NETWORK CODED PARALLEL NETWORKS AND THEIR APPLICATIONS

By

NITIN RAKESH

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING

UNDER THE SUPERVISION OF
DR. VIPIN TYAGI

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NOVEMBER, 2012

DECLARATION

I hereby certify that the work reported in the Ph.D. thesis entitled "**Network Coded Parallel Networks and Their Applications**" submitted to Department of Computer Science Engineering and Information Technology, **Jaypee University of Information Technology (JUIT), Waknaghat** is an authentic record of my own work carried out under the supervision of **Dr. Vipin Tyagi**, Associate Professor, JUET, Guna. The work in this thesis is my original investigation and has not been submitted elsewhere for the award of any other degree or diploma.

Nitin Rakesh
Department of Computer Science and Engineering
Jaypee University of Information Technology,
Waknaghat, India
Dated: 6th November, 2012

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "**Network Coded Parallel Networks and Their Applications**", submitted by **Nitin Rakesh** submitted to Department of Computer Science Engineering and Information Technology at **Jaypee University of Information Technology, Waknaghat, India,** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Dr. Vipin Tyagi
Department of Computer Science and Engineering,
Jaypee University of Engineering and Technology,
Guna, M.P., India
Dated: ………..

ABSTRACT

This study asserts that parallel network coding is a new communication paradigm that takes advantage of the broadcast using the characteristic of network coding for parallel architectures. Network coding has recently developed as an innovative paradigm for optimization problems of high-scale computation between several nodes of parallel architectures. This study has proposed broadcasting problems in parallel networks using network coding approach to make this communication more efficient. We have proposed a decentralized approach by optimizing parallel communication with the use of network coding to overcome the problem of high communicational and computational time complexity in parallel architectures. To reduce the complexity of such communication, Linear Network Coding (LNC) is implemented in the parallel environment. Parallel architectures involve parallel communication with the aim of receiving the complete information fast with higher information rate at each node. To reduce these complexities of parallel communication author has considered Multi-Mesh of Trees (MMT) architecture for parallel communication. Linear Network Coding (LNC) has been implemented on MMT by proposing Linear-Code Multicast with Parallel Algorithms (LCM-PA) to achieve max-flow from the source to each receiving nodes. We have minimized the communication steps and time complexity involved in transfer of data from one processor to the other processor during parallel communication while achieving the desired throughput in a multicast scenario. This work presents a novel distributed parallel network approach, which is enriched by LNC.

Furthermore, we investigated the combination of data buffering and network coding for improving the performance of parallel communication. We have proposed a data loss recovery approach described as Link Failure-Recovery Mechanism (PAC:LF-RM) for improving the performance of 2D-Mesh network during communication failures. PAC:LF-RM combines the coding opportunities with data buffering at the alternate level of network, which allow coding advantages to continue with data buffering. This approach minimizes the chance of data loss due to communication failure. The simulation results and performance evaluation factors, which shows that PAC:LF-RM can significantly improve performance of parallel communication and secure the data loss. The combination of network coding and node buffering for handling communication failure situation in the 2D-Mesh network is implemented for different cases of communication failures.

Energy (the information rate, node involvement per steps of communication, size of communicating data, storage at each node) involved in parallel architectures decides whether the network is efficient or not. We examined the problem of broadcasting information to all nodes in 2D-Mesh architecture by using the network coding such as when the communication increase, the rate of information also increases and the involvement of nodes during this communication is higher in comparison with the traditional approach of data communication in parallel networks. A novel approach to network coding in parallel communication is proposed, which reduces the storage requirement at each node and thus solves the problem of high data size to communicate in 2D-Mesh network. It is proved that this approach of failure detection and recovery is efficient in terms of data recovery time, connection reestablishment and complexity. The network coding approach and recovery process runs concurrently and it is found that this approach is effective for recovery process. Finally, the failure situations in network with contour approach of communication are presented.

LIST OF ACRONYMS

S. NO.	ACRONYM	FULL NAME
1.	LNC	Linear Network Coding
2.	XOR	Exclusive-OR
3.	PAC	Parallel Architecture Coding
4.	LF-RM	Link Failure-Recovery Mechanism
5.	MMT	Multi-Mesh of Trees
6.	LCM-PA	Linear-Code Multicast with Parallel Architectures
7.	MM	Multi-Mesh
8.	WA	Working Array
9.	AAB	All-to-All Broadcast
10.	CS	Communication Step
11.	P_ID	Processor Index
12.	INFO	Information
13.	RDT	Recursive Diagonal Torus
14.	MoT	Mesh of Trees
15.	R|S	Receiver|Sender
16.	PN	Parallel Network
17.	NC	Network Coding
18.	NEC	Network Error Correction
19.	CODEB	Coding-based Broadcast protocol for ad hoc networks
20.	P2P	Peer-to-Peer
21.	LRF	Local-Rarest-First
22.	ROCX	Routing with Opportunistically Coded Exchanges
23.	MORE	MAC-independent Opportunistic Routing protocol
24.	ANCC	Adaptive Network Coded Cooperation
25.	NB-JNCD	Non-Binary Joint Network-Channel Decoding

LIST OF SYMBOLS

S. NO.	SYMBOL	DESCRIPTION
1.	$đ_1$, $đ_2$	Data Bits
2.	$\check{N}$	Parallel Network used in Figure 3.1
3.	$Ᵽ_1$- $Ᵽ_7$	Nodes of $\check{N}$ network in Figure 3.1
4.	ζ_1-ζ_1	Coefficient
5.	n	Number of Nodes
6.	$WA_{1\text{-}8}$	Working Array of eight Processors
7.	P_n	Processor Index
8.	I_n	Information Associated with n^{th} Processor
9.	α, β	Block Index in MMT
10.	i, j	Node Index in MMT
11.	M	Network used in Figure 4.3
12.	ν	Vector Space
13.	φ	Collection of Non-Source Nodes
14.	Ω	d-dimensional Vector Space
15.	$\oplus$	Exclusive-OR
16.	y	Information Field
17.	g	Encoding Vector
18.	x	Data Symbol
19.	T_{nc}	Total number of Transmission with NC
20.	T_w	Total number of Transmission without NC
21.	$\mathscr{g}$	Network Graph in Figure 6.6
22.	∂_1	Data of Node 1 in Figure 6.6
23.	$\mathfrak{D}$	Degree of communication
24.	$\mathscr{b}$	Buffering levels
25.	$\mathcal{N}\mathscr{b}$	Number of node Buffered
26.	$\mathcal{T}$	Time to encode the data

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am extremely indebted to my supervisor, Dr. Vipin Tyagi for unswerving supervision in this research work. I am heartily thankful to my GOD for blessing me with the opportunity to accomplish my work under such a knowledgeable personality. I have learnt several things from him while pursuing my research work. He is a very nice Human Being and a very strong Individuality. He gave me every possible solution which can help me in exploring new dimension in my research. I am very fortunate to become closer to my supervisor and friendly getting the knowledge and encouragement from him.

I like to devote my very special thanks particularly to Dr. Yajulu Medury, The Chief Operating Officer, Jaypee Education System who guided and supported me during copious important phases and decisions. I am grateful to Prof. Ravi Prakash, Honorable Vice-Chancellor and Brigadier (Retired) Balbir Singh, Director, Jaypee University of Information Technology, Waknaghat, for providing admirable research atmosphere and amenities for my research. I am also thankful to Prof. S. P. Ghrera, Head of Computer Science & Engineering and Information & Communication Technology Department, Jaypee University of Information Technology, Waknaghat, for his support during this research. I am incredibly thankful to IT BHU for providing support of National Super Computing facility GARUDA, to validate my results.

I am ceaselessly appreciative to my respected grandparents, my Mother and Father and affectionate younger Brother. Furthermore, I am gratified beyond words for my most devoted Wife and Charming Daughter for their love and endless support. Without their continual moral support, my work is impossible.

I strongly commit and devote this accomplishment to Anant Shri Vibhushit Mahamandleshvar Shri Swami Gurusharnanand Ji Maharaj, who is providing his Holiness and Strength for my encouragement and have blessed me with his presence.

TABLE OF CONTENTS

CONTENT PAGE NO.

CHAPTER 1

CHAPTER 2

CHAPTER 3

EFFICIENT BROADCASTING IN PARALLEL NETWORKS USING NETWORK CODING …………………………………………………….. 21–34

CHAPTER 4

LINEAR-CODE MULTICAST ON PARALLEL ARCHITECTURES (LCM−PA) ………………………………………………………… 35–62

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 1

INTRODUCTION

1.1 *Problem Statement*

The application of network coding with parallel networks is an innovative application area. Communication in parallel networks is always in research for achieving low storage and communication cost. It is interesting to recognize that expansions within parallel networks are for achieving efficient communication and developing the practicable prospects of such networks [1, 2]. The communication in such networks thus revolves into higher communication and computational cost. Further, during parallel communication each node is either computing or communicating with other nodes. So during communication failure, reason for this failure and recovering data loss is unfeasible. It is required to resolve these problems for energy-efficient broadcasting.

The aim to implement algorithms on these networks is to achieve efficient communication and computation cost. The literature review reveals that active research on parallel networks development overcomes the issues of communication and computation cost on several networks. However, tradeoffs between networks still have disadvantages:

- *High information size in communication between nodes.*
- *Exponential increase in communication and computation cost.*
- *Performance is inadequate.*
- *It does not provide failure recovery.*
- *Possibilities of data loss in communication.*
- *No generic communication approach.*
- *Provision for handling communication failure is not present.*

We studied these disadvantages of parallel communication and resolved them. Advantages of proposed system are:

- *Limits the increase in communicating information size.*
- *Limits the increase in communication and computation cost.*
- *Performance is acceptable.*
- *Failure recovery approach with further communication advantage.*
- *Overcome the possibilities of data loss with provision of failure repair.*
- *Contour approach for generic communication.*
- *Communication failure recovery with generic approach.*

We present an approach for multisource multicast in parallel networks by which faulty nodes, information size and communication complexity decreases with code length. This approach also achieves capacity asymptotically as given by max-flow min-cut bound of [1, 2]. Our analysis uses insights from network coding, which lead to an innovative application in parallel architectures. We have also given a generic coding method for different parallel networks, and showed how network coding affect the crisis issues of parallel communication. Taking different network set-ups, we have shown that linear network coding effectively reduces the chance of errors. Finally, we have proved that network coding is useful over routing approaches. The study suggests the nature and robustness of network coding can offer a significant advantage in feasibility of parallel architectures. We have also solved the problem of reliability in parallel communication and propose efficient approach for communication in such networks.

The robustness of a network depends on the way it manages the failures. In networks with huge data communication the chances of failures are high. Consequence of failure in parallel communication results in data loss, incomplete information delivery, communication errors, and high data recovery complexity. These problems require effective and robust solutions. Further it is required to have generic solution to the issues of the connection failure and minimum energy amount required to recover the data loss. This study considered the issues of data loss because of communication failure reasons and performed experiments for various failures. Later, we proposed an approach to handle the failures and recover data loss. The Contour approach proposed in this work has following main advantages:

- *This approach reduces the data size at each communication step by performing XOR operation on the data set.*
- *Reduced data size.*

- *The involvement of nodes in communication increases (processor utilization).*
- *The rate of information transfer also increases.*
- *The present applicability of Network Coded Contour approach makes the parallel architectures more practical.*
- *Leads in developing an efficient communication approach for parallel architecture.*

1.2 *Contributions*

We have proposed a novel application of Network Coding [1, 2] with parallel communication We have studied and resolved the tribulations of parallel networks and optimized the utilization of processing unit. We have proposed a standard approach for these networks. We proved successively that applying this approach exponentially decreases the effect of faulty nodes, information size and communication complexity. Besides we considered the problem of parallel communication failure and examined based on proposed Contour approach of communication in parallel networks and network coding.

Further we studied several important contributions to the problem of parallel communication in various issues. This study gives many productive and important results for the problematic issues of parallel communication. We brief important contribution of the thesis in order of event in this section. At first, this thesis presents the problem of information size, faulty nodes and algorithmic complexity involved in parallel communication.

- LINEAR-CODE MULTICAST ON PARALLEL ARCHITECTURES:
  This research sets up Linear Network Coding (LNC) in the parallel environment. We considered some parallel architecture for proof and examined the results in a generic environment. This thesis develops an approach to remove the problems of parallel networks.

Energy involved in parallel architectures, i.e., the information rate, node involvement at each of communication, size of communicating data, storage at each node decides the efficiency of the architecture.

- XOR IN PARALLEL ARCHITECTURES:
  In this work we examine the problem of broadcasting information to all nodes in 2D-Mesh architecture. We performed comparative analysis using network coding with traditional approaches of data communication. This resulted in increased rate of information and the involvement of nodes is higher. This research work proposes an approach of network coding in parallel communication, which reduces the storage requirement at each node and thus solves the problem of high data size to communicate in 2D-Mesh network.

Parallel network coding is a new communication model that takes advantage of the broadcast using characteristic of network coding for parallel architecture. Network coding has recently developed as an innovative model for optimization problems for high-scale computation between several nodes of these architectures.

- PARALLEL ARCHITECTURE CODING: LINK FAILURE-RECOVERY MECHANISM:
  In this work we evaluated chances of communication failure and proposed an efficient solution for such evolving circumstances. Communication failures are unavoidable and results in data loss. We proposed an approach which overcomes the data loss because of these failures. Using combination of network coding and buffering at an alternate degree of network nodes, this stipulation reduces. In this thesis, we researched this combination of network coding and node buffering for handling communication failures in the 2D-Mesh network. We presented analogous results for different cases of communication failures.

Finally, this thesis explains that with some comparative study on traditional approach and the proposed approach of data recovery, we gain a lower communication cost for failure using our approach.

- FAILURE DETECTION USING CONTOUR APPROACH ON NETWORK CODED PARALLEL NETWORKS:

  This thesis studies the recovery from communication or node failures in network coded parallel architectures using contour approach. We proposed a reciprocally efficient approach to recovery from failed communication in parallel network as Parallel Architecture Coding- Link Failure-Recovery Mechanism (PAC: LF-RM). Besides, this thesis also propose a broadcast format and design a linear network coded parallel networks using contour approach to communicate in such networks. We showed the advantages of the proposed approach by comparing this approach with other traditional recovery mechanisms.

1.3 *Outline of the Thesis*

The thesis is organized in seven chapters. CHAPTER 1 presents the Introduction of the research work. This introductory section discusses the research problems and describes the contribution of the author. CHAPTER 2 presents the background for the research in this thesis. This chapter states introduction of network coding principle and its applications. CHAPTER 3 presents efficient broadcasting in parallel networks using network coding approach. This chapter briefs the result for the work proposed in next chapter. CHAPTER 4 presents generic approach of network coding implementation over several parallel networks. This chapter presents an innovative application in parallel architectures. CHAPTER 5 presents the solution to the problem of high data size to communicate using 2D-Mesh network. CHAPTER 6 presents the combination of network coding and node buffering for handling communication failure in the 2D-Mesh network. CHAPTER 7 presents the failure detection using contour approach on network coded parallel networks. Finally, CHAPTER 8 concludes this thesis and presents the experimental results and simulations. Further this chapter presents the future extensions (scope) of research in this field.

CHAPTER 2

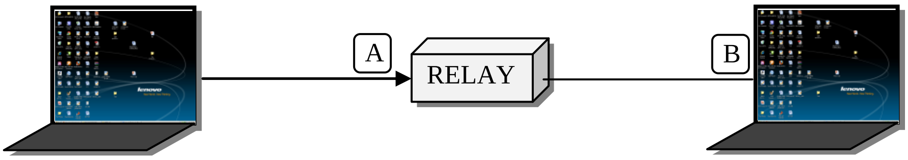
BACKGROUND

Communication means sending data from one to another node within a network. It may involve several other intermediate nodes, which collects data and transfers it to other connected node. The information communicates in a store and forward manner. Network coding has advantages over the traditional store and forward technique of data communication. It increases throughput and capacity of a network. Ahlswede *et al*. in 2000 proposed network coding technique. This technique studied comprehensively for solving various computer networking problems. Network coding provides improved solutions to several problems of computer networks and its diverse applications. In this chapter we give a brief description of network coding and its applications. Further, we highlight the problems in parallel networks and possible proposed solutions to these problems. This chapter discusses various fields and problems where network coding principle applies.

