consensus with dithered quantization	[10]		Probabilitic quantization/dithered quantization	d(k)≠0	has disagreement	Strict consensus	feasible in case of Bandwidth /power

Average consensus Algorithm with quantized communication	[11]		uniform quantization	d(k)≠0	has disagreement	Near consensus	
Gossip Algorithom	[25]		Uniform quantization (rounding)	d(k)≠0	has disagreement	Near consensus	
Real value consensus over noisy quantized channel	[42]	Error compansated by feedback	Deterministic or probabilistic	d(k)→0		Strict consensus	
Distributed algorithm (Quantized data and Random link failure)	[43]		Dithered quantization	d(k)≠0	has disagreement	Near consensus	Time varying
Quantized consemus by gossip algorithm	[26]		uniform quantization	d(k)≠0	has disagreement	Near consensus	
Distributed consensus of multiagent system with finite level quantization	[38]	Error compensation by gain,scalling function and quantization level	uniform quantization	d(k)≠0	has disagreement	Near consensus	need more bit for faster rate
Quantized algorithm by quantized communication link	[28]	quantized feedback control	uniform	d(k)≠0	has disagreement	Near consensus	feasible for dynamic network

Table 2.1. Comparison between different type of algorithm with different quantization method

CHAPTER 3

AVERAGE QUANTIZED CONSENSUS BUILDING BY GOSSIP ALGORITHM USING 16 BIT QUANTIZATION AND EFFICIENT DATA TRANSFER

This chapter proposes an effective data transfer method of using 16 bit quantization on average consesus building by gossip algorithm.

3.1. Introduction

In average consensus by gossip algorithm, at each time, two randomly chosen nodes exchange their values, make average and finally reach to consensus. Most of the literature assume that the communication channel between nodes allows real number to transfer. But from practical point of view, the sensor network communicates by wireless fashion and energy and bandwidth limit the capacity of the channel. This suggests that communication channel is rather as digital channel. This clearly forces quantization on real number that nodes have to send. This issues have been noticed in [30], [8], [31] and [12]. The quantization effect [16, 31] was examined from the different point of view restricting the attention on integer value and introduced the definition of quantized communication. The quantized communication is achieved by the vector x for $i = 1, 2, 3, \ldots\ldots\ldots, N$ if it belongs to the set S defined as $S = \{x : \{x_i\}^N \in \{L, L+1\}, \sum_{i=1}^{N} x_i = T\}$ with T and L being the sum of initial node values and an integer respectively. It was shown that under some constraints and restricted communication, the network reaches to consensus. But the consensus is not clearly strict sense consensus, the nodes may not have exactly same value.

Various consensus methods are found based on bandwidth limitation and strictness of consensus. The quantization method [13] takes the closest integer value adding some constant and the network is converged into consensus up to the size of the quantization bin for any connected undirected graph. But in this method, significant mean square error is seen from average of agents' initial values (at time $k = 0$) and consensus is not strict. So the algorithm calls further modification. In my study, I propose that the 16 bit quantization for node values. Since we are reducing the resolution size of quantization, 16 bit represents the node value more accurately compared to integer approximation. Also I propose the optimization of bit transmission. We transmit the 16 bit index of quantized node value at one cycle and 8 bit MSB (most significant bit) of the 16 bit index at the next 4 cycles. At the receiving end, from the index, the node value is reconstructed using codebook as like pulse code modulation. For optimizing of bit transmission, the network also converges to average consensus and the mean square error from initial average is reduced significantly as compared to integer approximation of the node value. Since we send MSB in the 1-4th cycle and MSB+LSB in the 5th cycle, in the 1-4 cycle, the bandwidth is 50% less utilized. So, a significant amount of energy is saved.

In the following section, statement of problem, the proof of convergence, convergence speed and simulation result is shown respectively.

3.2. Problem Statement

Let us consider, a network of N nodes specified by graph $\mathcal{G} = (\upsilon, \varepsilon)$ where υ be the set of vertices (nodes) and ε be the set of edges, which is subset of $\{\{i, j\} : \{i, j\} \in \upsilon, i \neq j\}$. The nodes are connected by links. Since the graph is connected, there is a path between any two nodes i and j. Therefore, a path in $\mathcal{G}$ consists of a sequence of vertices $(i = i_1, i_2, ...i_r = j)$ such that $\{i_j, i_{j+1}\} \in \mathcal{G}$ for every $j \in \{1, 2.....r-1\}$. A graph is fully connected or complete if $\varepsilon = \{\{i, j\} : \{i, j\} \in \upsilon, i \neq j\}$. Each node is given a time dependent finite value $x_i[k]$ at time k such that $x_i[k] \in \mathbb{R}$. Since the nodes are connected by digital links, they can not extract the real value but the quantized (16 bit) value of exact value. Since $\mathcal{G}$ is a undirected graph of N node, at each time step, an edge $(i, j) \in \upsilon$ is chosen randomly with probability $P_{(i,j)}$ such that $\sum_{(i,j)\in\upsilon} P_{(i,j)} = 1$.
Let, the probability distribution $P \in \mathbb{R}^{N\times N}$ be, then

$$P_{ij} = P_{ji} = \begin{cases} P_{(i,j)} & \text{if \{i,j\}} \in \nu \\ 0 & \text{otherwise} \end{cases}$$

where P_{ij} and P_{ji} are the probability of selecting edge (i,j) and (j,i) respectively.

The randomy choosen nodes adjourn their states by below formula [13]:

$$x_i[k+1] = x_i[k] - \alpha q[x_i[k]] + \alpha q[x_j[k]] \tag{12}$$

$$x_j[k+1] = x_j[k] - \alpha q[x_j[k]] + \alpha q[x_i[k]] \tag{13}$$

where $q[x[k]]$ =16 bit quantization of $x[k]$ and $\alpha = \frac{1}{2}$.

Now, we define quantized average consensus. A quantized average consensus is a state $\bar{x} \in \mathbb{R}^{\mathbb{N}}$ such that $|\bar{x}_i - N^{-1} \sum_{j=1}^{N} x_j[0]| < 1$. If for any time k, there exists a convergence time T_{con} such that $k > T_{con}$, then the network is said converged into consensus and x[k] is a quantized average consensus state.

3.3. Quantization Method

Suppose that the scalar value $x_i \in \mathbb{R}$ is bounded to finite interval $[U, -U]$. Further suppose that we wish to obtain a quantized message with lenght l bits, where l is application dependent. We, therefore, have $L = 2^l$ quantization points given by the set $\tau = \{\tau_1, \tau_2 ...., \tau_L\}$. This points are uniformly spaced such that $\Delta = (\tau_{j+1} - \tau_j)$ for $j \in \{1, 2...., L-1\}$. It follows that $\Delta = [\frac{2U}{2^l - 1}]$. Now suppose $x_i \in [\tau_{j-1}, \tau_{j+1}]$, then $q(x_i) = \tau_i$.

3.4. Convergence Time

Gossip algorithm with real value reaches to consensus exponentially fast and for averaging time ϵ $(0 < \epsilon < 1)$, its convergence time T_ϵ [6, 10] is

T_ϵ =supinf$_{x(0)}\{t : P(\frac{||x(t)-x_{ave}1||_2}{||x(0)||_2}) \geq \epsilon) \leq \epsilon\}$

[1] Thus the averaging time ϵ is the smallest time it takes for $x(.)$ to get within ϵ of $x_{ave}\mathbf{1}$ with high probability, regardless of the initial value x[0].

For equal probability of selecting any edge

- ☐ $T_\epsilon = \Theta(N)$ for $N \to \infty$ for complete graph
- ☐ $T_\epsilon = \Theta(N^3)$ for $N \to \infty$ for ring graph

Now we see the probabilistic analysis for the bounds of convergence time. Let $T_1(x)$ be the random variable denoting the time of the first non-trival averaging when $x[0] = x$. For the given graph $\mathcal{G}$ and the given probability distribution on the edge, lets define $\bar{T}(\mathcal{G}) = \max_x E[T_1(x)]$, where the maximum expectation is taken over all possible initialization x that are not quantized consensus distribution and for which $m \leq x_i \leq M$ for $i = 1, 2, ...N$. Since for all such initializations, the number of non-trival averagings required is at least 1 and at most $\frac{(M-m)N}{8}$, so it follows that the expection of final convergence time T_{con}

$$\bar{T}(\mathcal{G}) \leq \max_{x:m\leq x_i\leq M} E[T_{con}(x)] \leq \frac{(M-m)^2 \times N \times \bar{T}(\mathcal{G})}{8}$$

The upper bound can be found with high probability [16]. This is very conservative method for simulation.

Let us define, a function of x[k], $V[x[k]] = x^*[k]\Omega x[k] = (||x[k] - x_{ave}||_2)^2$ where $\Omega = \mathbf{I} - N^{-1}\mathbf{1}\mathbf{1}^*$, $x^*[k]$ is the transpose of x[k] and $\mathbf{I}$ is the identity matrix.