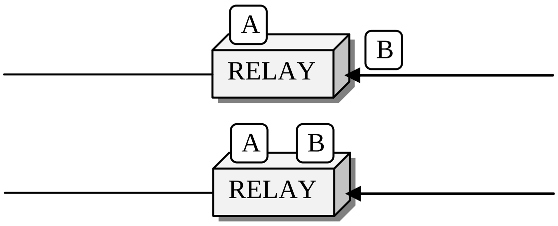
2.1 *Network Coding*

Routing problems always open possibilities to improve communication complexity and throughput. The traditional technique used to communicate in a network is by store and forward approach. Research in this field shows that network coding offers better manner to communicate same information in comparison with traditional approach in a network. Network coding allows the intermediate node to encode the received information which increases the maximum data transfer rate. In conventional store and forward technique, only the source node is responsible to encode the data.

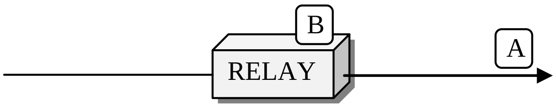
All intermediate nodes send the information to other connected nodes. This advantage of network coding leads to a revolution in the research of information theory and network communication. Let us understand the advantage of network coding over store and forward method with the help of an example.



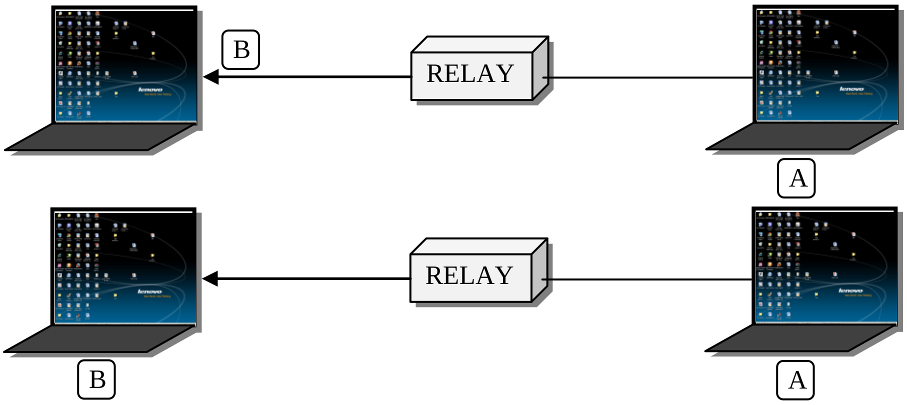
Transmission 1: Node A sends its data to intermediate node.



Transmission 2: Node B sends its data to intermediate node.



Transmission 3: Intermediate node sends information of Node A to Node B.

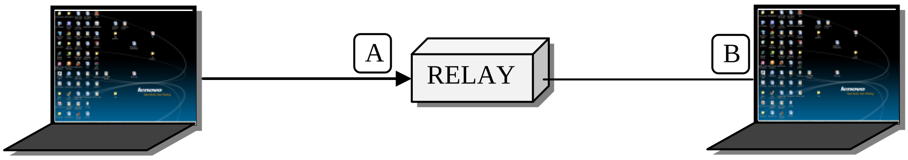


Transmission 4: Intermediate node sends information of Node B to Node A.

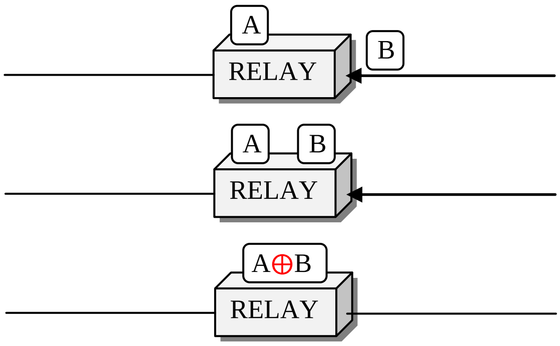
Figure 2.1: Transmission using Store and Forward technique.

In figure 1, two nodes (A and B) communicate with each other using store and forward method. Node A sends data to the intermediate node and then node B sends data to same intermediate node. This data collection at the intermediate node consumes two transmissions.

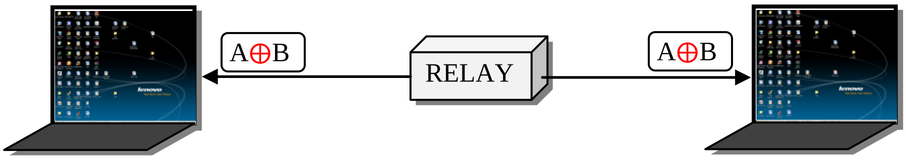
Further, the intermediate node sends data of A to node B which will take one transmission. Similarly, this intermediate node will transfer data of B to node A in another transmission. This shows that using store and forward method, the data exchanges between node A and B in four transmissions. While using network coding, the data exchanges in three transmissions between node A and B. Figure 2 shows the advantage of network coding over store and forward technique as it reduces the number of transmissions from four to three.



Transmission 1: Node A sends its data to intermediate node.



Transmission 2: Node B sends its data to intermediate node.



Transmission 3: Intermediate node sends information A⊕B to Node A and Node B.

Figure 2.2: Transmission using Network Coding technique.

Network coding also provides advantage to increase the capacity of the network. Using the common example of butterfly network (figure 3) we have shown this advantage. Node P_1 and P_2 are the sender node which send $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ to node P_5 and P_6. While in this communication, node P_3 receives data from both node P_1 and P_2. Now, network coding encode (XORs) this received information and communicates this information ($\begin{bmatrix}1\\1\end{bmatrix}$) to P_4. Thus in this example, network coding reduces the transmissions and increases the capacity of the network. Further this chapter explains the encoding and decoding operations in network coding techniques.

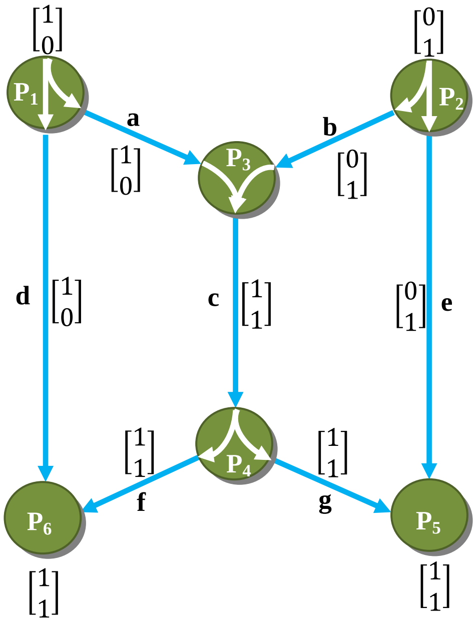


Figure 2.3: Network coding technique on Butterfly Network.

Performance of network is directly based on the flow of data with respect to algorithm. A node without input is called *source node* and a nodes with one or more inputs is called *non-source node*. The set of non-source nodes contains the destination node. Now by coding nodes it receives multiple data at a unit of time and results in combined encoded value from these nodes. This encoded value is combined in one data unit and the receiving nodes decode the information to receive the data. Thus, network coding has four major advantages: on the throughput, complexity, robustness and security. Network coding provides efficient methods for several applications and remedy for many problems:

2.1.1 *Throughput*

Throughput of a communication network is successful delivery rate of a message over a network. One of the major benefits of network coding approach is throughput [2, 3]. Ahlswede *et al*. showed that network coding provides maximum achievable flow in a network. Noguchi *et al*. presented the benefits of network coding for high and effective network utilization. Thus, throughput is the most important utility of network coding. This approach provides increase in throughput by reducing the number of transmissions [1, 2]. Figure 2.1 and 2.2 shows an example which compares network coding and store and forward technique.

2.1.2 *Complexity*

In several applications network coding has proved that it provides more optimal solution to complex problems [3, 5]. Besides, network coding improves performance by removing or reducing the practical limitations associated with suboptimal solutions [4]. This approach reduces the communication and computation complexities.

2.1.3 *Robustness*

Network coding is an optimal solution for robustness in several applications which inherits the encoding scheme for data communication in network nodes [6-9]. Network coding provides robustness against the packet loss and link failure in the network. Several solutions are in literature which provide robustness for packet loss in a network, even network coding is most significant solution to this problem. Similarly for link failure (which may be because of several reasons) network coding is efficient mechanism.

2.1.4 *Security*

The concepts of information theory, like entropy, depend on the communication methods used in a network. Security issues are based on strength of the adversary during and after communication. These problems are addressed by deploying network coding in communication. As security is one of the advantages which network coding provides, it still lacks in some security issues.

The principle of network coding is implemented with network routers and switches [3, 4]. This advantage of network coding is useful with various applications. This chapter discusses the applications and advantages of network coding:

2.2 *Applications*

The network coding applications are almost in diverse and several fields. The foremost applications are as following:

1: Distributed Storage System [5-8]
2: Content Distribution [9-14]
3: Layered Multicast [15-20]
4: Throughput Enhancement [21-35]
5: Flooding (Broadcast Storm Problem) [36-45]
6: Network Error Correction Code [46-57]
7: Erasure Correction Code [58-74]
8: Loss Tomography [75-83]
9: Topology Inference [84-86]
10: Pollution Attacks [59, 87-94]
11: Eavesdropping [59, 95-99]

2.2.1 *Distributed Storage System*

The operation performed dynamically need data storage during communication. However, it is difficult to have everything on one disk as disk failure results in losing entire information. So, Dimakis *et al*. provided a solution to this problem by spreading data on several nodes with redundancy [5]. Several applications follow the distribution storage system. [5] describes an approach that can repair failures in encoded systems by deciding the information to repair this failure. Dimakis *et al*. have achieved reduced repair bandwidth at the cost of high storage. Wu *et al*. further characterized tradeoffs between storage and repair bandwidth using a cut based approach [6]. They also presented the approach an algebraic path-weaving technique to prove the existence of codes which achieves the optimal tradeoff. Wu presented in [7] the techniques for constructing network codes. Use of these codes achieves optimal tradeoff between storage efficiency and repair network bandwidth. The size of these codes depends only on the number of nodes at an instant of communication, whereas it is independent of number of failures or repairs. Dimakis *et al*. have proposed a survey on network codes for distributed storage [8]. In this survey, they presented a review based on erasure coding approach in distributed storage systems to solve the problem of reducing repair traffic. They analyzed the previous results based on various models of erasure coding. These researches evolved network coding as an effective solution for high storage and failure scenarios.

2.2.2 *Content Distribution*

Content distribution over network is a huge problem of reliability, security, robustness and resource optimization in several applications. Cohen presented BitTorrent file distribution system which search for Pareto efficiency using tit-for-tat method [9]. Zhu *et* al. have proposed multicasting in application-layer overlay networks using network coding [10]. This research considerably improves end-to-end throughput of multicast by considering advantages of the overlay network nodes (which are capable beyond basic operations of store and forward method) and the overlay topology. They used network coding in the overlay network nodes to encode and decode data at the message level. However, advantage of the overlay topology to make a 2-redundant multicast graph. Further, Gkantsidis *et* al. proposed that content distribution of large files over a network (see figure 2.4) is efficient by using network coding principles [11]. Large files distributed using this approach in a dynamic environment. This distribution based on local information without any centralized knowledge of network topology. This makes scheduling large-scale content propagation easier. The network coding approach is also useful as heterogeneous networks do not have any knowledge about the arrival and departure of network nodes. Further, network coding restricts the attack of malicious nodes for content distribution [100, 101].

Chiu *et al*. studied the use of network coding in P2P (Peer-to-Peer) networks [12]. Based on a simple star network, the maximum achievable throughput is analyzed comparatively between network coded and traditional approach of routing in P2P networks. This work states that, there is no coding advantage using a simple model for P2P content distribution networks. Further Zhang *et al*. have proposed an approach for peers which are seeking maximization of individual payoff [13]. This distinguishes a coding based P2P content distribution. This work characterizes the traditional P2P approach as a special case of network coding. Further, it is shown that market flexibility to impatient agents is improved using network coding. Ma *et al*. proposed another network coding approach for P2P content distribution [14]. This approach verifies the dependencies in blocks before transmitting. They also implemented two other systems: encode and LRF (Local-Rarest-First) and experimentally proved the significance of the approach over traditional approaches.

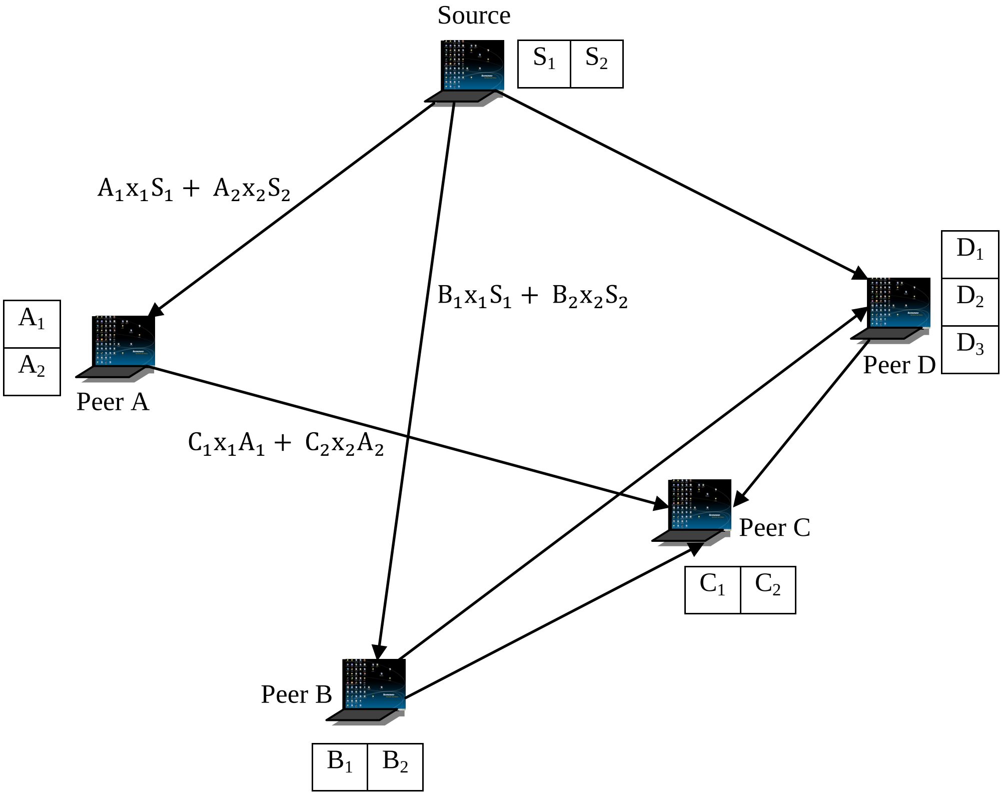


Figure 2.4: Content Distribution using Network Coding.

2.2.3 *Layered Multicast*

Multicast is the method to communicate data from one source to multiple destinations. While based on the receivers' capacity, the layering of each receiver enables flexibility in the process. Network coding provides better throughput for layered multicast and increases the possibility of achieving throughput based on the progression of large size. Cui *et al.* solved the problem of dynamic media distribution problem using a P2P streaming solution [15]. Zhao *et al.* have proposed a solution to improve the throughput of an overlay multicast session [16]. Using cache-and-relay and layer-encoded streaming techniques the solution of asynchrony of user requests and heterogeneity of peer network bandwidth are resolved in this work. Chenguang *et al.* proposed multirate streaming for multimedia data in directed networks using network coding [17]. [18] provide solution for computing the optimal size of each receiver layer. Further in [19] Dumitrescu *et al.* proposed layered multicast with inter-layer network coding. This proposed work utilizes the full potential of network coding by multicasting in different data layers. Thus the throughput increases due to flexibility in optimizing the data flow. Applications like video streaming requires multicast at different rates to different receivers [20]. To enable multi-resolution multicast Kim *et al*. proposed two-stage message passing algorithm which generates network codes for single-source multicast. This work focuses on maximizing the receiving of total layers by all receivers.

2.2.4 *Throughput Enhancement*

Considering figure 2.1 and 2.2, using network coding, in three transmission steps the data exchanges between node A and B. Therefore, in multi-hop network, this approach increases the network throughput and improves the possibility of energy consumption. It is necessary to use the capacity of a wired or wireless network to implement enhancement approaches with maximum opportunities [21]. Ni *et al.* studied these opportunities of network coding in wireless mesh network [22]. They consider that using network coding what is performance gain while routing in wireless mesh network. COPE analyzes the Routing with Opportunistically Coded Exchanges (ROCX) approach resulting in reduced number of routing transmissions. Further for theoretical perspective, Sengupta *et al*. analyzed COPE-type network coding in wireless networks for throughput improvements [23]. Chachulski *et al*. utilizes the advantages of opportunistic routing to further improve network throughput [24]. They proposed MORE (MAC-independent opportunistic routing protocol) approach. MORE enables random mixing of packets before transmitting to ensure the routers to transfer different packets. This approach increases the throughput and robustness in transmissions. While Chaporkar *et al.* suggested adaptive network coding and scheduling for maximizing throughput in wireless networks [25]. Sagduyu *et al.* studied network coding advantages over wired networks. The proposed model develops network codes by working with MAC schedulers [26]. [27-35] further studied techniques to improve the network throughput and provide better routing methods.

2.2.5 *Flooding: Broadcast Storm Problem*

Flooding is an advantage to broadcast by which the opportunity to network receiving nodes increases. But the basic broadcasting with flooding creates problem of redundancy, collision etc. and increases the broadcasting cost. This problem is known as Broadcast Storm Problem [36]. Ni *et al.* identified this problem and analyzed and simulated the results. Further Ni *et al.* proposed several methods to overcome the problem of redundancy in broadcasting and timing to broadcast. Peng *et al.* studied this broadcast storm problem. They proposed an approach to avoid redundant broadcasts by utilizing the topological and statistical information [37]. Several approaches addressed this problem and have proposed various approaches to resolve it. Network coding enables the flooding advantages for its maximum utilization [38-45]. Network coding-based broadcast [102-104] is the solution to the problem of several data packets with several nodes i.e., flooding. Implementation of network coding with flooding reduces the traffic size and opportunities of receiving nodes are still many. Network coding in flooding is also a solution to the broadcast storm problem [105-108].

2.2.6 *Network Error and Erasure Correction Code*

The traditional error correction codes are not efficient for transmission redundancy. Cai and Yeung have proposed Network Error Correction (NEC) [47, 48] to recover the last packets using network coding. NEC recovers the loss of data packets and the number of links. Besides, Ho *et al*. have proposed two recovery schemes. First one is receiver based recovery scheme and second is the network wide recovery scheme. Network coding has the utility of spatial redundancy which is used by network erasure correction code while recovering the last packets. Yang *et al*. proposed an algorithm which constructs network codes that achieves Singleton bound [51]. Further, Matsumoto *et al.* also proposed an algorithm for constructing linear network error-correcting codes. They defined the association of robust network coding with the network error-correcting codes [52]. Zhang defined the concept of the minimum distance of a network error-correction code. Using two proposed decoding algorithms the performance on MDS code is analyzed [53]. Bao *et al.* presented Adaptive Network Coded Cooperation (ANCC) for wireless relay networks [54]. Further Koetter *et al.* have presented the approach for error in random networks and proposed the method for error control in random network coding [55]. Supplementary to this approach, Silva *et al.* proposed an approach to control error in random network coding. They have constructed the codes using Rank-Metric approach. The tools developed for this approach is directly applicable to random network coding [56]. Guo *et al.* developed Non-Binary Joint Network-Channel Decoding (NB-JNCD) approach [57] as a reliable scheme for wireless communication. They compared this approach with other traditional approaches for comparative study. As an erasure correction code Koetter *et al.* have proposed Algebraic Approach to Network Coding [58]. They examined the issues of network capacity in an algebraic framework. Several other erasure correction codes mechanisms are proposed with the aim to reduce error in network and increase network capacity [59-74].

2.2.7 *Loss Tomography*

Network based tomography is the characteristic of the network to send and receive data packets from network edges. Similarly, loss tomography is the characteristic of rate of link loss in the network. Network coding provides advantage in estimating the rate of link loss in the network, identifying these links, tradeoff between bandwidth efficiency and estimated accuracy. The use of distributed network codes infers the link failures location and loss in a network [75]. Using network coding network monitoring for such conditions and loss tomography are further studied for active probing method, coding schemes and loss rate [76-78]. Factor graph approaches are used to visualize the factorization and understand large number of different algorithms. This work developed the understanding of several algorithms by factorizing in graphical form [79]. Mao *et al.* used this approach of factor graph to visualize the problem of network monitoring for link failures and losses [80]. They considered this approach in wireless sensor networks for monitoring the link loss monitoring. Lin and Gui *et al.* infer the network monitoring issues using network coding [82, 83].

2.2.8 *Topology Inference*

In networks with multiple sources and receivers, the communication is iterative and from the network edges. Network coding with topology inference utilizes the network bandwidth and gains the network throughput. Network coding enables reduced transmissions with robust communication with topology inference. Thus this advantage of network coding provides robustness in network communication. Fragouli *et al.* presented this approach in [84]. Further this approach combines tomographic techniques with network coding [85]. The network coded topological information exactly distinguishes between 2-by-2 subnetwork components. Then the topology is generated and this information is used to merge subnetworks. Yao *et al.* studied network tomography using network coding for failures. They classified the result as topology estimation and failure localization [86].

2.2.9 *Pollution Attack*

If in a network, some of the routing nodes are malicious, then these nodes can communicate false information of any combination and further combine with other non-malicious network nodes and soon this network will become polluted. Such attacks need prevention at early stage. If polluted information is injected in the network then at next hop this information is detected and prevented from creating pollution. Jaggi *et al.* firstly introduced approach of distributed polynomial-time rate-optimal network codes which works with Byzantine nodes. The algorithms presented targets the various attacking capability of the enemies [87, 94]. In [88], using random network coding in multicast networks Byzantine detection is performed. Gkantsidis *et al.* proposed the security issues for file distribution in a network using network coding [89]. Several diverse solutions are proposed for the pollution attack on network coding. Another significant approach is signature-based scheme to provide secure network coding. [90-92] have utilized the significance of signature based scheme to resolve the problems of pollution attack. Kehdi *et al.* proposed Null Keys (a security algorithm) which utilized properties of random linear network coding to detect malicious attacks [93].

2.2.10 *Eavesdropping*

In a network coded network the packets are communicated from the source node to the receiver. Meantime, if eavesdropping of additional packets occurs, then these packets are incorporated in the network by network coding itself. So, a secure transmission and coding scheme is required to act against this problem. Cai *et al.* proposed a new model which integrates the concepts of network coding and information security [95]. Further, Feldman *et al.* showed that making a linear network code secure is equivalent to find a linear code with certain generalized distance properties [96]. Strongly secure linear network coding was proposed by Harada *et al* [97]. Bhattad and Silva *et al.* proposed secure linear network coding. Silva *et al.* have considered the weak security requirements of Bhattad *et al.* They propose an approach which is independent of network code [98, 99].

Lucani *et al.* [109] has extended the principle of random linear network coding in consideration of time duplexing channel. They also proposed a method to minimize completion time by finding the packets to be coded in a network. By this research the search time of the coded packets are reduced to an optimal performance. This research ensures the network performance estimating the completion time for packet erasure probability and the packets to be coded. Several problems in the field of network coding are analyzed and resolved by translating in graph theory problem. In [110] the author has converted the problem of linear network coding to a graph theory based problem. In this research Chou *et* *al*. models linear code by taking help of hypergraphs. Authors of this research proposed an algorithm for iterative refinement and in polynomial time this algorithm satisfies linear code constraints. This was the primary systematic approach which solves several network coding problems. Authors have examined several algorithms by converting the problem into graph theory problem and as a result the network bandwidth and computation time is minimized or saved.

Network coding is an advantage for broadcasting in terms of energy efficiency. Network coding improves a factor and this theoretical gain improves the performance. In [41] distributed algorithms are proposed for wireless ad-hoc network scenario. Using simulations it was justified that these proposed distributed algorithms are performing optimally. The proposed work [41] signifies that when network coding is used with wireless ad-hoc networks than the benefits will further increases. Thus network coding enables energy efficient benefits when implemented with wireless ad-hoc network environment. In [40] two algorithms are proposed: a simple XOR based coding algorithm, by which without waiting for more and more coded packets it performs decoding process which makes it NP hard; and a Reed-Solomon based coding algorithm, which is optimal coding algorithm with a limitation that a node will wait until receiving the exact number of coded packets. The first algorithm gives a again upto 45% in comparison to a non coding algorithm and the second algorithm gains upto 61% based on the simulation results. This research proposes a broadcast protocol for mobile ad-hoc networks with network coding i.e. CODEB (Coding-based Broadcast protocol for ad-hoc networks). CODEB consists of three techniques: opportunistic listening; forwarder selection and pruning; and opportunistic coding.

Another approach which considered efficient broadcasting in mobile ad-hoc networks is [45]. This approach considered efficiency problems in mobile ad-hoc networks and provide a solution using network coding and directional antennas. In this approach an additional reduction in the energy consumption while performing broadcast using network coding is achieved using directional antennas. So, this approach combines the advantages of both directional antennas and network coding to achieve efficient broadcast mechanism. An algorithm, efficient broadcast using network coding and directional antennas, was proposed in this research. This algorithm studies the performance variation in a static forwarding node selection. Comparative to traditional broadcast scheme, i.e. CDS-based broadcast, the proposed approach [45] have better performance.

Wireless mesh network is a prominent network in research due to several advantages provided by this network. Broadcasting in wireless mesh network is also an important issue to consider. In [44] using R-code, which is a broadcast protocol based on network coding some one-to-all broadcast scenarios are considered. The backbone of this approach is the minimum spanning tree by which the minimum broadcast overhead and delays. With the help of intra flow network coding an efficient protocol: R-code is presented in this research which reduces the number of transmissions and delay as 14% and 50%.

Other than broadcasting network coding is utilitarian for several security applications in communication. [63] is a prominent application example scenario in which network coding is used for protecting many-to-one wireless flows. Prior to network coding approach, several other approaches like (1+1) protection scheme, (1:N) protection schemes are used for protecting the survivability of many-to-one flows in wireless sensor networks or wireless mesh networks. The major drawbacks with these traditional approaches are the way these approach consume the network resources. [63] proposed a network coding based protection of many-to-one flow in wireless flows which gives advantages over the traditional approaches. In this approach some necessary and sufficient conditions are studied based on network coding. This approach affect the network performance and using the greedy algorithm the scheduling of the transmissions from the source node are proposed. In this approach a polynomial time algorithm is proposed which perform network coding using {0,1} coefficients.

Random linear coding is a variant of network coding and several applications are proposed which utilizes the advantages of this approach. In [70] random linear coding is utilized for unicast applications in disruption tolerant network. Application with the factor of buffer space required random linear coding approach to achieve high probability and minimized block delay. This approach is significantly efficient when the constraints like, appropriate token limit choice under bandwidth constraint and under nodal buffer constraint.

Network coding applicability is dependent on the advantages provided by it. Network monitoring for failure occurrence and security is another important issue. [75] utilizes theses advantages of network coding for monitoring networks during multicast networks. This approach shows that network coding provides robustness due to the robust distributed network codes which are utilized to hold information for situation like link failure and losses in the network. This application is an optimal example of network coding applications for failure situations in a network.

Furthermore, network coding applications in the field of security became more prominent. Network coding applicability for content distribution is described in section 2.2.1. [89] states that network coding application for the security issues in file distribution are effectively and efficiently resolved using this approach. [89] proposed and approach with efficient content distribution and robust protection against malicious blocks. The protection schemes in this approach are efficient based on the amount of effort from the attackers. Network coding is providing better throughput for content distribution. [58] is an extension in the previously proposed approach in which multicast networks are studied in terms of network capacity. In [58] the link failure problem is illustrated for network recovery on networks with delay and delay free networks. That means robustness of networking is ensures by this approach.

These application shows that network coding is one of the innovative and vast field with enormous research possibilities. In this thesis we have proposed an application of network coding in the field of parallel communication. Furthermore, this thesis examines applicability of network coding in the parallel communication applications. The subsequent chapters reveal the application of this principle with in the parallel network and are examined based on certain parameters.

CHAPTER 3

EFFICIENT BROADCASTING IN PARALLEL NETWORKS USING NETWORK CODING

This chapter describes the applicability of linear network coding on parallel architecture for multi-source finite acyclic network. Further it discloses the problem in which different messages in diverse time periods are broadcast and every non-source node in the network decodes and encodes the message based on further communication. It is shown how to minimize the communication steps and time complexity involved in transfer of data from node-to-node during parallel communication. Multi-Mesh of Trees (MMT) topology is used for implementing network coding. To evaluate results with and without network coding, All-to-All Broadcast algorithm is considered. For data communication at different step, the source and destination nodes changes according to the flow of data in the algorithm..

3.1 *Network Coding and Parallel Communication*

Parallel architectures involve parallel communication with the aim of receiving complete information at higher rate. To reduce the complexity of parallel communication we have considered Multi-Mesh of Trees (MMT) [115] architecture and implemented Linear Network Coding (LNC) [2]. We have proposed Linear-Code Multicast with Parallel Architectures (LCM-PA) which achieves max-flow from the source of each receiving nodes. While achieving the desired throughput in a multicast scenario, we have minimized the communication steps and time complexity. We have developed a novel application of LNC for parallel network communication.

The meaning of parallel algorithm is to reduce the time complexity of several problems to enable fast and efficient communication. Several interconnection networks have developed parallel processing approaches based on the mesh topology. We have presented an efficient application of LNC [110-114], for MMT parallel network. This approach asymptotically achieves the capacity of multicast networks with network coding [2]. We assume a parallel multisource multicast, which is possible with correlated sources on MMT architecture. We have used an approach in which all nodes other than the receiving nodes perform random linear mapping from inputs on outputs (see figure 1).

In each incoming transmissions from source, the destination has knowledge of overall linear combination of data set. This information updates at each coding node by applying the same linear mapping to the coefficient vectors as applied to the information signals. As an example, let the set of two bits ($đ_1, đ_2$) multicast in a directed parallel network (Ň). Figure 1 shows that from the source node $Þ_1$ (unique node, without any incoming at that instant of time) to node $Þ_2, Þ_3$ and $Þ_4, Þ_7$.

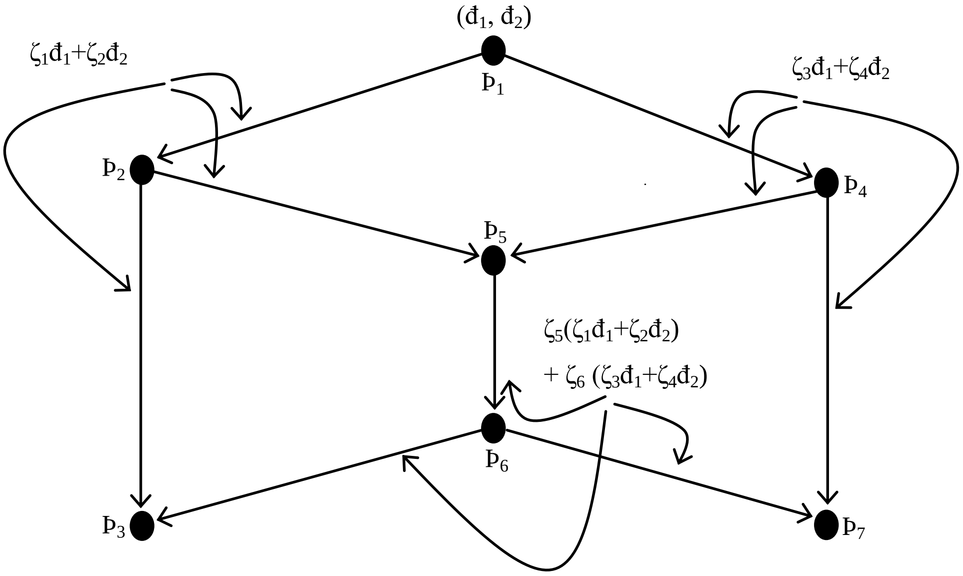


Figure 3.1: A network (Ň) used, as an example, to explain LNC with coefficient added at each data transfer from different nodes (the network has seven nodes $Þ_1, Þ_2... Þ_7$ and nine edges $Þ_1 Þ_2, Þ_1 Þ_4, Þ_2 Þ_5, Þ_4 Þ_5, Þ_2 Þ_3, Þ_5 Þ_6, Þ_4 Þ_5, Þ_6 Þ_3, Þ_6 Þ_7$ directed in this order). ($đ_1, đ_2$) is the set of data being multicast to destinations, and coefficients $ζ_1, ζ_2…ζ_6$ are randomly chosen elements of a finite field. Each link represents the data transmission.