Now for any node (i, j), $\sum\limits_{(i,j)} P_{(i,j)}(x_i[k] - x_j[k])^2 = 2x^*[k](diag(P\mathbf{1}) - P)x[k])$ where $P_{(i,j)}$ is the probability of selecting edge (i, j) and P is the probability distribution on all edges. Using the result from [13], the expected value of v at time (k+1)

$$E[v(x[k+1])] \leq (1-\lambda)E[v(x[k])] + \Delta$$

where λ is the smallest igenvalue of $(diag(P\mathbf{1}) - P)$ and Δ is the resolution of 16 bit quantization.

[1] $||x||_2 = (\sum_i x_i^2)^2$

For recurrent operation, we can argue that

$E[v(x[k+1])] \leq (1-\lambda)^k E[v(x[t])] + \frac{1-(1-\lambda)^k}{\lambda}\Delta$

and then $E[v(x[k])] \leq (1-\lambda)^t E[v(x[0])] + \frac{1}{\lambda}\Delta$

So, $v(x[k])$ decreases to $(1-\lambda)$ rate initially and saturates to $\frac{\Delta}{\lambda}$. Notice that λ is the function of P and topology of the graph.

For any edge $\{i,j\} \in \varepsilon$, $P = \frac{1}{|\varepsilon|}$

where $|\varepsilon|$ is the cardinality of $\mathcal{G}$.

so $diag(P\mathbf{1}) - P = \frac{1}{|\varepsilon|}L_{\mathcal{G}}$ where $L_{\mathcal{G}}$ is the laplacian matrix of ε.

As shown[13], for complete graph $\lambda = \frac{2}{N-1}$ and for ring graph $\lambda = \frac{2}{N}(1 - cos(\frac{2\pi}{N}))$ and then $\lambda = \frac{4\pi^2}{N^3} + o(\frac{1}{N^3})$for $N \to \infty$.

3.5. Bandwidth Save Calculation

In our algorithom, we send 8bit MSB of the 16 bit quantized values through 1st to 4th cycle and whole 16 bit in the 5th cycle.

In the 1 - 4 cycle, 50% bandwidth is saved. In the 5th cycle, 100% bandwidth is utilized.

So total bandwidth save in a period of 5 cycle $=\frac{0.5\times4+0}{5}$ $\times$100= 40 %

3.6. Simulation Result

From the simulation graphs (fig 3.1,3.2,3.3,3.4,3.5,3.6), we remark that for all method, the networks reach to consensus based on definition of our consensus $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$. After some iterations, the mean square error for all quantization methods become constant.

We also find that for all through 8 bit data transfer, the network does not reach to consensus. However, for 16 bit convergence is attained whcih is reflectred on fig 3.7.

With the increase of number of nodes, the convergence time also increases as shown in fig 3.8. The relation between convergence time and node is linear (i.e. $T_\epsilon = \Theta(N)$ for $N \to \infty$).

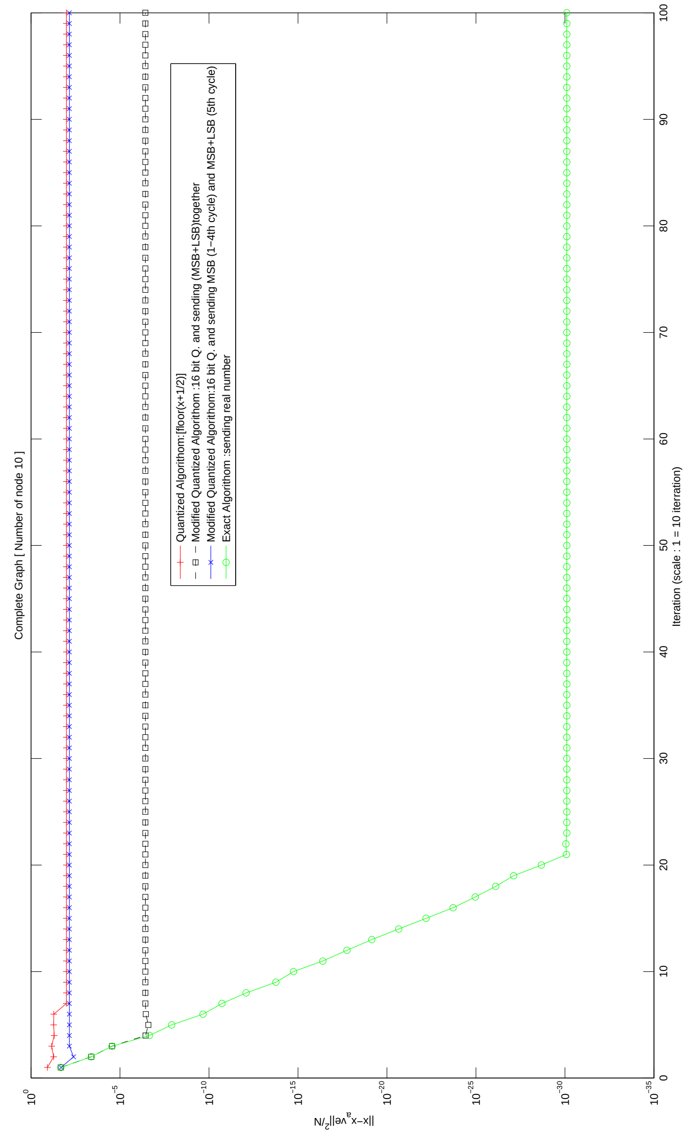

Figure 3.1. The mean square error for N=10 nodes. Remark that(1) after some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization < algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)< algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

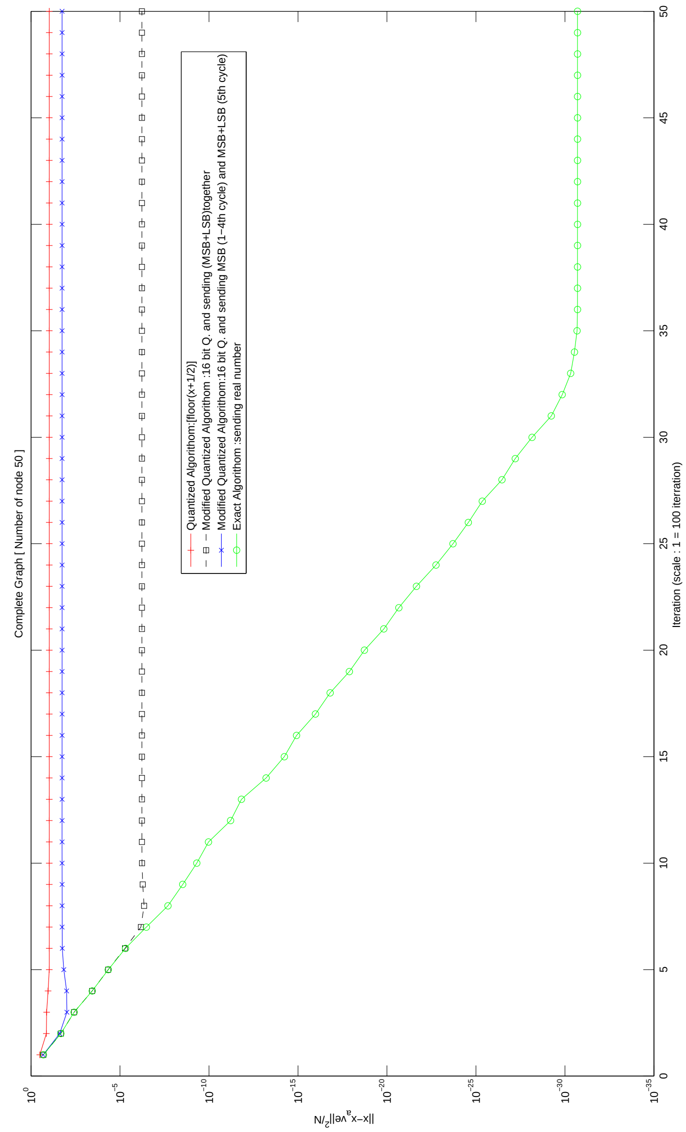

Figure 3.2. The mean square error for N=50 nodes. Remark that(1) after some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization < algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)< algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

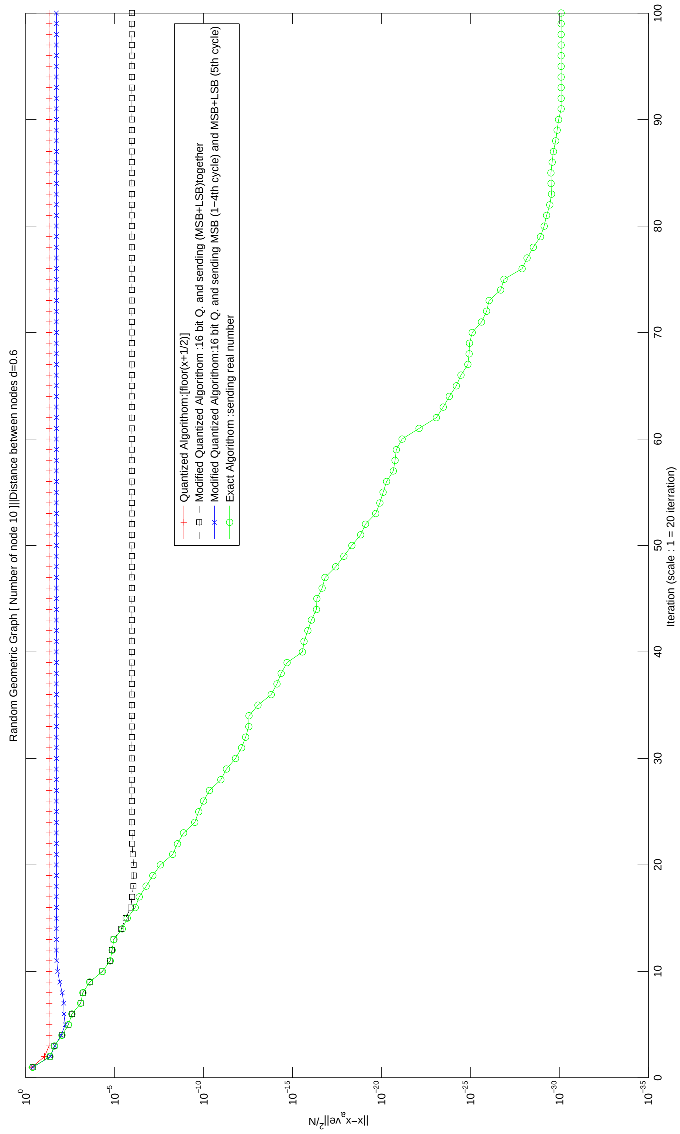

Figure 3.3. The mean square error for N=10 nodes. Remark that(1) After some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization < algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)< algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

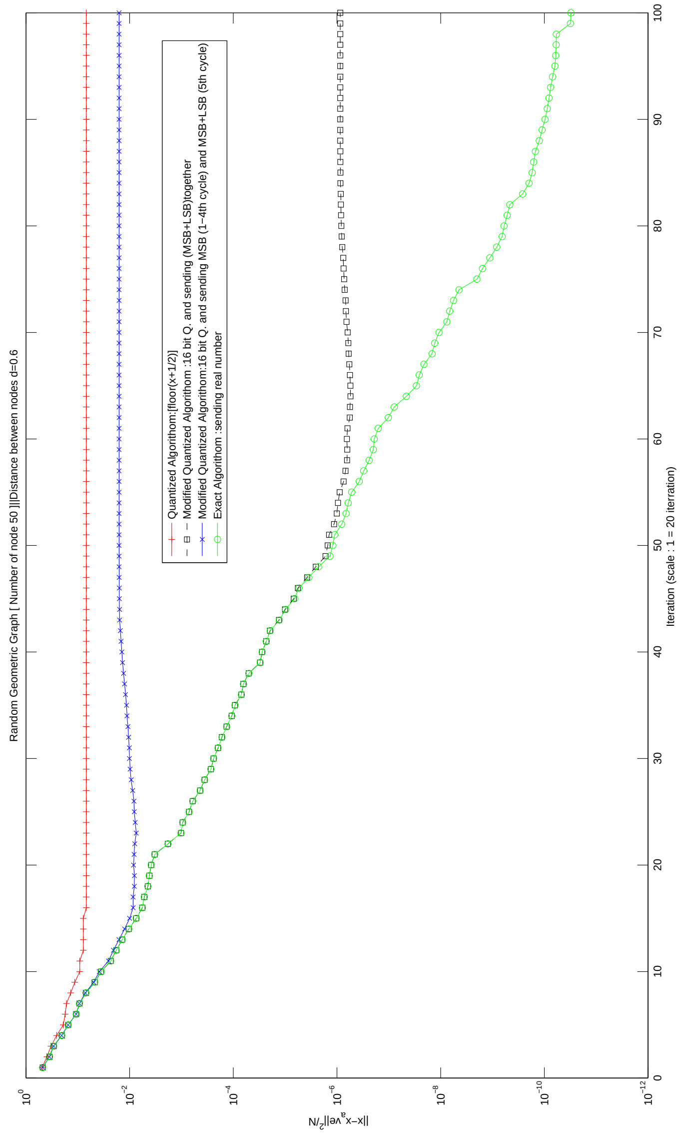

Figure 3.4. The mean square error for N=50 nodes. Remark that(1) after some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization < algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)< algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

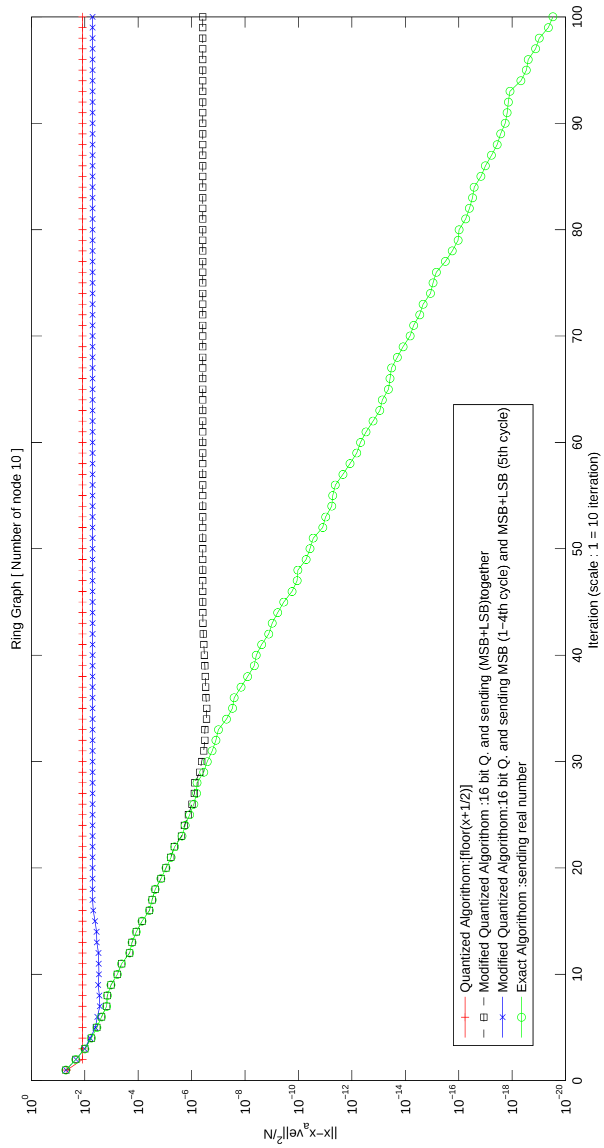

Figure 3.5. The mean square error for N=10 nodes. Remark that(1) after some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization < algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)< algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

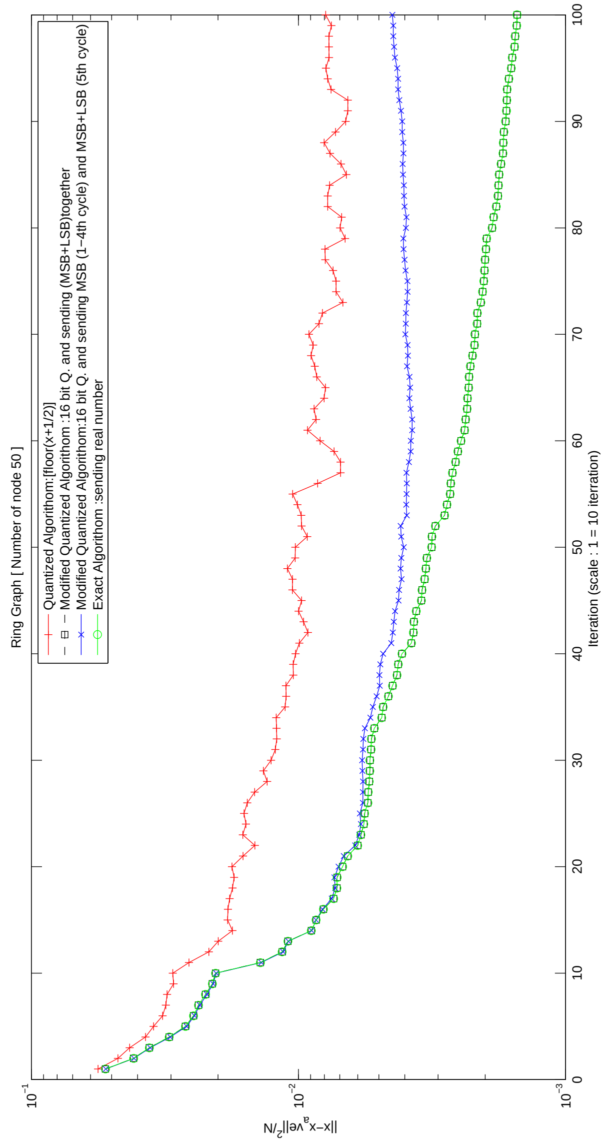

Figure 3.6. The mean square error for N=50 nodes. Remark that(1) after some iteration, three method except algorithm using real number, becomes almost constant (2) since we have defined average consensus by $|\bar{x}_i - \frac{1}{N}\sum_i^N x_i| < 1$, the agents reach to consensus by our agorithm (3) with increase of iteration, the mean square error in algorithm sending real number< algorithm with 16 bit quantization <algorithm with 16 bit quantization (sending MSB 1-4 cycle and (MSB+LSB) in 5 cycle)<algorithm sending approximate integar value($\lfloor x + 1/2 \rfloor$)