Figure 3.1 shows information multicast in a network (Ň) with network coding. So, let us explain network coding using network (Ň) given in figure 1. Node $Þ_1$ aims to multicast the data set ($đ_1, đ_2$) to destination nodes $Þ_3$ and $Þ_7$. A randomly chosen coefficient of finite field clubs the data at each node and then transferred. The transmission continues as: ($đ_1, đ_2$) sent to both $Þ_2$ and $Þ_4$ with different coefficients ($ζ_1, ζ_2$ and $ζ_3, ζ_4$). Node $Þ_2$ receives $ζ_1đ_1+ζ_2đ_2$ and $Þ_4$ receives $ζ_3đ_1+ζ_4đ_2$. Then both $Þ_2$ and $Þ_4$ transfer the information to $Þ_5$. Node $Þ_5$ and $Þ_6$ sends $ζ_5(ζ_1đ_1+ζ_2đ_2)+ζ_6(ζ_3đ_1+ζ_4đ_2)$ to $Þ_3$ and $Þ_7$, which decodes to receive the data ($đ_1, đ_2$).

This approach shows that multicast of different data is possible using network coding in the network. It is right to say that flow of information and flow of physical commodities are two different things [2]. So to code information does not increase its content. To transmit information from the source to destination, capacity of a network can become higher [2]. We have implemented LNC on MMT and it results in novel efficient way of data communication by coding information to multicast on other processors. This approach has reduced the chance of error and increased the capacity to transmit data between processors. The linearity of code, for parallel transmission, makes encoding (at source end) and decoding (at receiving end) easy to implement in practice.

3.2 *AAB (All-to-All Broadcast) on Parallel Network*

We have implemented AAB on parallel network (MMT). For implementation, we have used AAB algorithm, which involves ten steps of algorithms to transfer and receive information [116]. We considered the MMT network with $n = 8$, where n is the number of processors and we considered $N = n^2 \times n^2, \forall\ n \in N$ and a $block = n \times n = row \times column$. MMT is better than other traditional parallel networks (e.g., Multi-Mesh (MM) [117, 118]) based on the topological properties of MMT comparable based on efficiency parameters. Figure 3.2 shows a comparison of these networks based on some parameters and figure 3.3 shows comparison between 2D Sort on MM and MMT for different number of processor. Table 1 [115, 119] shows characteristics of various processor organizations based on the network optimization parameters. This shows that MMT is more efficient then these network architectures.

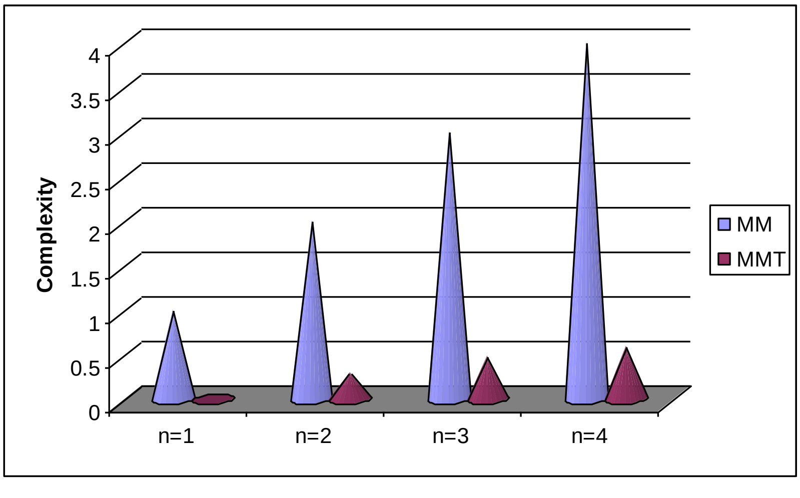


Figure 3.2: Comparison of MMT and MM on the basis of Communication links, Solution of Polynomial Equations, One to All and Row & Column Broadcast.

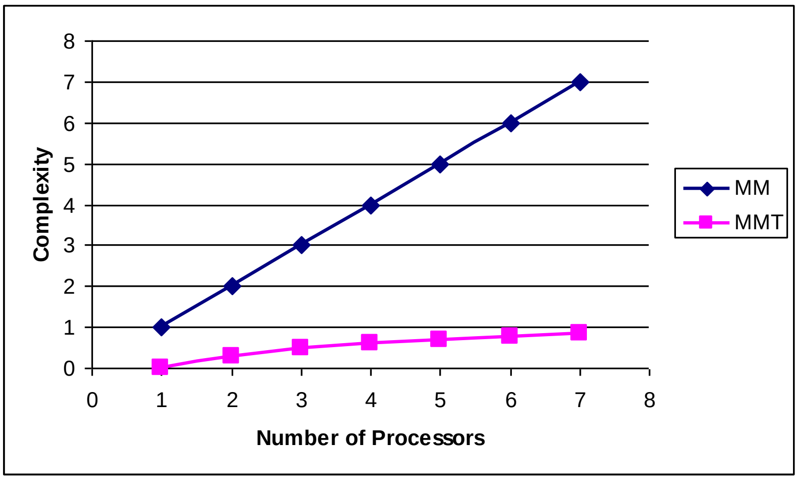


Figure 3.3: A comparison between 2D Sort on MM and MMT for different values of processor.

Table 3.1: Characteristics of Various Processor Organizations.

Network	Nodes	Diameter	Bisection Width	Constant Number of Edges	Constant Edge Length
1-D mesh	n	$n-1$	1	Yes	Yes
2-D mesh	n^2	$2(n-1)$	n	Yes	Yes
3-D mesh	n^3	$3(n-1)$	n^2	Yes	Yes
Binary tree	2^n-1	$2(n-1)$	1	Yes	No
4-ary hypertree	$2^n(2^n-1)$	$2n$	2^{n+1}	Yes	No
Pyramid	$(4n^2-1)/3$	$2\log n$	$2n$	Yes	No
Butterfly	$(n+1)2^n$	$2n$	2^n	Yes	No
Hypercube	2^n	n	2^{n-1}	No	No
Cube-connected cycles	$n2^n$	$2n$	2^{n-1}	Yes	No
Shuffle-exchange	2^n	$2n-1$	$\geq 2^{n-1}/n$	Yes	No
De Bruijn	2^n	n	$2^n/n$	Yes	No
MMT	n^4	$4\log n+2$	$2(n-1)$	Yes	No
MM	n^4	$2n$	$2(n-1)$	No	No

Now, to explain the algorithm, we consider $N = 8^2 \times 8^2 = 4096$ nodes, as the size of network, where each *block* consists of 8×8 i.e., *row* × *column*. To explain each step of algorithm we have used either one row or column to show the data flow. For every step the data flow varies so each step need different algorithm. Figure 3.4 shows first row in first block of the network and the connectivity between the processors based on the topological properties of MMT [115]. We have considered that each processor is having a Working Array (*WA*) which consist of the processor index (P_n) and information associated with that processor (I_n). The size of working array depends on the size of network used, i.e. for $n = 8$, the size of *WA* = 8.

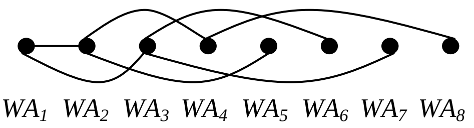
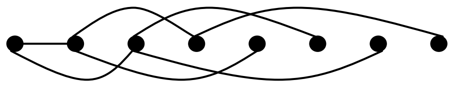


Figure 3.4: Shows initial condition of processors containing WA (only one row of a block of 8 × 8 MMT is shown)

The figure 3.5 (a) shows the position of data after completion of step 1 and figure 3.5 (b) shows the content of WA_1 after step 1.



$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{P_2}$	$\mathbf{P_4}$	$\mathbf{P_5}$	$\mathbf{P_6}$	$\mathbf{P_7}$	$\mathbf{P_8}$
$\mathbf{I_1}$	$\mathbf{I_2}$	$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{I_5}$	$\mathbf{I_6}$	$\mathbf{I_7}$	$\mathbf{I_8}$

Figure 3.5: (a) After Step 1 (b) Content of WA_1 after Step 1

Algorithm 1. Step 1 of AAB

```
a. /* This operation is common between all processors of each row of each block,
b. Each node is represented by $\boldsymbol{n}_{(\alpha,\beta,i,j)}$; where $\alpha$, $\beta$ are the block index and $i$, $j$ are node index
c. The transfer is conducted in order $\frac{n_{(\alpha,\beta,i,j)}}{2\times Count\ of\ Iteration-1} < j \leq \frac{n_{(\alpha,\beta,i,j)}}{2\times Count\ of\ Iteration-1}$ */
1: Starting from each row of each block of network, such that the processor with greater index value will
   transfer data to lower index processors linked according to the topological properties of network.
2: repeat
3:    Select nodes $n_{(\alpha,\beta,i,\frac{N}{2}+1)}, n_{(\alpha,\beta,i,\frac{N}{2}+2)}, n_{(\alpha,\beta,i,\frac{N}{2}+3)}, \ldots n_{(\alpha,\beta,i,N)} \in N$ from each block of network such that at
      each transfer the block is divided in two parts (e.g. if $N$ = 40, number of nodes in blocks will also be
      40 and division will be 1 to 20 and 21 to 40th index position) and transfer message to remaining nodes
      $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, n_{(\alpha,\beta,i,3)}, \ldots n_{(\alpha,\beta,i,\frac{N}{2})} \in N$ linked according to topological properties of this network.
      Note: The message will be transferred from higher processor index to lower.
4:    Select nodes $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, n_{(\alpha,\beta,i,3)}, \ldots n_{(\alpha,\beta,i,\frac{N}{2})} \in N$ (other than the nodes from which message has
      already transferred) from each block of network such that at each transfer these nodes are divided in
      two parts (same as in 3; i.e. $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, \ldots n_{(\alpha,\beta,i,\frac{N}{4})}$, and $n_{(\alpha,\beta,i,\frac{N}{4}+1)}, n_{(\alpha,\beta,i,\frac{N}{4}+2)} \ldots n_{(\alpha,\beta,i,\frac{N}{2})} \in N$.
      Now $n_{(\alpha,\beta,i,\frac{N}{4}+1)}, n_{(\alpha,\beta,i,\frac{N}{4}+2)} \ldots n_{(\alpha,\beta,i,\frac{N}{2})}$ will transfer respective messages to $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, \ldots n_{(\alpha,\beta,i,\frac{N}{4})}$,
      linked according to topological properties of this network.
5: until all nodes have finished transmitting and forwarding.
```

Algorithm 2. Step 2 of AAB

```
a. /* This operation is common between all root processors of each row of each block,
b. Root processors of each row of a block are identified as in figure 3.5 (a),
c. The transfer of information of all root processors of respective rows is conducted according to
   connectivity. */
1: Starting from each row of each block. The root nodes of respective rows will transfer data to
```

```
    connected nodes of that row.
repeat
until all nodes have received the information of root processors.
```

After the completion of step 2 the position of data in a row is shown in figure 3.6. The data from the root node of a row of all blocks of network receives the complete information of that row as the content of WA_1.

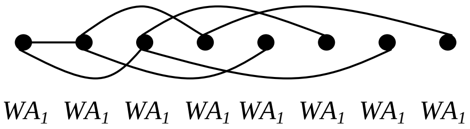


Figure 3.6: After Step 2

Algorithm 3. Step 3 of AAB

```
a. /* This operation is common between all root processors of each column of each block,
b. The transfer is conducted in order $\frac{n_{(\alpha,\beta,i,j)}}{2\times Count\ of\ Iteration-1} < i \le \frac{n_{(\alpha,\beta,i,j)}}{2\times Count\ of\ Iteration-1}$ */
Starting from each column of each block of network, such that the processor with greater index value
    will transfer data to lower index processors linked according to the topological properties of network.
repeat
    Select nodes $n_{(\alpha,\beta,i,\frac{N}{2}+1)}, n_{(\alpha,\beta,i,\frac{N}{2}+2)}, n_{(\alpha,\beta,i,\frac{N}{2}+3)}, \dots n_{(\alpha,\beta,i,N)} \in N$ from each block of network such that
    at each transfer the block is divided in two parts (e.g. if $N$ = 40, number of nodes in blocks will also
    be 40 and division will be 1 to 20 and 21 to 40th index position) and transfer message to remaining
    nodes $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, n_{(\alpha,\beta,i,3)}, \dots n_{(\alpha,\beta,i,\frac{N}{2})} \in N$ linked according to topological properties of this
    network.
```

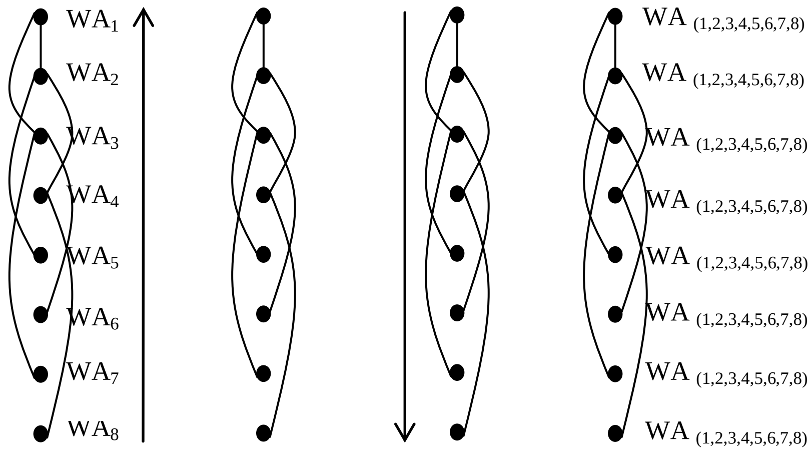


Figure 3.7: a) Step 3 b) After Step 3 c) Step 4 d) After Step 4

Figure 3.7: shows the Step 3 and 4 in which the communication is performed in each column of each block of the network. After the completion of step 4 each column of each block of network consists of complete information of respective column.