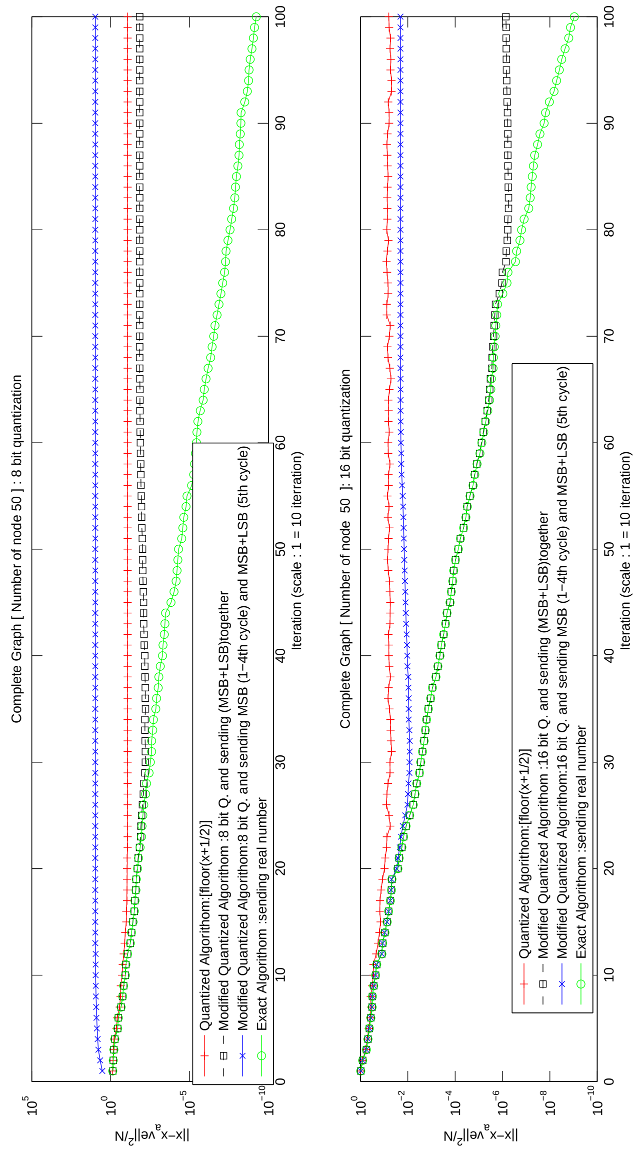

Figure 3.7. Effect of bit number on consensus. Remark that for only 8 bit data communication and based on our defination, the nodes do not converge to consensus

Another issue which is obvious that at a specific iteration number, the mean square error for large number of node is more than small number of nodes.

From the simulation graphs (fig 3.1,3.2,3.3,3.4,3.5,3.6), it is observed that the mean

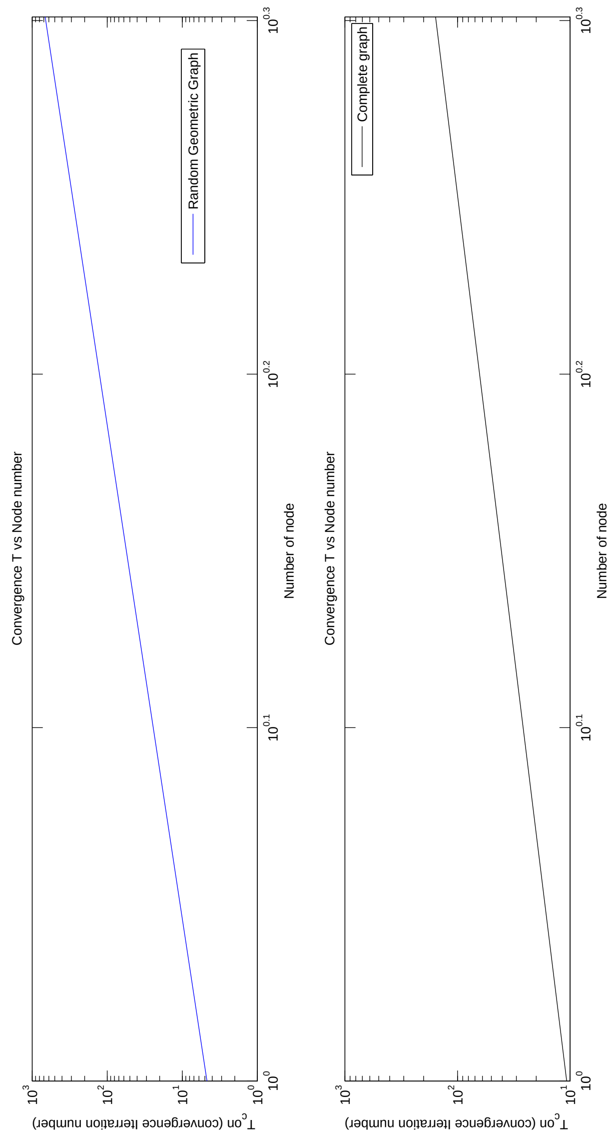

Figure 3.8. Convergence time vs node number. Note that with the increase of node number, the convergence time also increases.

square error for real number is the least and the mean square error for quantization method ($\lfloor x + 1/2 \rfloor$) is the highest. At the same time, mean square error for the algorithm with 16 bit (sending MSB in 1-4 cycle & MSB+LSB in 5 cycle) is greater than the algorithm with all through 16 bit quantization but less than the algorithm with quantization method ($\lfloor x + 1/2 \rfloor$). From fig 3.1, we find below data at iteration number 300

Table 3.1. Mean square error for Random Geometric Graph at iterration 300

Quantization method	**Node No = 10**	**Node No= 50**
quantization method($\lfloor x + 1/2 \rfloor$)	10^{-2}	10^{-1}
16bit quantization (sending in 1-4 MSB, MSB+LSB in 5)	5.5^{-3}	10^{-2}
16 bit quantization	10^{-7}	10^{-6}
Exact algorithm	10^{-30}	10^{-28}

From the table, we can conclude that algorithm with 16 bit (sending MSB 1-4 cycle MSB+LSB in 5 cycle) is better that the algorithm with quantization method ($\lfloor x + 1/2 \rfloor$).

3.7. Concluding Remarks

In most of the literature, the node value is assumed to be the nearest approximate integer value which includes large mean square error from average value and the network is converged to consensus in weaker sense. The quantization method introduced in [13] leads degradation $|q[x] - x| \leq \frac{1}{2}$ where $x \in \mathbb{R}$ is the node value. But in our algorithom, degradation $|q[x] - x| \leq \Delta$. For the smaller step size in quantization, the mean square error reduces significantly which is reflected in the simulation result.

From the simulation result, it is evident that as long as the distance from consensus is much larger than the quantization step, the speed of convergence is almost same as the non-quantized algorithom. Therefore, when we are near the agreement, the granularity effect comes out and the full understanding of the algorithom is based on non-quantization approximation, analysis of integer dynamics and quantization method.

Since in our algorithom, the optimization of data transfer by sending MSB (most significant bit) and LSB(least significant bit) seperately save energy, it can have great application on large network design specially when nodes are very far from each other, nodes are added/dropped frequently and energy/bandwidth constraint is a major issue.

CHAPTER 4

AVERAGE DISTRIBUTED CONSENSUS BY DITHERED QUANTIZATION (PQDA)

This chapter discusses the average distributed consensus by dithered quantization (probabilistic quantized distributed averaging) on four types of networks- random geometric graph, star graph, ring graph and slow co-herency graph.