Algorithm 4. Step 4 of AAB

```
a. /* This operation is common between all root processors of each column of each block,
b. Root processors of each column of each block are identified as in figure 3.7 (a),
c. The transfer of information of all root processors of respective columns is conducted according to connectivity. */
---
Starting from each column of each block. The root nodes of respective columns will transfer data to connected nodes of that row.
repeat
until all nodes have received the information of root processors.
```

Algorithm 5. Step 5 of AAB (Interblock Communication)

```
/* The step is performed using the horizontal interblock links of this network which transfers the information of all the blocks of respective rows to the root processors of respective block with processor index ($\boldsymbol{j = N}$) */
---
Starting from each blocks of each rows the information is communicated to the root processors of respective block in such a manner that the processor index $n_{(\alpha,\beta,i,j=N)}$.
In one communication step this information is broadcasted to every root processor of respective block of respective row. This step is performed on entire network.
    Note: At the end of this step every root processor contains the information of complete block from which this information is broadcasted.
```

Algorithm 6. Step 6 of AAB (Interblock Communication)

```
a. /* This step uses algorithm 3 for communicating the information received after step 5 (algorithm 5).
b. This operation is common between all root processors of each column of each block,
c. The transfer is conducted in order $\frac{\boldsymbol{n_{(\alpha,\beta,i,j)}}}{\boldsymbol{2\times Count\ of\ Iteration-1}} < i \leq \frac{\boldsymbol{n_{(\alpha,\beta,i,j)}}}{\boldsymbol{2\times Count\ of\ Iteration-1}}$ */
---
Starting from each column of each block of network, such that the processor with greater index value will transfer data to lower index processors linked according to the topological properties of network.
repeat
Select nodes $n_{(\alpha,\beta,i,\frac{N}{2}+1)}, n_{\left(\alpha,\beta,i,\frac{N}{2}+2\right)}, n_{\left(\alpha,\beta,i,\frac{N}{2}+3\right)}, \ldots n_{(\alpha,\beta,i,N)} \in N$ from each block of network such that at each transfer the block is divided in two parts and transfer message to remaining nodes $n_{(\alpha,\beta,i,1)}, n_{(\alpha,\beta,i,2)}, n_{(\alpha,\beta,i,3)}, \ldots n_{(\alpha,\beta,i,\frac{N}{2})} \in N$ linked according to topological properties of this network.
```

Algorithm 7. Step 7 of AAB (Interblock Communication)

```
/* One-to-all broadcast is used in the block*/
---
To transfer the information of a block in a row to other block of respective rows the one-to-all broadcast algorithm is used.
Note: At the end of this step, complete blocks of each row have information of all processors in that row.
```

Algorithm 8. Step 8 of AAB (Interblock Communication)

```
/* The step is performed using the horizontal interblock links of this network which transfers the information of all the blocks of respective columns to the root processors of respective block with processor index ($\boldsymbol{i = N}$) */
---
Starting from each blocks of each columns the information is communicated to the root processors of
```

```
   respective block in such a manner that the processor index $n_{(α,β,i=N,j)}$.
2: In one communication step this information is broadcasted to every root processor of respective block of respective column. This step is performed on entire network.
   Note: At the end of this step every root processor contains the information of complete block from which this information is broadcasted.
```

Algorithm 9. Step 9 of AAB

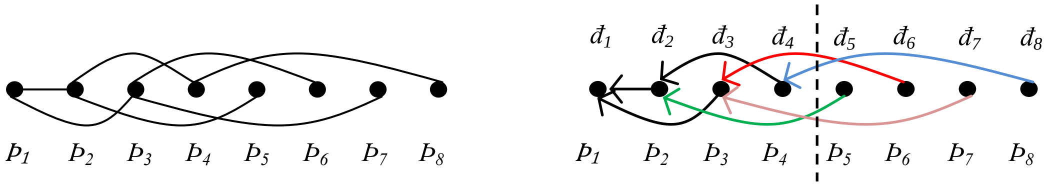
```
a. /* This operation is common between all processors of each row of each block,
b. Each node is represented by $\boldsymbol{n_{(α,β,i,j)}}$;
c. The transfer is conducted in order $\frac{n_{(α,β,i,j)}}{2×Count\ of\ Iteration-1} < i \le \frac{n_{(α,β,i,j)}}{2×Count\ of\ Iteration-1}$ */
---
1: Starting from each row of each block of network, such that the processor with greater index value will transfer data to lower index processors linked according to the topological properties of network.
2: repeat
3:    Select nodes $n_{(α,β,i,\frac{N}{2}+1)}, n_{(α,β,i,\frac{N}{2}+2)}, n_{(α,β,i,\frac{N}{2}+3)}, \dots n_{(α,β,i,N)} \in N$ from each block of network such that at each transfer the block is divided in two parts and transfer message to remaining nodes $n_{(α,β,i,1)}, n_{(α,β,i,2)}, n_{(α,β,i,3)}, \dots n_{(α,β,i,\frac{N}{2})} \in N$ linked according to topological properties of this network.
      Note: The message will be transferred from higher processor index to lower.
4:    Select nodes $n_{(α,β,i,1)}, n_{(α,β,i,2)}, n_{(α,β,i,3)}, \dots n_{(α,β,i,\frac{N}{2})} \in N$ (other than the nodes from which message has already transferred) from each block of network such that at each transfer these nodes are divided in two parts (same as in 3; i.e. $n_{(α,β,i,1)}, n_{(α,β,i,2)}, \dots n_{(α,β,i,\frac{N}{4})}$, and $n_{(α,β,i,\frac{N}{4}+1)}, n_{(α,β,i,\frac{N}{4}+2)} \dots n_{(α,β,i,\frac{N}{2})} \in N$. Now $n_{(α,β,i,\frac{N}{4}+1)}, n_{(α,β,i,\frac{N}{4}+2)} \dots n_{(α,β,i,\frac{N}{2})}$ will transfer respective messages to $n_{(α,β,i,1)}, n_{(α,β,i,2)}, \dots n_{(α,β,i,\frac{N}{4})}$, linked according to topological properties of this network.
5: until all nodes have finished transmitting and forwarding.
```

Algorithm 10. Step 10 of AAB

```
/* AAB is used in the block*/
---
Select block from each column to transfer information of a block in a column to other block of respective columns for this AAB is used.
Note: At the end of this step, all the processors of each block contains information of all processors of the network.
```

3.3 *LNC on AAB using MMT*

We have implemented LNC for each step to make the communication faster and increase the rate of information transmitted from each node. We considered network as delay-free (acyclic) and $o(l) \neq d(l)$. The algorithm results are analyzed later with $n = 8$ processors. For each step, independent and different algorithms are used and linear coding is implemented with each algorithm. According to algorithm 1, all processors transfers data with $n = 8$ and *count* = 1 to 2 i.e, $(8/(2×1 – 1)) < j \le (8/2×1) = 8 < j \le 4$. This means the processors $Ᵽ_1$, $Ᵽ_2$, $Ᵽ_3$ and $Ᵽ_4$ will receive data from $Ᵽ_5$, $Ᵽ_6$, $Ᵽ_7$ and $Ᵽ_8$, shown in figure 3.8.



(a) (b)

Figure 3.8: (a) Shows the indexing of processors with respect to nodes in the figure. (b) Shows the direction of flow of data in step 1 of AAB algorithm on MMT, $Ᵽ_1$, $Ᵽ_2$, $Ᵽ_3$ and $Ᵽ_4$ are the processor receiving data and $Ᵽ_5$, $Ᵽ_6$, $Ᵽ_7$ and $Ᵽ_8$ are the sending processors. The dotted line distinguishes between the receiving and sending processors in first iteration of step 1.

Step 1: Linear coding is implemented on $Ᵽ_1$, $Ᵽ_2$ and $Ᵽ_3$ processors, as these are receiving a set of data form source processors $Ᵽ_5$, $Ᵽ_6$, $Ᵽ_7$ and $Ᵽ_8$ in first iteration. Processor $Ᵽ_8$ is source and $Ᵽ_4$ is its destination; $Ᵽ_7$ and $Ᵽ_6$ are sources and $Ᵽ_3$ is their destination; lastly in log$n$ iteration i.e. (3 iteration for $n = 8$), $Ᵽ_1$ will receive data from $Ᵽ_2$ and $Ᵽ_3$. After implementation of LNC according to LCM-PA on these sources and destinations, step 1 will work as in figure 3.9. During first iteration of AAB on MMT, LCM-PA will work as in figure 3.9 (a). Nodes $Ᵽ_5$, $Ᵽ_6$, $Ᵽ_7$ and $Ᵽ_8$ send data to $Ᵽ_2$, $Ᵽ_3$, $Ᵽ_3$ and $Ᵽ_4$ respectively. So, the complete set of data from all processors reached processor $Ᵽ_1$, i.e. after execution of step 1 all data in a row will reach its root processors. But due to LCM-PA the data reached $Ᵽ_1$ will have time complexity of $(\log n - 1)$, as one step is reduced during transfer of the data using LCM-PA model.

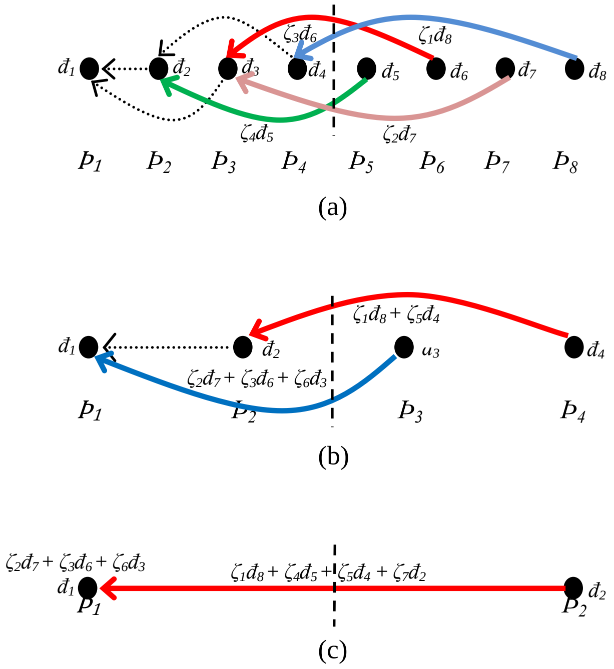
Step 2: The root processors of each row, ($Ᵽ_1$: root processor of first block and first row) will broadcast the data (from $Ᵽ_1$: $\zeta_1 đ_8 + \zeta_2 đ_7 + \zeta_3 đ_6 + \zeta_4 đ_5 + \zeta_5 đ_4 + \zeta_6 đ_3 + \zeta_7 đ_2$) to all the processors of respective row using intrablock links transfer, see figure 3.10.

The time complexity for this step will be reduced by n i.e., $n(\log n\text{-}1)$. This step is a broadcasting step in each block with intrablock links of MMT. At the end of this step, complete data from root processor is received by other processors of that row. LCM-PA is applied at the same level as in step 1, but the size of data increases to n.

Step 3: This step is similar to step 1, but in this step the data is broadcasted in column-wise order of each block. Linear coding is implemented on $Ᵽ_{11}$, $Ᵽ_{12}$ and $Ᵽ_{13}$ processors, as these are receiving a set of data form source processors $Ᵽ_{15}$, $Ᵽ_{16}$, $Ᵽ_{17}$ and $Ᵽ_{18}$ in first iteration. Processor $Ᵽ_{18}$ is source and $Ᵽ_{14}$ is its destination; $Ᵽ_{17}$ and $Ᵽ_{16}$ are sources and $Ᵽ_{13}$ is their destination;

lastly in log$n$ iteration i.e. (3 iteration for $n = 8$), $Þ_{11}$ will receive data from $Þ_{12}$ and $Þ_{13}$. After implementation LCM-PA on these sources and destinations, step 3 works as in figure 3.11.

Step 4: In this step all the root processors of each column and each block, ($Þ_{11}$: root processor of first block and first column) will broadcast the data (from $Þ_{11}$: $\zeta_1 đ_{18} + \zeta_2 đ_{17} + \zeta_3 đ_{16} + \zeta_4 đ_{15} + \zeta_5 đ_{14} + \zeta_6 đ_{13} + \zeta_7 đ_{12}$) to all the processors of respective column using intrablock links transfer, see figure 3.12. The time complexity of this step is reduced by n^2 i.e, $n^2(\log n\text{-}1) = n^2\log n\text{-}\ n^2$. The coefficient value (ζ_i) in step 4 is different from the coefficient value in step 1.



(a)

(b)

(c)

Figure 3.9: (a) Iteration first of step 1; data from processers $Þ_5$, $Þ_6$, $Þ_7$ and $Þ_8$ is sent to processors to $Þ_4$, $Þ_3$, $Þ_3$ and $Þ_2$ respectively. (b) Iteration second of step 1; data from processers $Þ_4$ and $Þ_3$ is sent to processors to $Þ_2$ and $Þ_1$ respectively. (c) Iteration third of step 1; data from processers $Þ_2$ is sent to processors $Þ_1$.

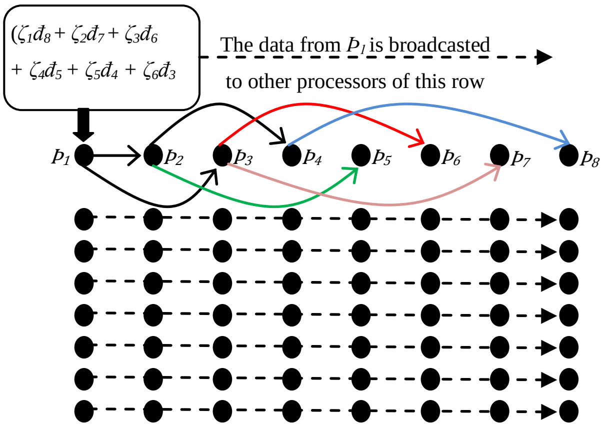
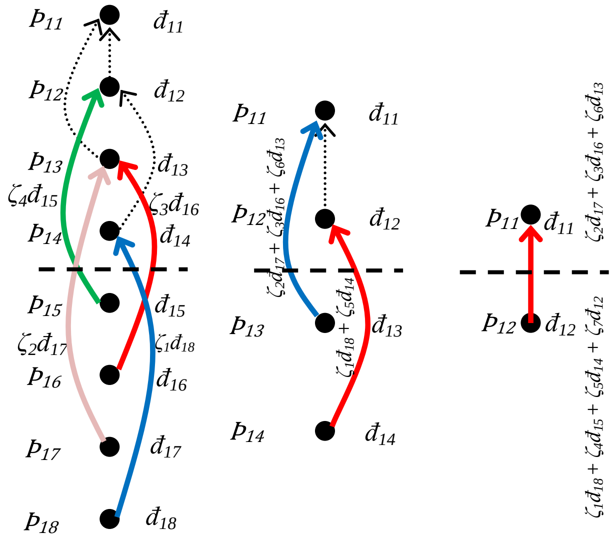


Figure 3.10: The data from each row root processor is broadcasted to other processors of respective row in each block.

Step 5: After step 4, each processors of respective columns contains information of all processors of that column. Step 5 perform the interblock communication using the horizontal interblock links which transfers this information (of all the blocks of respective rows) to the root processors (of respective block) and this requires one communication step (CS). The time complexity of this step will be same as of AAB i.e. 1CS.

Step 6: Using step 3, for transferring information of all the processors in the column at the processors with P_ID ($j=n$), so the WA of all the processors is transferred in the column in the order $n/(2count–1) < j \leq n/(2count)$. Time Complexity of step 6: $n^3\log n$.

Step 7: Call one–to–all algorithm in the block to transfer the INFO of other blocks (of respective rows) in $n^3 logn$ time. At the end of this step, complete blocks of each row have INFO of all the processors in that row. Time Complexity of step 7: $n^3\log n$.



(a) (b) (c)

Figure 3.11: (a) Iteration first of step 3; data from processers $Ƥ_{15}$, $Ƥ_{16}$, $Ƥ_{17}$ and $Ƥ_{18}$ is sent to processors to $Ƥ_{14}$, $Ƥ_{13}$, $Ƥ_{13}$ and $Ƥ_{12}$ respectively. (b) Iteration second of step 3; data from processers $Ƥ_{14}$ and $Ƥ_{13}$ is sent to processors to $Ƥ_{12}$ and $Ƥ_{11}$ respectively. (c) Iteration third of step 3; data from processers $Ƥ_{12}$ is sent to processors $Ƥ_{11}$.

Step 8: This step performs the interblock communication using horizontal link transfer that transfers the INFO (of all the blocks of respective column) to the root processors (of respective block) with P_ID ($i=n$), and this requires one communication step. Time Complexity of step 8: *1CS.*

Step 9: Using step 1 transfer of INFO of all the processors with P_ID ($i=n$). Time Complexity of step 9: n^4log$n$.

Step 10: Call AAB algorithm in the block to transfer the INFO of other blocks that column in the block with $n^4 logn$ time complexity. Time Complexity of step 10: n^4log$n$. At the end of this step all the processors of each block have the INFO of all processors of other blocks.