4.1. Motivation

Average distributing averaging [5] is an attractive algorithm for wireless network. The reason behind this, is that there is no need of making complicated routing and every node only maintains the values of their immediate neighbors. Since nodes are not included in the spanning tree of the network, there is no probability of failure of the algorithm. Another attraction of the distributed averaging algorithm is at any time, the value of any agent represents the consensus value.

In the literature, we find mainly two kinds of quantization method- uniform quantization method [14] and probabilistic quantization method[5]. In the deterministic quantization method, if a value falls between two points of a quantization level, midpoint, lower value or upper value of the two points are taken as quantized value.

On the other hand, in the probabilistic quantization method [5], dither is mixed randomly and then uniform quantization is performed accordingly. Since random value is mixed, the network is reached to the consensus of initial average in expectation. Also, the distributed averaging algorithm has analyzability for the bound of mean square error and convergence time.

4.2. Background

In the sensor networking, the network is modeled by the graph i.e. the nodes are treated as points and network structure is defined based on transition from one node to another node.

In our experiment, we will use mainly four types of network structures- (1)Random Geometric Graph (2)Ring Graph (3) Star Graph (4) Slow Co-herency Graph

Random Geometric Graph: Random geometric graph is random undirected graph bounded by some boundary usually in unit torus $[0, 1)^2$.

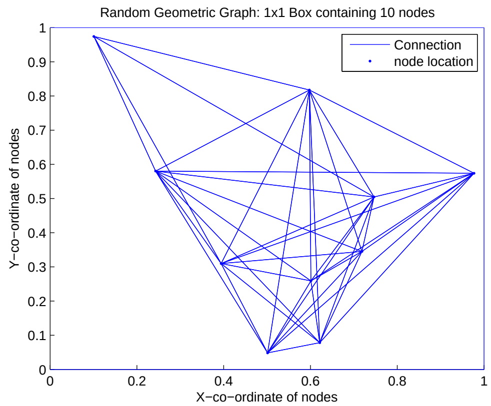

Figure 4.1. Random geometric graph

Ring Graph: Ring graph is a graph in which every node is connected to immediate two neighbour nodes and node connection type is just like a ring.

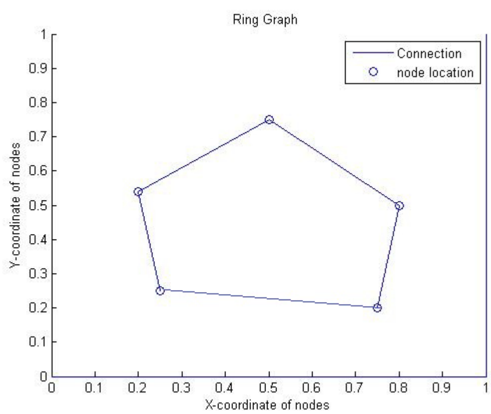

Figure 4.2. Ring graph

Star Graph: Star graph is a graph in which a central node is connected to other nodes and node connection is just like a star.

Slow Co-herency Graph: Slow Co-herency graph is a graph in which agents are segmented into some clusters and the agents of a cluster are considered to one agent in respect

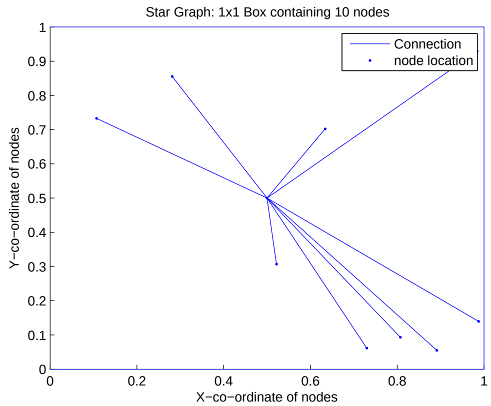

Figure 4.3. Star graph

of some paramenter (as for example - distance).

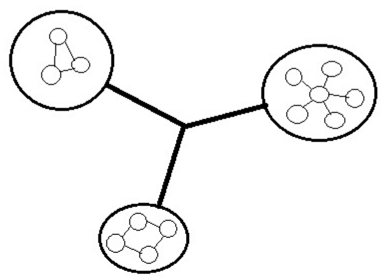

Figure 4.4. Slow co-herency

Quantization method: Suppose that the scalar value $x_i \in \mathbb{R}$ is bounded to finite interval $[U, -U]$. Further suppose that we wish to obtain a quantized message with length l bits, where l is application dependent. We, therefore, have $L = 2^l$ quantization points given by the set $\tau = \{\tau_1, \tau_2 ...., \tau_L\}$. This points are uniformly spaced such that $\Delta = (\tau_{j+1} - \tau_j)$ for $j \in \{1, 2.....L-1\}$. It follows that $\Delta = [\frac{2U}{2^l - 1}]$. In the probabilistic quantization method,at time k, $x_i[k]$ is quantized by Bernoulli outcome

$P(q_i = \tau_{j+1}) = r$ and $P(q_i = \tau_j) = 1 - r$ where $r = \frac{x_i - \tau_j}{\Delta}$ and q_i is the quantized value of $x_i[k]$.

4.3. Problem Statement

Let us consider, a network of N nodes with values $x_i[k]$ at time k where $i = 1, 2, 3......, N$ and $x_i \in [U, -U]$. If w_{ij} is the weight matrix for any node i and j, then

$x_i[k+1] = w_{ii} q_i[k] + \sum_{j \in N} w_{ij} q_j[k]$

Setting $w_{ii} = 0$, we can write in matrix form

$x[k+1] = wq[k]$

where $q[k] = Q(x[k])$ =quantized value of $x[k]$.

To analyze the performance, we will use normalize error (Distance between maximum and minimum node value). So for non-quantized value,

Normalized error, $r_x[k] = V_{max}[k] - V_{min}[k]$

where V_{max} and V_{min} are the maxmimum and minimum value of nodes at time k respectively.

For quantization,

Normalized error, $r_q[k] = Q(V_{max}[k]) - Q(V_{min}[k])$

where V_{max} and V_{min} are the maxmimum and minimum value of nodes at time k.

The theoritical upper bound [4] of the normalized error,

$r_{theo}[k] = \sqrt{N/(N-1)}\rho(w-J)^k r_x[0] + 2\Delta$

where w= weight matrix

$J = (1/N)\mathbf{1} * \mathbf{1}^\top$

$r_x[0]$= Non-quantized error at time k=0

Δ= Quantization resolution

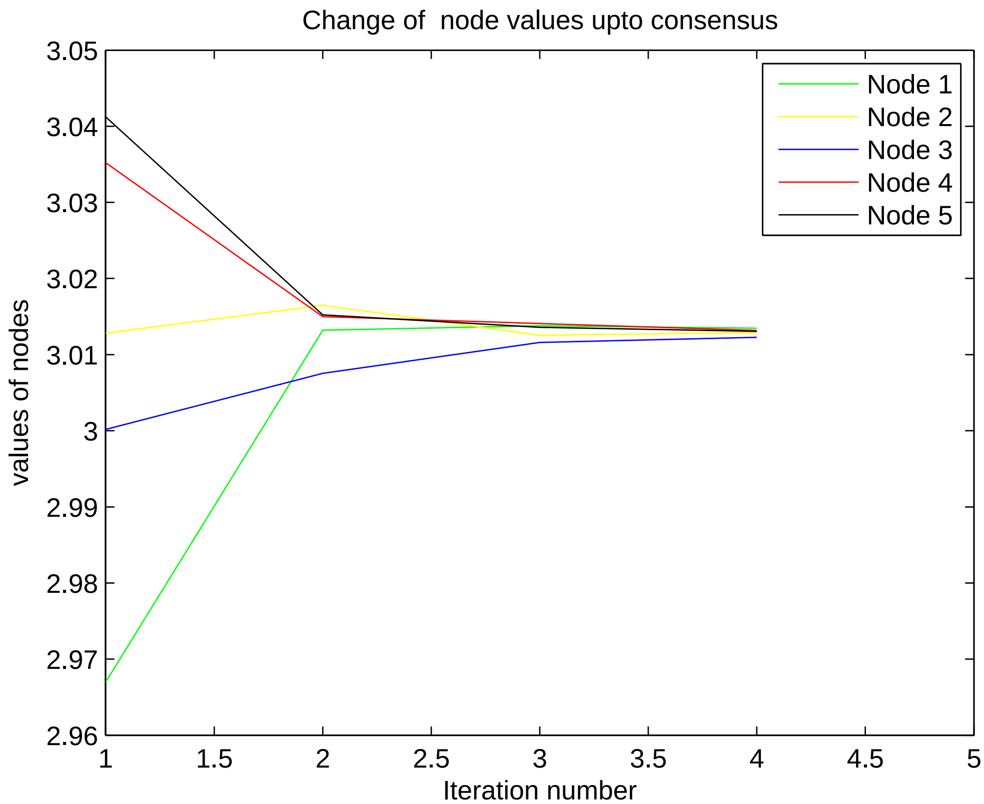

Figure 4.5. Convergence of nodes (Random geometric graph)

4.4. Simulation Result

In this part, distributed averaging algorithm has been implemented in four types of networks- ring graph, star graph, random geometric graph and slow co-herency graph.