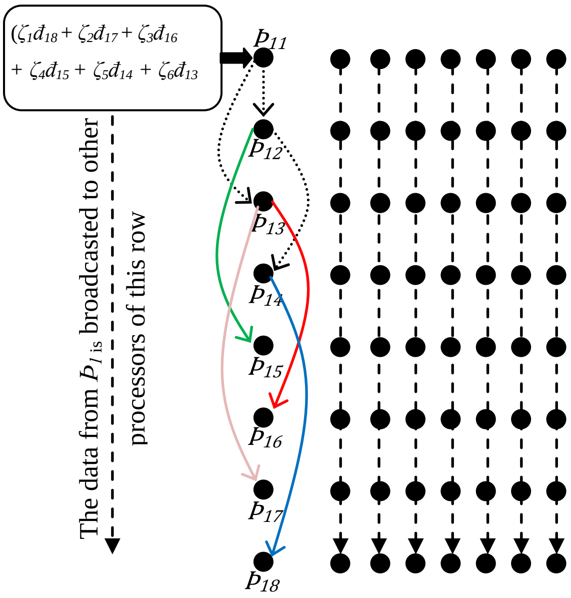


Figure 3.12: The data from each column root processor is broadcasted to other processors of respective column in each block.

3.4 *Results and Simulations*

To implement linear coding using AAB on MMT enables sharing of data between multiple processors more convenient and easy. As the algorithm becomes complex, the involvement of processors also increases. For parallel architectures, important issue is to make these architectures more processor utilitarian, otherwise the processors in these architectures are idle, and all are not in use at every step of algorithms. Higher coefficient participates to broadcast data compared with LCM-PA. So, this makes the algorithm less complex as fewer amounts of coefficients involves in broadcasting data using linear coding. While broadcasting the data in AAB, the time involved to communicate and deliver/receive data from different processors is more. The fall of time complexity at different number of processors makes positive implementation of algorithm.

The algorithm starts with executing each step in the order defined as step 1...step 10. As it starts, the involvement of each processor also increases. In parallel processing the algorithm starts with active processor and involves other processors as it progresses [109]. Figure 3.13 show participation of processors with average percentage of iteration in each step.

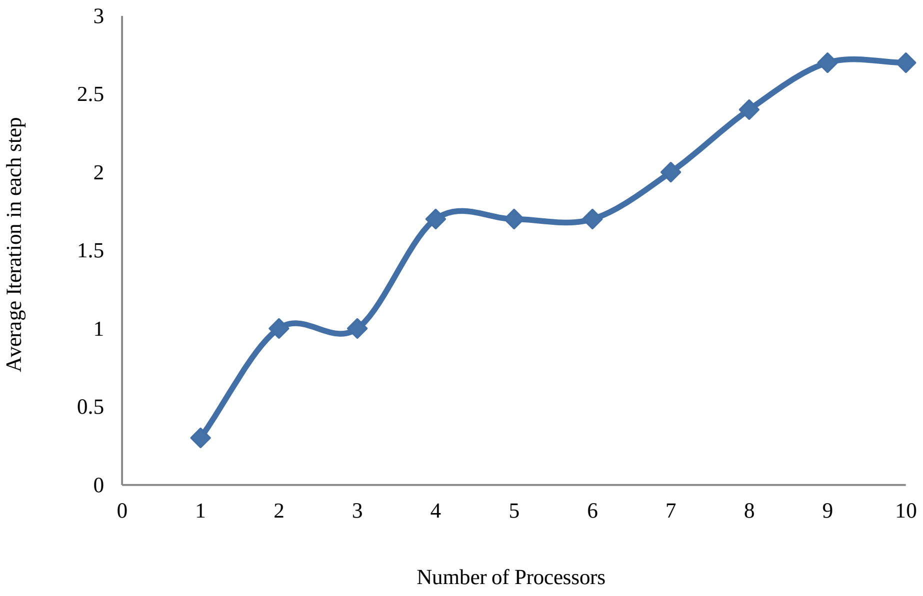


Figure 3.13: Involvement of processors at different steps of algorithm.

Based on result in figure 3.13, we conclude that as the iterations increase the involvement of processors also increases. The algorithm with LCM-PA approach utilizes the maximum number of processors besides without LCM-PA approach. So to utilize processors in parallel architectures also increases using linear coding.

3.5 *Chapter Outline*

This chapter presents LCM-PA: a model of linear coding on parallel architecture. This proposed model efficiently implements AAB algorithm over MMT network. The results are discussed with comparative time complexity after implementation with LCM-PA. Further it is shown that the model is network independent and can be implemented on any parallel architecture considering the topological properties of that architecture.

CHAPTER 4

LINEAR-CODE MULTICAST ON PARALLEL ARCHITECTURES (LCM−PA)

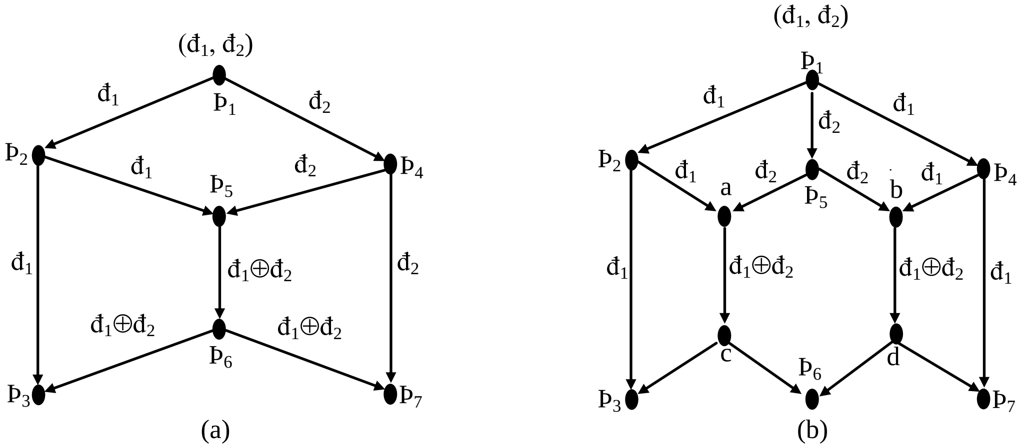
This chapter proposes *Linear-Code Multicast on Parallel Architectures* (LCM−PA). It is well-known that parallel architectures intend fast receiving of complete information at several nodes and reduces the execution time by dividing task between various nodes. This practice requires high communication (as data from diverse nodes travels simultaneously) and computation (as diverse nodes perform different operations on data simultaneously). Several architectures have been proposed to overcome the problem of high communicational and computational time complexity for transferring and receiving information. In this chapter, to reduce the complexity of such communication, Linear Network Coding (LNC) is implemented in parallel environment. For verification of the approach some parallel architecture are considered for implementing network coding approach and the results are examined on these networks in a generic environment. Further a standard approach for parallel networks is formulated, which shows that, by applying this approach effect of faulty nodes, information size and communication complexity exponentially decreases with code length.

4.1 *Information Rate in Parallel Architectures*

Parallel architectures are the means to enable fast and efficient communication. It reduces time to complete a task as several processing units perform parallel execution. These architectures distribute large problems on several processing units and solves in less time than any single processor system. To utilize algorithms for efficient and fast communication, several interconnection networks have developed. In topologies with multiple nodes taking part at a time, the issues like *faulty nodes* (receiving multiple data from multiple sources), information size and communication complexity are prominent. Data communication needs better approaches to 1) remove these issues and 2) utilize processing units. *Linear Network Coding* (LNC) evolved as an efficient approach for data communication [1, 69, 120-123].

The information rate from source node to sink can potentially become higher when coding scheme is wider [1]. Further, in [1] by linear coding alone, the rate at which a message reaches each node, can achieve the individual max-flow bound. And provide a realization of the transmission scheme for both acyclic and cyclic networks [1].

We have considered a general approach for parallel multicast in multisource networks which is possible with correlated sources (figure 4.1). In each incoming transmission from source, the destination has knowledge of overall linear combination of data (see figure 4.1). This information updates at each coding node by applying the same linear mapping to the coefficient vectors as applied to the information signals. As an example, in a directed parallel network (*Ň*) (as in figure 4.1) set of two bits ($đ_1$, $đ_2$) is multicast from the source node $Þ_1$ (unique node, without any incoming at that instant of time) to other nodes of network. Figure 4.1 shows the network coding at each node, when information is multicast in a network. So, let us explain network coding using a network *Ň* as given in figure 4.1(a). At $Þ_1$, the data set ($đ_1$, $đ_2$) is multicast to destination nodes $Þ_3$ and $Þ_7$. The transmission progresses as: ($đ_1$, $đ_2$) traverse via node $Þ_2$ and $Þ_4$ nodes of network *Ň*; $Þ_2$ receives $đ_1$ and $Þ_4$ receives $đ_2$. Then both $Þ_2$ and $Þ_4$ transfer the information to $Þ_5$. Now as $Þ_5$ receives two diverse data from different sources. To avoid data loss $Þ_5$ perform network coding ($đ_1 \oplus đ_2$). Node $Þ_5$ and $Þ_6$ send this encoded data to $Þ_3$ and $Þ_7$ which is decodes to receive the data ($đ_1$, $đ_2$).



(a) (b)

Figure. 4.1: Networks used to explain LNC at different nodes to perform complete data transfer. Each link in these networks shows data transmission.

This approach shows that multicast of different data is possible when network coding is used in the network. It is right to say that flow of information and flow of physical commodities are two diverse ideas [1]. So, the coding information does not increase the information content. Capacity of network to transmit data from the source to destination can become higher if the coding scheme becomes wider, but it limits to max-flow for a wide coding scheme [1]. We have carried out this approach with parallel networks, and it results in a novel efficient manner of data communication. For parallel transmission of information, the linearity of code makes encoding (at source end) and decoding (at receiving end) easy in practice.

4.2 *Review stage for network coding*

In figure 4.1, we have considered a single source acyclic network to multicast two bits $đ_1$ and $đ_2$ to non-source nodes. However, in parallel networks there are multiple sources to multicast data to other several and diverse non-source nodes. In parallel networks, nodes may act as source and non-source nodes at different step of an algorithm. Parallel networks are acyclic at each step of an algorithm thus needs network coding. We have considered the approaches and examples of Li *et al.* [1] to fetch-out basic needs of network coding and justify our implementation on parallel networks. To follow the needs for network coding, let us consider network as in figure 4.1(a) in which $Ᵽ_1$ is a source node and $Ᵽ_2$, $Ᵽ_4$, $Ᵽ_5$ and $Ᵽ_6$ are routing nodes between $Ᵽ_1$ and $Ᵽ_3$, $Ᵽ_7$. Where, $Ᵽ_6 Ᵽ_3$ stands for a channel between node $Ᵽ_6$ and $Ᵽ_3$.

Parallel networks consist of busy channels in the flow. These are 1) channels that do not form directed cycles, (for example figure 4.1(a) and consider any sub-network between $Ᵽ_1$ and $Ᵽ_3$, $Ᵽ_7$, let the sub-network is $Ᵽ_1$, $Ᵽ_2$, $Ᵽ_5$, $Ᵽ_6$, $Ᵽ_3$ which is acyclic) 2) For nodes except $Ᵽ_1$ and $Ᵽ_5$, the number of incoming busy channels is equal to outgoing busy channels 3) The number of outgoing channels to $Ᵽ_1$ is equals to number of incoming channels to $Ᵽ_5$. This shows that if information rate increases, the rate of network coding also increases [1]. For parallel architectures having several coded nodes, communication using network coding may become impractical. This setback of parallel networks solves using Max-Flow law of information flow [124] (minimize the number of nodes to be coded, despite the acyclic constraint). According to Max-Flow Min-Cut Theorem, for every non-source node $Ᵽ_2$ (figure 4.2 and 4.3 for explanation of Max-Flow Min-Cut on network $\check{N}$ and M) the minimum value of all cuts between $Ᵽ_1$ and $Ᵽ_2$ is equal to $maxflow(Ᵽ_2)$ [124, Ch. 4, Theorem 2.3].

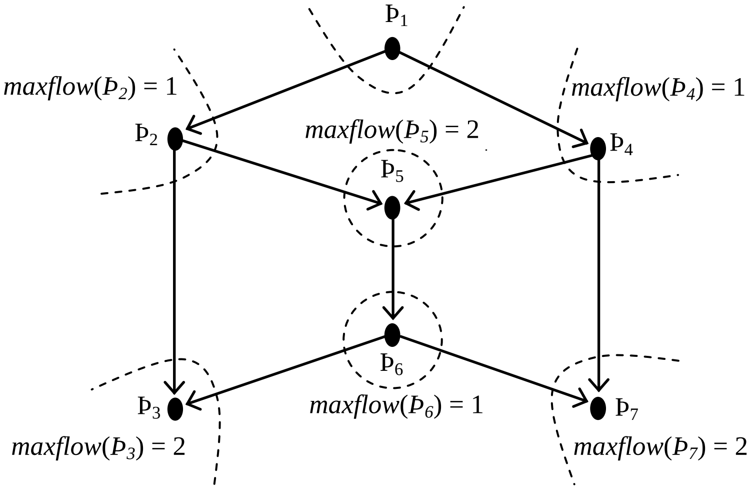


Figure 4.2: Max-Flow for Network $\check{N}$.

Using the conventions defined by Li *et* al. [1], assume that d is the maximum $maxflow(Þ_2)$ over all $Þ_2$ and the symbol Ω represents a d-dimensional vector space over a large base field. Let us define *linear-code multicast* (*LCM*) ν on a communication network ($\check{N}$, $Þ_1$) with vector space $\nu(Þ_6)$ assigned to every node $Þ_2$ and a vector $\nu(Þ_6 Þ_3)$ to every channel $Þ_6 Þ_3$ such that

1) $\nu(Þ_1) = \Omega$;
2) $\nu(Þ_6 Þ_3) \in \nu(Þ_6)$ for every channel $Þ_6 Þ_3$;
3) For φ, collection of non-source nodes in the network

$$\langle \nu(Þ_2)\colon Þ_2 \in \varphi \rangle = \langle \{\nu(Þ_6 Þ_3)\colon Þ_6 \notin \varphi,\ Þ_3 \in \varphi\}\rangle.$$

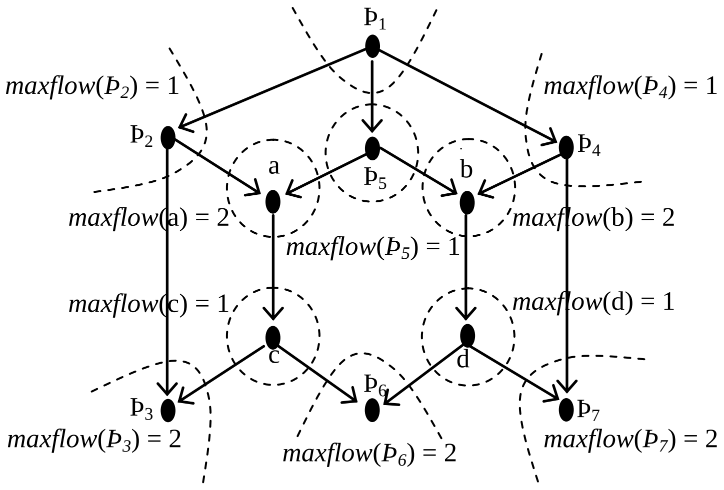


Figure 4.3: Max-Flow for Network M.