From the simulation graphs (fig 4.5,4.7,4.9,4.11), it is obvious that the nodes reach to consensus for all four types of networks. Based on the network structure, different networks take different iteration numbers for reaching consensus for same number of nodes. According to graphs, for 10 number of nodes, the convergence time for random geometric, ring, star and slow coherency graph are 4, 9, 10 and 16 respectively. Random geometric graph takes the least time and slow coherency graph takes the highest time for convergence.

From normalize error graphs (fig 4.6,4.8,4.10,4.12), we find that distance between maximum and minimum node value(normalized error) decreases with number of iterations. But after some iteration numbers, the normalized error reaches to saturation. For random

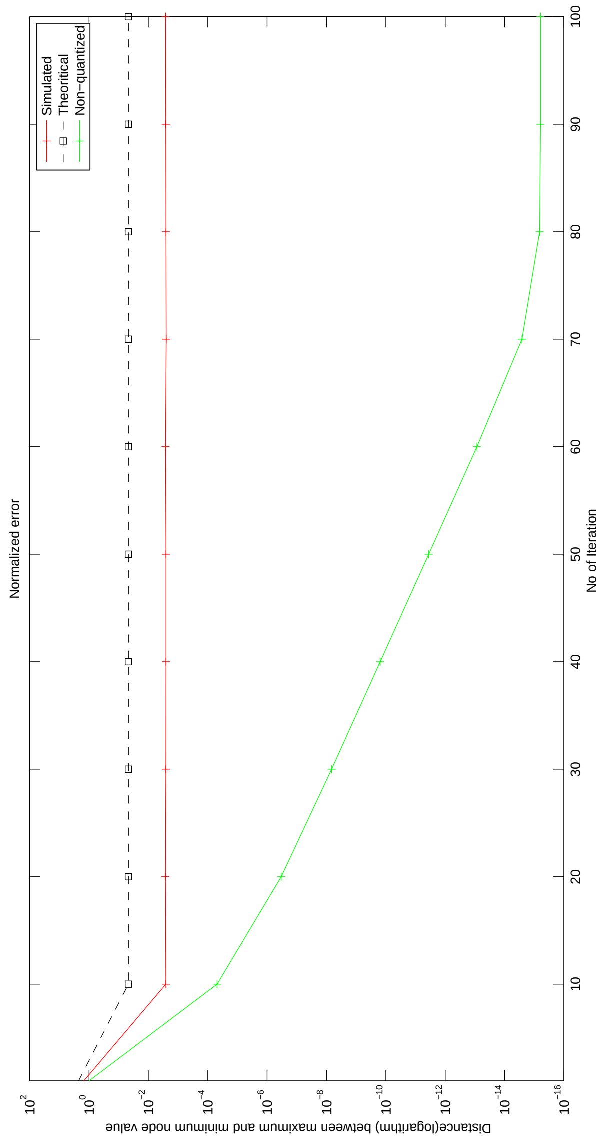

Figure 4.6. Normalized error between maximum and minimum node value(Random Geometric Graph)

geometric graph, at iteration number 100, the normalized distance between maximum and minimum node (normalized error) value is

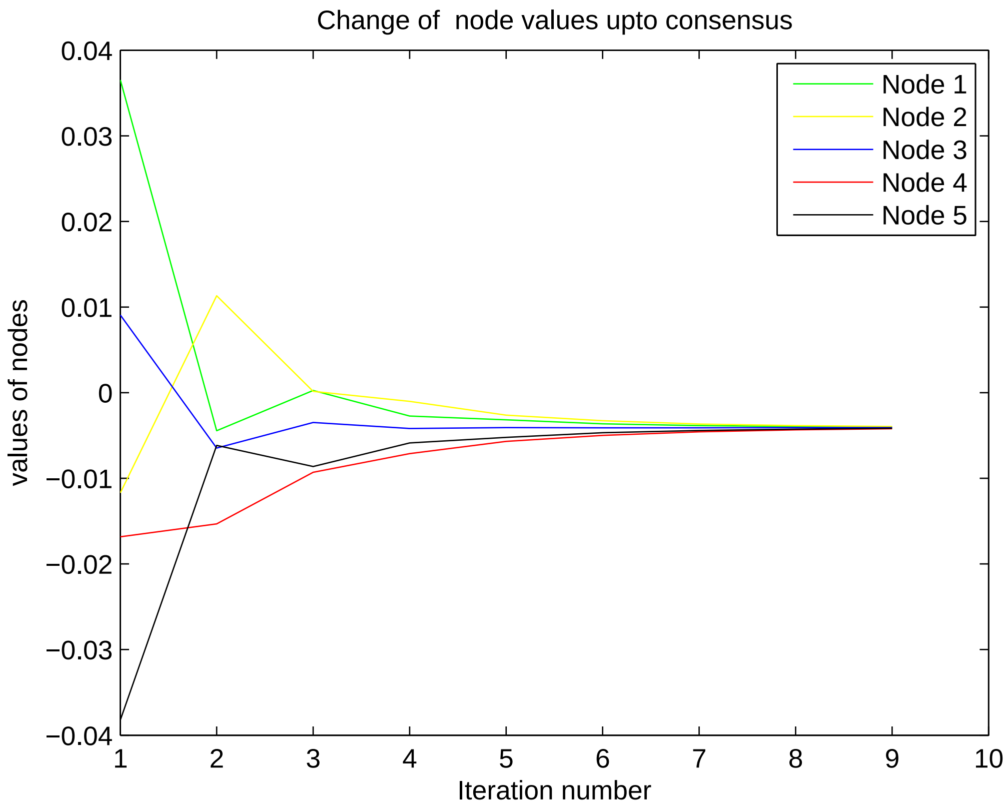

Figure 4.7. Convergence of nodes (Ring graph)

Table 4.1. Mean square error for Random Geometric Graph at iterration 100

Method	**Normalized distance(normalized error)**
Theoritical upper bound	10^{-1}
Simulated (dithered quantized)	10^{-3}
Non quantized	10^{-15}

So, the normalized error for dithered quantization is less than the upper bound of error and greater than that of non-quantization algorithm of averaging.This is also true for other types of network which is reflected in the simulation graphs.

4.5. Concluding Remarks

When the distance among nodes are far from the quantization step, the convergence speed is equal to non-quantized averaging algorithm. But when the distance falls between two bins, convergence rate decreases significantly.

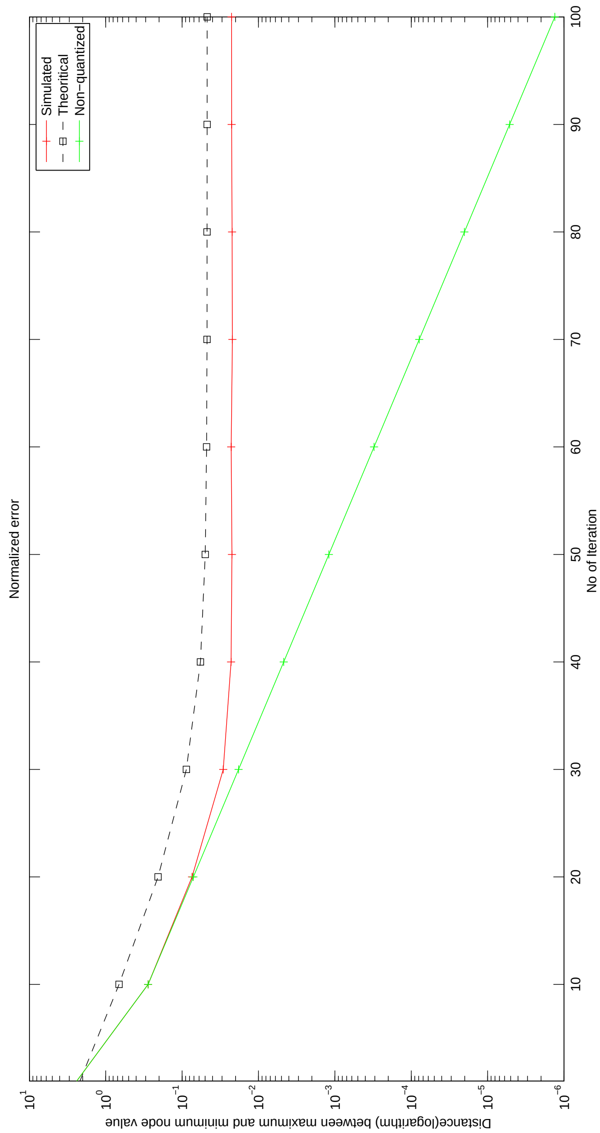

Figure 4.8. Normalized error between maximum and minimum node value (Ring graph)

Since distributed averaging using dithered quantization has some excellent features and reliability, it can have great application in power/bandwidth constrained network and in case of large volume of data.