Example 1: Suppose in network $\check{N}$ of figure 4.1(a), $Þ_1$ multicasts two bits $đ_1$ and $đ_2$ to destination nodes $Þ_3$ and $Þ_7$. Using LCM ν specified by

$$\nu(Þ_1Þ_2) = \nu(Þ_2Þ_5) = \nu(Þ_2Þ_3) = \begin{pmatrix}1\\0\end{pmatrix}$$

$$\nu(Þ_1Þ_4) = \nu(Þ_4Þ_5) = \nu(Þ_4Þ_7) = \begin{pmatrix}0\\1\end{pmatrix}$$

$$\nu(Þ_5Þ_6) = \nu(Þ_6 Þ_3) = \nu(Þ_6 Þ_7) = \begin{pmatrix}1\\1\end{pmatrix}$$

Example 2: Suppose in network M of figure 4.1(b), $Þ_1$ multicasts two bits $đ_1$ and $đ_2$ to destination nodes $Þ_3$, $Þ_6$ and $Þ_7$. Using LCM ν specified by

$$\nu(Þ_1Þ_2) = \nu(Þ_1Þ_4) = \nu(Þ_2a) = \nu(Þ_4b) = \nu(Þ_2Þ_3) = \nu(Þ_4Þ_7) = \begin{pmatrix}1\\0\end{pmatrix}$$

$$\nu(Þ_1Þ_5) = \begin{pmatrix}0\\1\end{pmatrix}$$

$$\nu(ac) = \nu(cÞ_3) = \nu(cÞ_6) = \nu(bd) = \nu(dÞ_6) = \nu(dÞ_7) = \begin{pmatrix}1\\1\end{pmatrix}$$

Data sent on a channel is the product of information vector with the assigned channel vector. Example: data set on $Þ_5$ $Þ_6$ is $đ_1 \oplus đ_2$ in network $\check{N}$ and data set on ac and bd is $đ_1 \oplus đ_2$ in network M of figure 4.1.

4.3 *LCM–PA (Linear-Code Multicast on Parallel Architectures)*

To demonstrate network coding potential for parallel networks and justify our approach, we have implemented it on Recursive Diagonal Torus (RDT), Multi Mesh of Trees (MMT), Mesh of Trees (MoT), Multi-Mesh (MM) and 2D-Mesh networks [116, 118, 119, 125, 126]. We have implemented the conventions of network coding to ensure that communication in these parallel networks can improve. For clarity, we have defined the topological properties of different parallel architectures. For diverse networks, different communication steps are used to show the flow of data in each step. For each network different algorithms are used and the data flow varies for each step. Table 3.1 shows characteristics of various processor organizations based on some of the network optimization parameters.

4.3.1 *Network coding on RDT (Recursive Diagonal Torus)*

RDT (figure 4.4) network provides better performance than 2D/3D/4D torus network and is developed with the best use of recursively structures diagonal torus connection [124]. This network has a diameter 11 for 2^{16} nodes and 8 edges per node. However, in other parallel networks (while communicating between different nodes having different data) the receiving node can receive one data from source nodes at a time (shown in figure 4.5). Due to arrival of multiple data at a receiver node it results data loss. This is handled by coding such nodes so that no data loss may occur. The *faulty nodes* are coded to XOR the data received from multiple nodes. This will eradicate the problem of data loss. In RDT topology, network coding principle is implemented to remove the *faulty nodes* and accomplish data communication. Figure 4.5 shows RDT topology which contains 32 communicating nodes out of which 8, 10-16, 18-24, 26-32 are *faulty nodes*. These nodes are receiving more than one input at unit time which eradicates either of the received data. This makes communication in this network complex. The communication is possible by using the network coding principle to XOR multiple data received at a node in unit time. Figure 4.6 shows an implementation of network coding in RDT network.

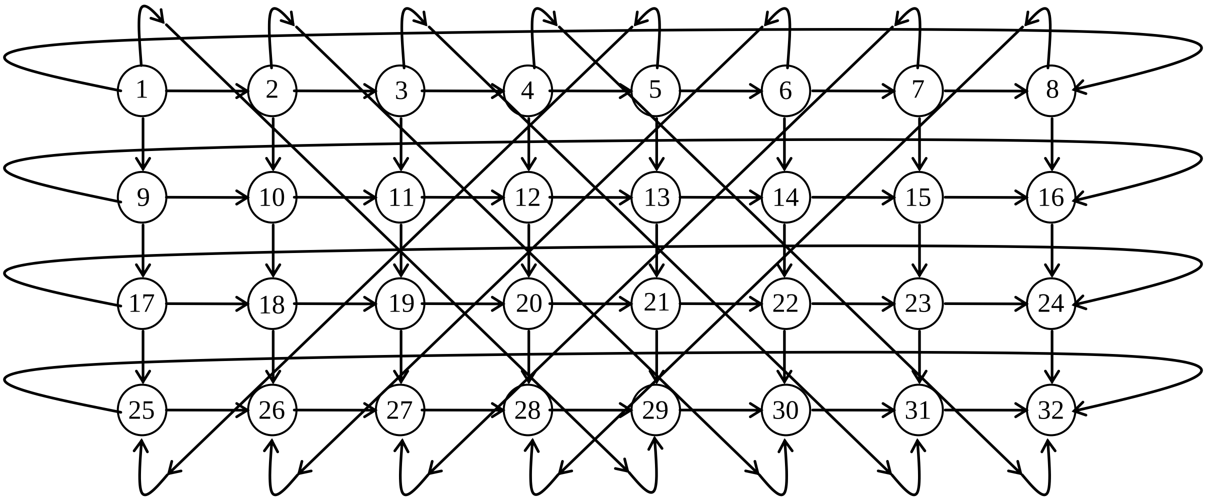


Figure 4.4: RDT network.

To implement network coding on RDT network, it is required to find *maxflow* to evaluate total number of inputs to different nodes. While calculating *maxflow* in figure 4.6, nodes receiving a different number of data are to be identified (table 4.1). The non-source nodes are marked for evaluating a different number of inputs to nodes. These non-source nodes are shown in figure 4.7.

Table 4.1: *Maxflow* in RDT network.

Nodes Receiving Single Data Input	Nodes Receiving Two Data Inputs	Nodes Receiving Three Data Inputs	Nodes Receiving Four Data Inputs
maxflow ($Þ_{2}$) = 1 *maxflow* ($Þ_{3}$) = 1 *maxflow* ($Þ_{4}$) = 1 *maxflow* ($Þ_{5}$) = 1 *maxflow* ($Þ_{6}$) = 1 *maxflow* ($Þ_{7}$) = 1 *maxflow* ($Þ_{9}$) = 1 *maxflow* ($Þ_{17}$) = 1	*maxflow* ($Þ_{8}$) = 2 *maxflow* ($Þ_{10}$) = 2 *maxflow* ($Þ_{11}$) = 2 *maxflow* ($Þ_{12}$) = 2 *maxflow* ($Þ_{13}$) = 2 *maxflow* ($Þ_{14}$) = 2 *maxflow* ($Þ_{15}$) = 2 *maxflow* ($Þ_{18}$) = 2 *maxflow* ($Þ_{19}$) = 2 *maxflow* ($Þ_{20}$) = 2 *maxflow* ($Þ_{21}$) = 2 *maxflow* ($Þ_{22}$) = 2 *maxflow* ($Þ_{23}$) = 2 *maxflow* ($Þ_{25}$) = 2	*maxflow* ($Þ_{16}$) = 3 *maxflow* ($Þ_{24}$) = 3 *maxflow* ($Þ_{26}$) = 3 *maxflow* ($Þ_{27}$) = 3 *maxflow* ($Þ_{28}$) = 3 *maxflow* ($Þ_{29}$) = 3 *maxflow* ($Þ_{30}$) = 3 *maxflow* ($Þ_{31}$) = 3	*maxflow* ($Þ_{32}$) = 4

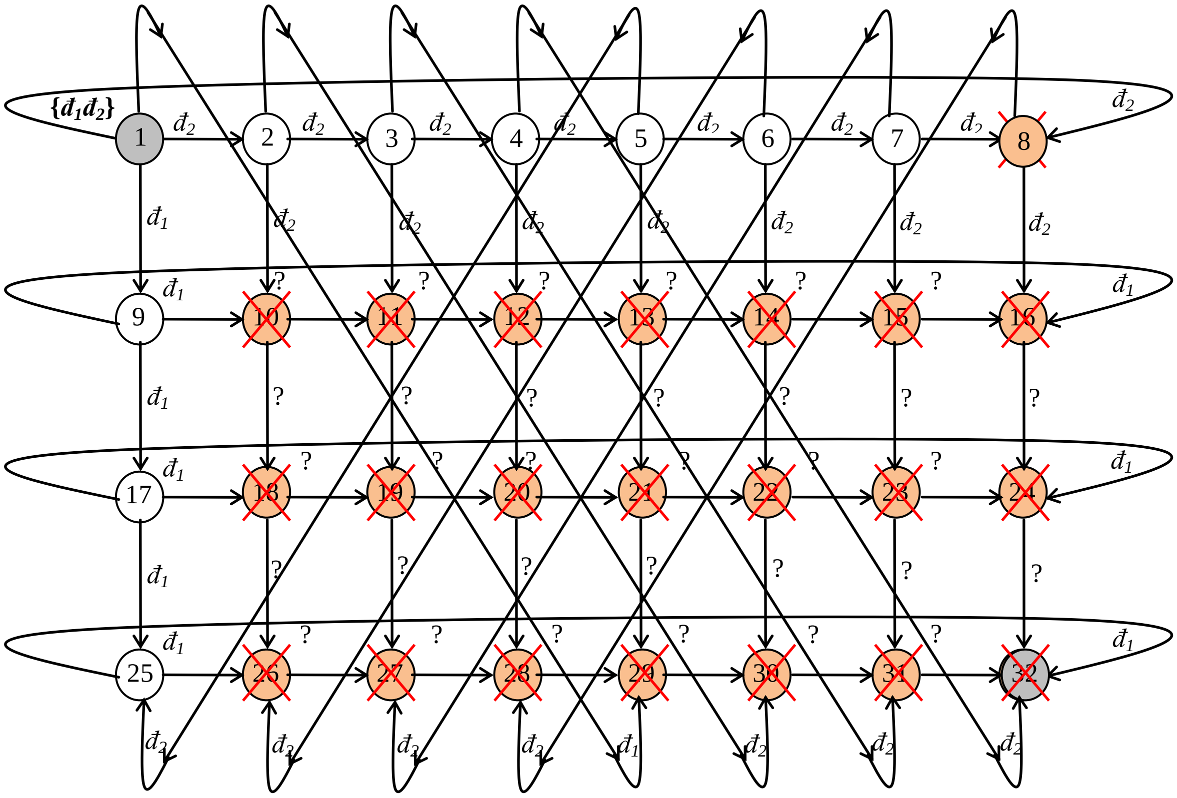


Figure 4.5: Nodes in RDT network with *faulty nodes*.

In the figure given below, RDT network consists of four different *maxflow* on different nodes. If the statistics of this network is evaluated, eight nodes ($Þ_{2}$, $Þ_{3}$, $Þ_{4}$, $Þ_{5}$, $Þ_{6}$, $Þ_{7}$, $Þ_{9}$, and $Þ_{17}$) have *maxflow* = 1; fourteen nodes ($Þ_{8}$, $Þ_{10}$, $Þ_{11}$, $Þ_{12}$, $Þ_{13}$, $Þ_{14}$, $Þ_{15}$, $Þ_{18}$, $Þ_{19}$, $Þ_{20}$, $Þ_{21}$, $Þ_{22}$,

$Ᵽ_{23}$, and $Ᵽ_{25}$) have *maxflow* = 2; eight nodes ($Ᵽ_{16}$, $Ᵽ_{24}$, $Ᵽ_{26}$, $Ᵽ_{27}$, $Ᵽ_{28}$, $Ᵽ_{29}$, $Ᵽ_{30}$, and $Ᵽ_{31}$)have *maxflow* = 3; and one node ($Ᵽ_{32}$) have *maxflow* = 4.

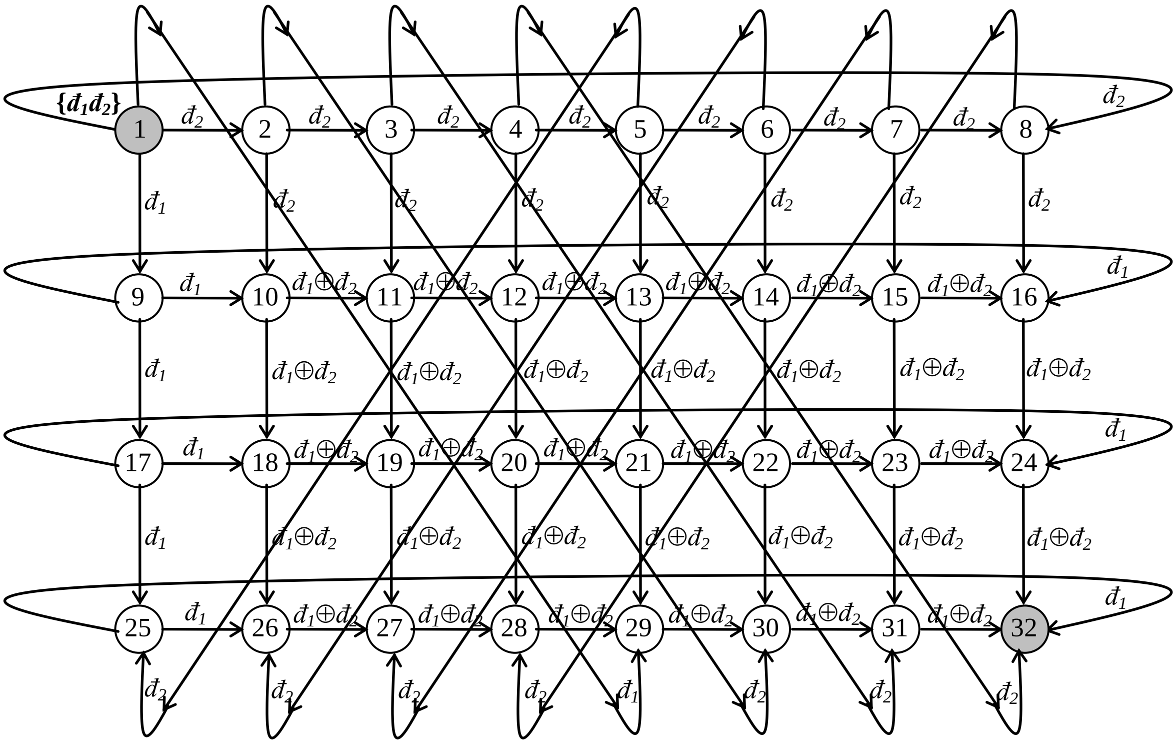


Figure 4.6: RDT network with network coding. Source node 1 transfers data $đ_1đ_2$ to destination node 32.

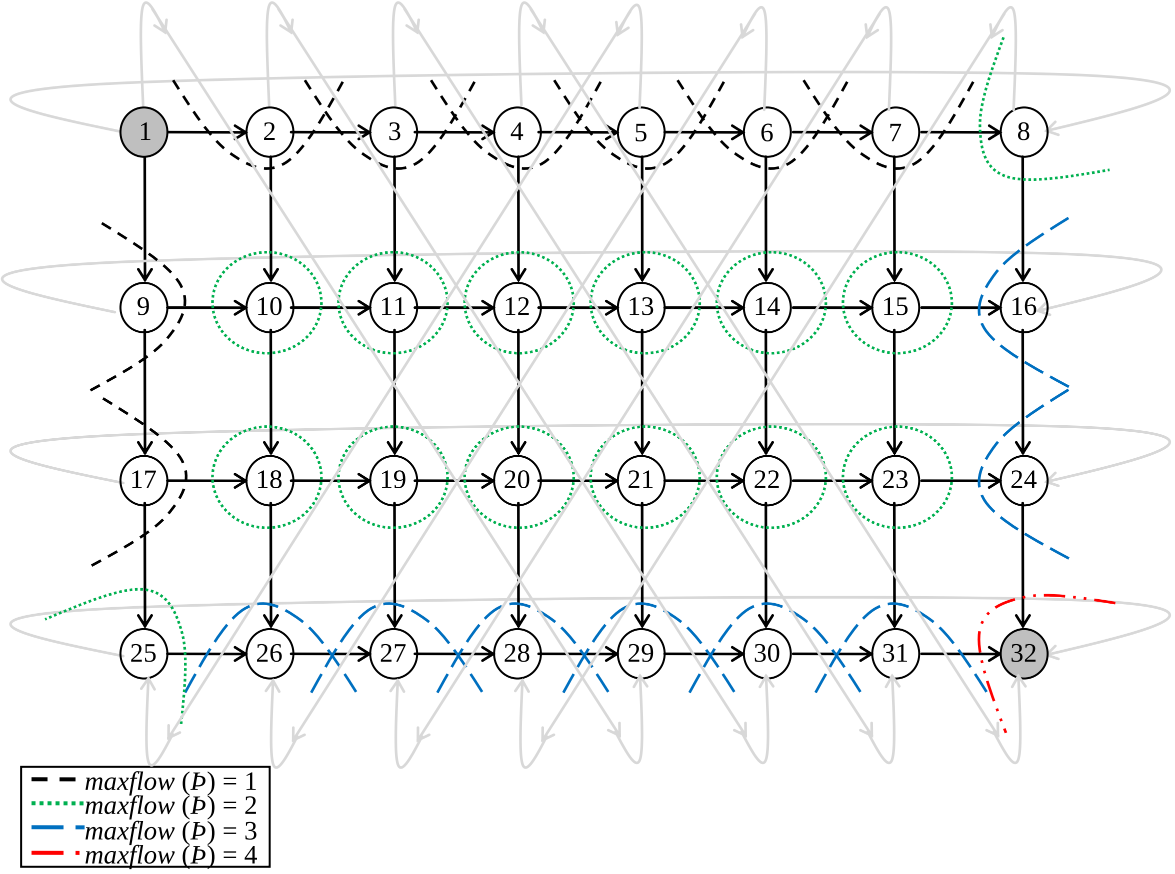


Figure 4.7: RDT network with different values of *maxflow* at different nodes.