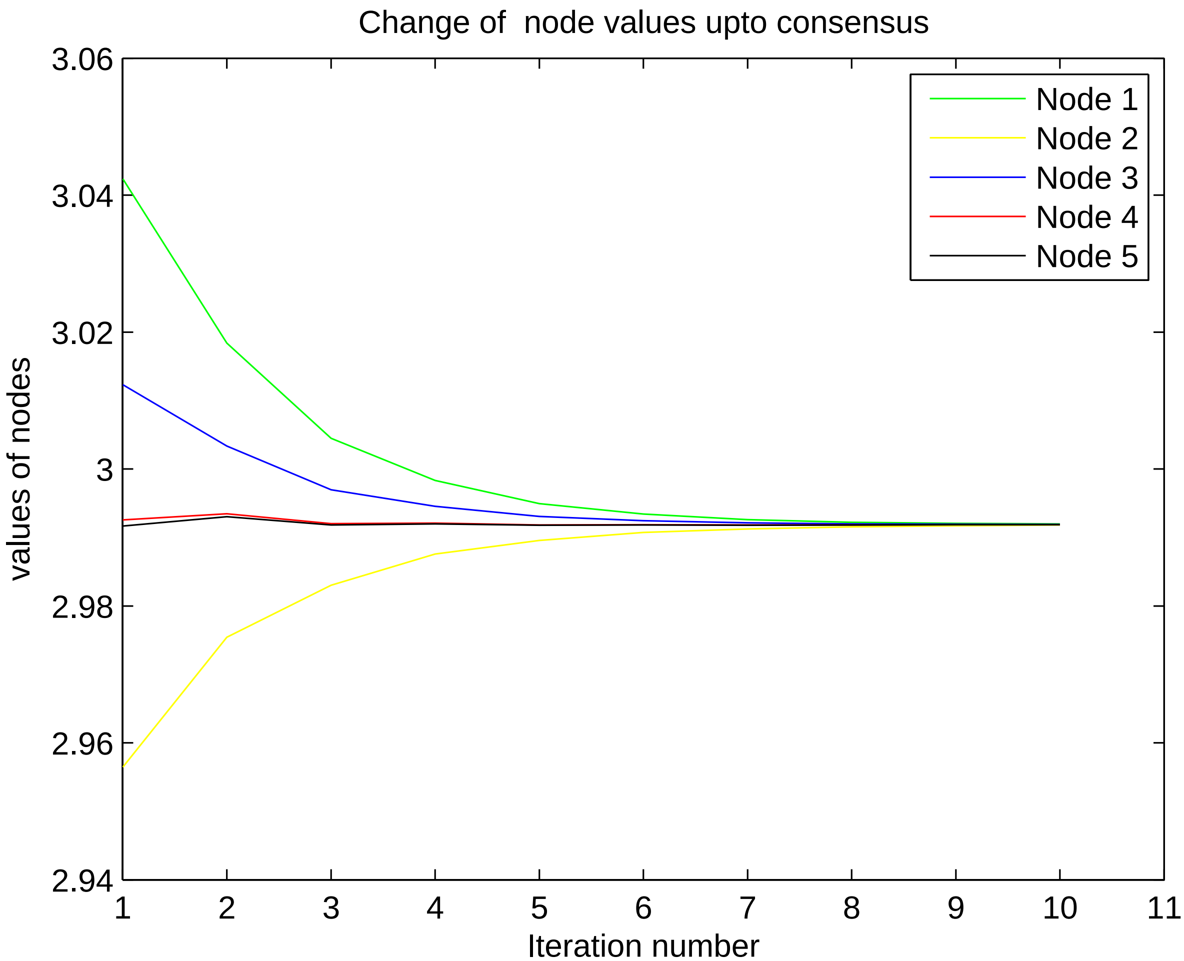

Figure 4.9. Convergence of nodes (Star Graph)

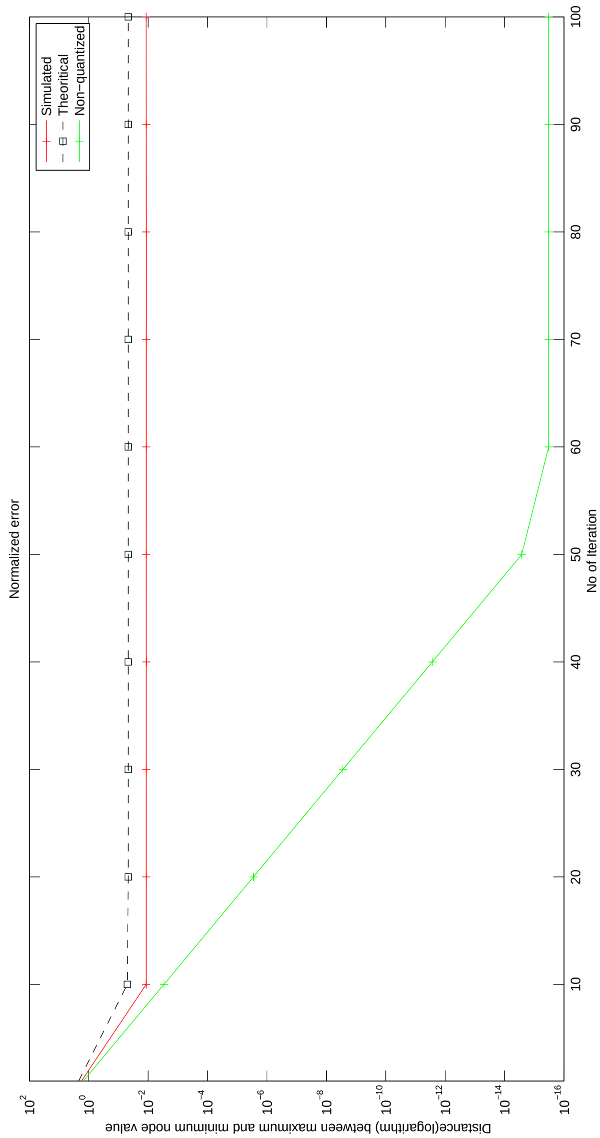

Figure 4.10. Normalized error between maximum and minimum node value(Star graph)

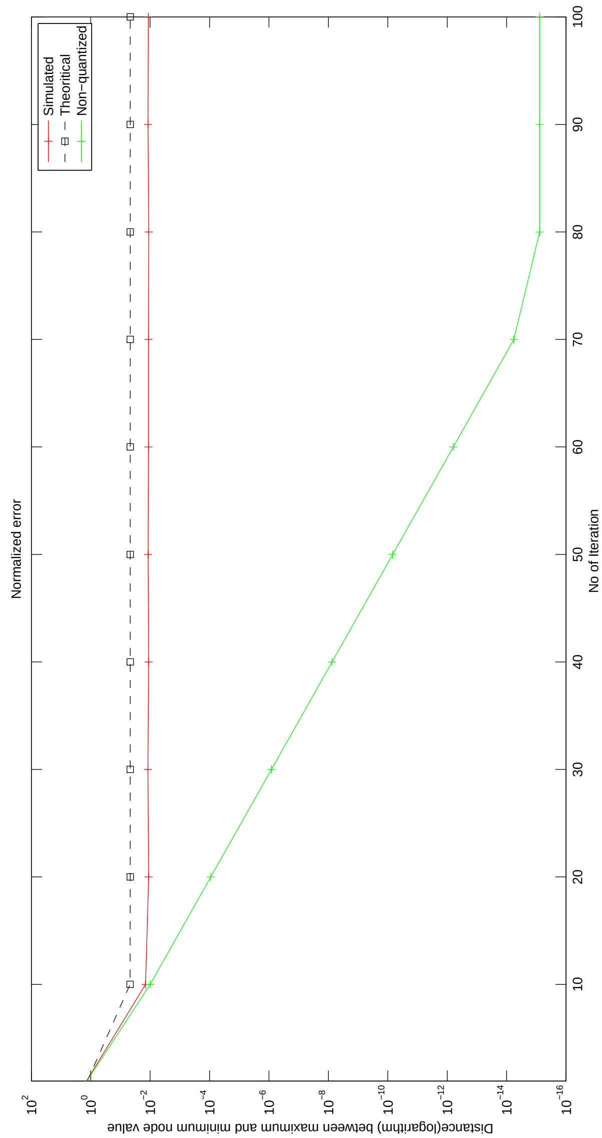

Figure 4.12. Normalized error between maximum and minimum node value (Slow co-herency graph)

CHAPTER 5

CONCLUSION AND FUTURE WORK

The gossip algorithm with 16 bit quantization lowers the mean square error and power consumption significantly. In the algorithm, we propose a period of 5 cycles based on common 16 bit quantization and PCM modulation technique. But optimized period is kept remained for further investigation based on number of agents, network structure and quantization resolution. The number of cycles in a period may vary with number of bit send, network structure (whether complete graph, star graph or slow co-herency) and number of nodes.

In the distributed averaging with probabilistic quantization algorithm, the algorithm has been implemented on four types of graph- random geometric, star, ring and slow co-herency graph. The random geometric graph, star and ring graph give the same result. However, slow co-herency demonstrated different result. It requires further theoretical investigation. Moreover, changing quantization method and considering noisy channel may be further topics for research on this sector.

Let us consider, a network of N nodes specified by graph $\mathcal{G} = (\upsilon, \varepsilon)$ where υ be the set of vertices (nodes) and ε be the set of edges, which is subset of $\{\{i, j\} : \{i, j\} \in \upsilon, i \neq j\}$. Since $\mathcal{G}$ is a undirected graph of N node, at each time step, an edge $(i, j) \in \upsilon$ is chosen randomly with probability $P_{(i,j)}$ such that $\sum\limits_{(i,j)\in\upsilon} P_{(i,j)} = 1$.

Let, the probability distribution $P \in \mathbb{R}^{N\times N}$ be, then

$$P_{ij} = P_{ji} = \begin{cases} P_{(i,j)} & \text{if \{i,j\}} \in \nu \\ \\ 0 & \text{otherwise} \end{cases}$$

where P_{ij} and P_{ji} are the probability of selecting edge (i, j) and (j, i) respectively.

The randomy choosen nodes adjourn their states by below formula:

$$x_i[k+1] = x_i[k] - \alpha q[x_i[k]] + \alpha q[x_j[k]]$$

$$x_j[k+1] = x_j[k] - \alpha q[x_j[k]] + \alpha q[x_i[k]]$$

where $q[x[k]]$ =16 bit quantization of $x[k]$ and $\alpha = \frac{1}{2}$

For a network of N nodes, average value at each step.

$x_i[k+1] = 1/N[x_1[k] + x_2[k] + \ldots\ldots\ldots\ldots + x_N[k]]$

APPENDIX

QUANTIZATION TECHNIQUE

In the paper [13], the author used the quantization method as like below:

$q[x] = \lfloor x + 1/2 \rfloor$

But in Gossip algorithm, 16 bit quantization has been used.

Quantization bit=n

Quantization level= $L = 2^n$

Partition values=b_l

Partition region for l , $B_l = [b_(l-1), b_l)$

Reconstruction values: g_l

Quantization Index : $Q_i(f) = l, iff \in B_l$

Quantization value: $Q_i(f) = g_l, iff \in B_l$

For n=16 and bound [0,6]

Quantization level = $L = 2^16 = 65536$

Partition=6/65536 = 0.0000915527

Non- Quantized value x=3.6

Quantized value of x: q[x]=3.6

But in Quantization method $q[x] = \lfloor x + 1/2 \rfloor$

$q[3.6] = \lfloor 3.6 + 1/2 \rfloor$=$\lfloor 4.1 \rfloor = 4$

BIBLIOGRAPHY

[1] Konstantin Avrachenkov, Mahmoud El Chamie, and Giovanni Neglia, *A local average consensus algorithm for wireless sensor networks*, INRIA Sophia Antipolis, Mediterrance (2004).

[2] T. C. Aysal, M. J. Coates, and M. G. Rabba, *Rate of distributed average consensus with dithered quantization*, Signal Processing, IEEE Transactions on 4905 - 4918 (2008).

[3] T. C. Aysal, M. J. Coates, and M. G. Rabbat, *Distributed average consensus with dithered quantization*, Signal Processing, IEEE Transactions on 4905 - 4918 (2008).

[4] Tuncer C. Aysal, Mark Coates, and Michael Rabbat, *Distributed average consensus using probabilistic quantization*, IEEE/SP 14th Workshop on (2007), 640644 ,26–29.

[5] Tuncer Can Aysal, Mark J. Coates, and Michael G. Rabbat, *Distributed average consensus with dithered quantization*, IEEE transaction on signal processing 56 (2008).

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, *Randomized gossip algorithm*, IEEE Transactions on Inf ormation Theory, 52(6):2508 2530y (2006).

[7] Ruggero Carli, *Topics on the average consensus problems*, University of Padova (2008).

[8] Ruggero Carli, Fabio Fagnani, Paolo Frasca, Thomas Taylor, and Sandro Zampieri, *Average consensus on networks with transmission noise or quantization*, In ECC 07, Kos (2007), pages 18521857.

[9] Andrea Censi and Richard M. Murray, *Real-valued average consensus over noisy quantized channels*, American Control Conference 1088-1092 (2009).

[10] F. Fagnani and S. Zampieri, *Randomized consensus algorithms over large scale networks*, In Proceed- ings Inf ormation Theory and Application Workshop (2007).

[11] J. A. Fax and R. M. Murray, *Information ow and cooperative control of vehicle formations*, IEEE Trans. on Automatic Control 49, pp. 1465–1476 (2004).

[12] Paolo Frasca, Ruggero Carli, Fabio Fagnani, and Sandro Zampieri, *Average consensus on networks with quantized communication*.

[13] ———, *Average consensus by gossip algorithom using quantized communication*, 47th IEEE Conference on Decision and Control (2008).

[14] Robert M. Gray and David L. Neuhoff, *Quantization*, IEEE Conference on Information Theory 44 (1998), no. 6.

[15] Soummya Kar and José M. F. Mourat, *Distributed consensus algorithms in sensor networks: Quantized data and random link failures*, Signal Processing, IEEE Transactions on 4905 - 4918 (2010).

[16] A. Kashyap, T. Basar, and R Srikant, *Quantized consensus*, 2007, p. 43:11921203.

[17] Deniel J. Klein, *Co-ordinated control and estimation for multi-agent system:theory and practice*, University of Washington (2008).

[18] Javad Lavaei and Richard M. Murray, *Quantized consensus by means of gossip algorithm*, Automatic Control, IEEE Transactions on 57 (2012).

[19] Tao Li, Minyue Fu, Lihua Xie, and Ji-Feng Zhang, *Distributed consensus of multi-agent systems with finite-level quantization*, Proceedings of the 7th Asian Control Conference (2009).

[20] N. A. Lynch, *Distributed algorithms*, Morgan Kaufmann Publishers, Inc. (1996).

[21] MHdegroot, *Consensus and cooperation in networked multi-agent system*, Invited Paper.

[22] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray, *Reaching a consensus*, J. Am. Statist Assoc. (1974).

[23] A. Olshevsky, Ozdaglar A., and Tsitsiklis J.N., *On distributed averaging algorithms and quantization effects*, Conference Publications (9-11 Dec. 2008).

[24] Sandip Ro, Yan Wan, Ali Saberi, and Mengran Xue, *Designing linear distributed algorithms with memory for fast convergence*, International journal of robust and non-linear control 57 (2012).

[25] G. Royle and C. Godsil, *Algebraic graph theory*, Springer Graduate Texts in Mathematics 207 (2001).

[26] Reza Olfati Saber, J. Alex Fax, and Richard M. Muray, *consensus and co-operation in netwoked multi-agent systems*, Invited paper.

[27] Stefania Sardellitti, Massimiliano Giona, and Sergio Barbarossa, *Fast distributed average consensus algorithms based on advection-diffusion processes*, IEEE transactions on signal processing 58 (2010), no. 2, 891–900.

[28] Yan Wan, Kamesh Namuduri, Swathik Akula, and Murali Varanasi, *The impact of multi-group multi-layer network structure on the performance of distributed consensus building strategies*, International journal of Robust and non-linear control (2012).

[29] L. Xiao and S. Boyd, *Fast linear iterations for distributed averaging*, IEEE Conference on Decision and Control 58 (2003), no. 2.

[30] L Xiao, S Boyd, and S J Kim, *Distributed average consensus with least-mean-square deviation*, Journal of Parallel and Distributed Computing (2006).

[31] L. Xiao, S. Boyd, and S. Lall, *A scheme for robust distributed sensor fusion based on average consensus*, In IPSN 05, Los Angeles (2005).

[32] Lin Xiao, Stephen Boyd, and Seung-Jean Kim, *Distributed average consensus with least-mean-square deviation*, Stanford.

[33] Mehmet E. Yildiz and Anna Scaglione, *Differential nested lattice encoding for consensus problems*, Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on (2007).

[34] Deming Yuan, Shengyuan Xu, and Huanyu Zhao, *Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms*, IEEE Transaction on system 41 (2011), no. 6.