The statistics of *maxflow* in RDT network is: as *maxflow* increases the number of nodes decreases. That is, when *maxflow* = 1, number of nodes = 8; when *maxflow* = 2, number of nodes = 14; when *maxflow* = 3, number of nodes = 8; and when *maxflow* = 4, number of nodes = 1. This validates possibility to implement network coding on RDT. The simulation results of this prospect are given in figure 4.8.

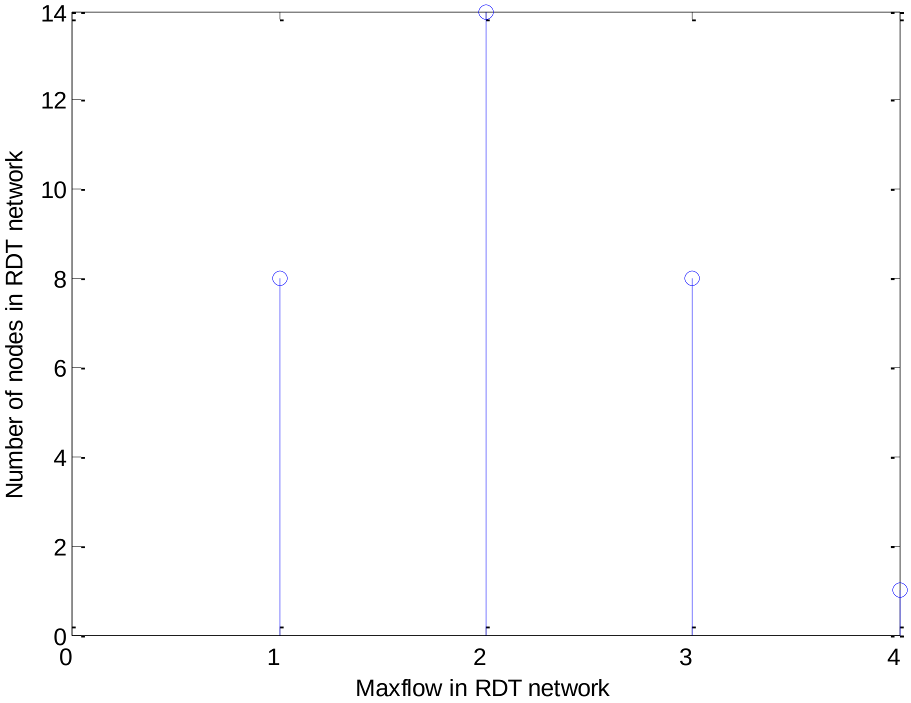


Figure 4.8: Different values of *maxflow* at different number of nodes.

To achieve linear code multicast in RDT network, a vector space is assigned to every node and a vector to every channel to achieve parameters as in section 4.1. Assuming that the flow of data between two nodes ($đ_1$, $đ_2$) is multicast (figure 4.4). Some channels in this network contain either $đ_1$ or $đ_2$ and some contain $đ_1 \oplus đ_2$. Based on the above evaluated *maxflow*, the vector space is generated (figure 4.9) and using this vector space the linear code multicast is achieved. The vector space is shown below:

$$\nu(Ᵽ_{1}Ᵽ_{9}) = \nu(Ᵽ_{1}Ᵽ_{29}) = \nu(Ᵽ_{9}Ᵽ_{10}) = \nu(Ᵽ_{9}Ᵽ_{16}) = \nu(Ᵽ_{9}Ᵽ_{17}) = \nu(Ᵽ_{17}Ᵽ_{18}) = \nu(Ᵽ_{17}Ᵽ_{24}) = \nu(Ᵽ_{17}Ᵽ_{25}) = \nu(Ᵽ_{25}Ᵽ_{26}) = \nu(Ᵽ_{25}Ᵽ_{32}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad -\,(\mathbf{1})$$

$$\nu(Ᵽ_{10}Ᵽ_{11}) = \nu(Ᵽ_{10}Ᵽ_{18}) = \nu(Ᵽ_{11}Ᵽ_{12}) = \nu(Ᵽ_{11}Ᵽ_{19}) = \nu(Ᵽ_{12}Ᵽ_{13}) = \nu(Ᵽ_{12}Ᵽ_{20}) = \nu(Ᵽ_{13}Ᵽ_{14}) = \nu(Ᵽ_{13}Ᵽ_{21}) = \nu(Ᵽ_{14}Ᵽ_{15}) = \nu(Ᵽ_{14}Ᵽ_{22}) = \nu(Ᵽ_{15}Ᵽ_{16}) = \nu(Ᵽ_{15}Ᵽ_{23}) = \nu(Ᵽ_{16}Ᵽ_{24}) = \nu(Ᵽ_{18}Ᵽ_{19}) = \nu(Ᵽ_{18}Ᵽ_{26}) = \nu(Ᵽ_{19}Ᵽ_{20}) = \nu(Ᵽ_{19}Ᵽ_{27}) = \nu(Ᵽ_{20}Ᵽ_{21}) = \nu(Ᵽ_{20}Ᵽ_{28}) = \nu(Ᵽ_{21}Ᵽ_{22}) = \nu(Ᵽ_{21}Ᵽ_{29}) = \nu(Ᵽ_{22}Ᵽ_{23}) = \nu(Ᵽ_{22}Ᵽ_{30})$$

$$=\nu(Ᵽ_{23}Ᵽ_{24}) = \nu(Ᵽ_{23}Ᵽ_{31}) = \nu(Ᵽ_{1}Ᵽ_{2}) = \nu(Ᵽ_{1}Ᵽ_{8}) = \nu(Ᵽ_{2}Ᵽ_{3}) = \nu(Ᵽ_{2}Ᵽ_{10}) = \nu(Ᵽ_{2}Ᵽ_{30}) = \nu(Ᵽ_{3}Ᵽ_{4}) = \nu(Ᵽ_{3}Ᵽ_{11}) = \nu(Ᵽ_{3}Ᵽ_{31}) = \nu(Ᵽ_{4}Ᵽ_{5}) = \nu(Ᵽ_{4}Ᵽ_{12}) = \nu(Ᵽ_{4}Ᵽ_{32}) = \nu(Ᵽ_{5}Ᵽ_{6}) = \nu(Ᵽ_{5}Ᵽ_{13}) = \nu(Ᵽ_{5}Ᵽ_{25}) = \nu(Ᵽ_{6}Ᵽ_{7}) = \nu(Ᵽ_{6}Ᵽ_{14}) = \nu(Ᵽ_{6}Ᵽ_{26}) = \nu(Ᵽ_{7}Ᵽ_{8}) = \nu(Ᵽ_{7}Ᵽ_{15}) = \nu(Ᵽ_{7}Ᵽ_{27}) = \nu(Ᵽ_{8}Ᵽ_{16}) = \nu(Ᵽ_{8}Ᵽ_{28}) = \begin{pmatrix}0\\1\end{pmatrix} \qquad -(\mathbf{2})$$

$$\nu(Ᵽ_{24}Ᵽ_{32}) = \nu(Ᵽ_{26}Ᵽ_{27}) = \nu(Ᵽ_{27}Ᵽ_{28}) = \nu(Ᵽ_{28}Ᵽ_{29}) = \nu(Ᵽ_{29}Ᵽ_{30}) = \nu(Ᵽ_{30}Ᵽ_{31}) = \nu(Ᵽ_{31}Ᵽ_{32}) = \begin{pmatrix}1\\1\end{pmatrix} \qquad -(\mathbf{3})$$

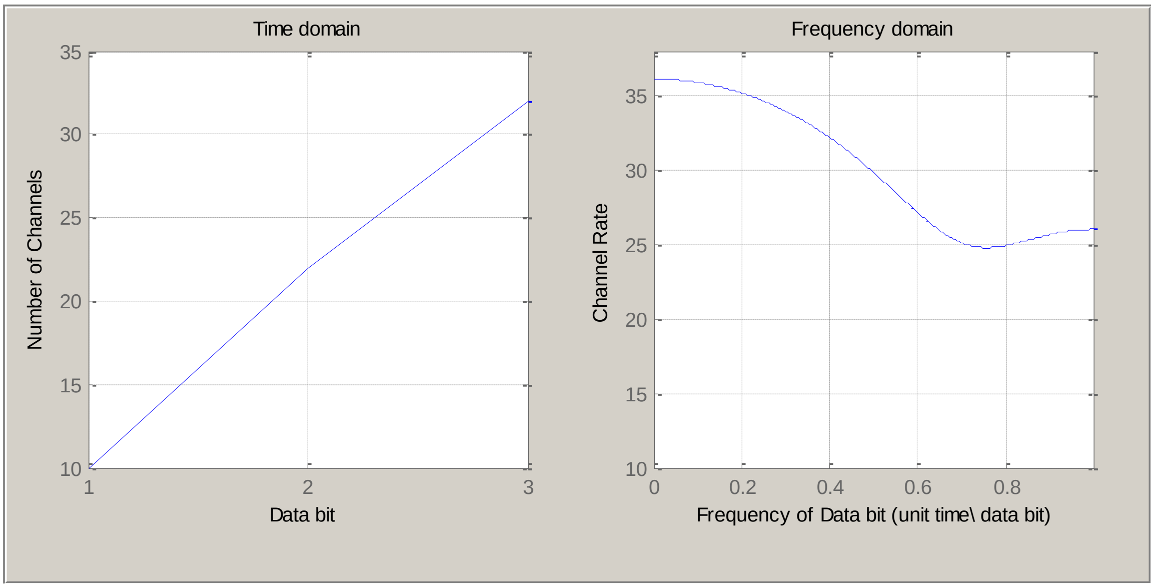


Figure 4.9: Channels with different data bits ($đ_1$, $đ_2$ and $đ_1 \oplus đ_2$). In time domain x-axis captions 1,2 and 3 represents $đ_1$, $đ_2$ and $đ_1 \oplus đ_2$.

Vector space generated in equations 1, 2 and 3 represent channels in RDT networks with data $đ_1$, $đ_2$ and $đ_1 \oplus đ_2$. Equation 1 contains 10 channels, which carry data $đ_1$. Similarly, equation 2 and 3 contains 22 and 32 channels, which carry data $đ_2$ and $đ_1 \oplus đ_2$. After evaluation of vector space it is observed that network coding is required on those nodes with matrix principles. Equation 3 states data channels with XOR value ($đ_1 \oplus đ_2$) of $đ_1$ and $đ_2$. So network coding is essential on nodes concerned in connectivity of channels as in equation 3. As equation 3 consists of 32 channels which involves 23 nodes ($Ᵽ_8$, $Ᵽ_{10}$ $Ᵽ_{16}$, $Ᵽ_{18}$ $Ᵽ_{24}$, and $Ᵽ_{25}$ $Ᵽ_{32}$), so network coding approach is implemented on these nodes.

4.3.2 *Network coding on MMT and MoT(Multi Mesh of Trees and Mesh of Trees)*

In this section, we have implemented the network coding on MMT and MoT. Both these networks have identical interblock connectivity [116, 126] (figure 4.10), therefore implementation of network coding for interblock connectivity of these networks will remain same.

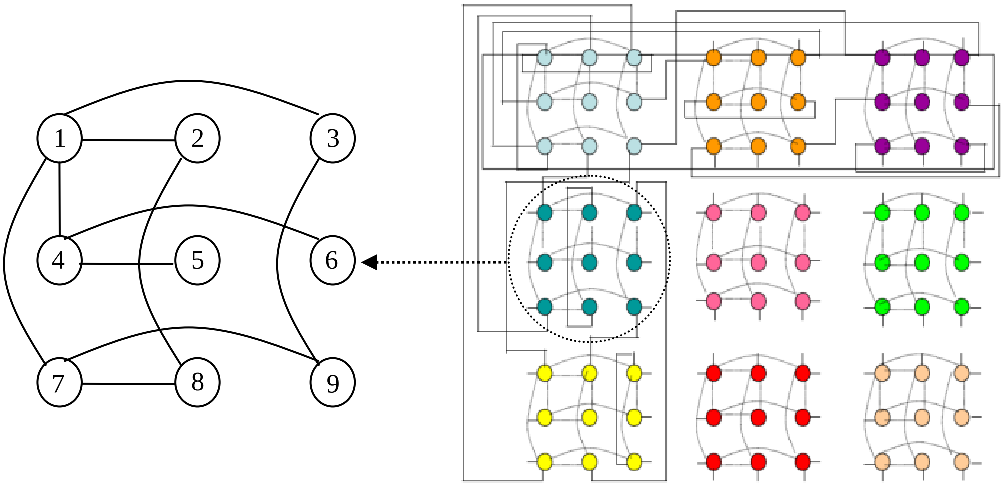


Figure 4.10: Figure in left shows MoT and one block of MMT network with 3×3 nodes. The figure in right shows MMT network with $3^2 \times 3^2$ nodes.

To implement network coding, it is necessary to estimate *maxflow* in these networks. We consider different size of these networks to optimize our approach and study network coding with more results. We have analyzed network coding approach on MMT and MoT with network size 16 (4×4) and 36 (6×6). Let us assume that the network size of a block of MMT and entire MoT is 4×4, i.e. 16 nodes in a block of MMT and MoT topology. For *maxflow*, classification of nodes with the different number of incoming is mandatory. Figure 4.11 shows different *maxflow* in MMT and MoT network. We have considered two data bits {$đ_1$, $đ_2$} with node 1 which it multicast to other nodes in the network. Channels {(1, 5) and (1, 9)} carries data $đ_1$ and channels {(1, 2) and (1, 3)} carries data $đ_2$.

Now, data bits ($đ_1$, $đ_2$) are multicast to other nodes in the network. Assuming node 1 is the active node [120], in first step, data $đ_1$ is sent to nodes (5, 9) and data $đ_2$ is sent to node (2, 3). In second step, $đ_1$ data is communicated to nodes (*6*, *7*, *10*, *11* and 13) and $đ_2$ data is communicated to (4, *6*, *7*, *10* and *11*). This illustrates that in second step of data communication in MMT and MoT network, nodes (*6*, *7*, *10*, and *11*) receives both đ_1 and đ_2 data from different source (5, 9) and (2, 3). It shows that, this problem can be solved using network coding. In other words, without network coding, it is impossible to multicast two bits đ_1 and đ_2 per unit time from source (4, 7) and (2, 3) to nodes (6, 7, 10 and 11) [1]. The *maxflow* in MMT and MoT is evaluated as in figure 4.12.

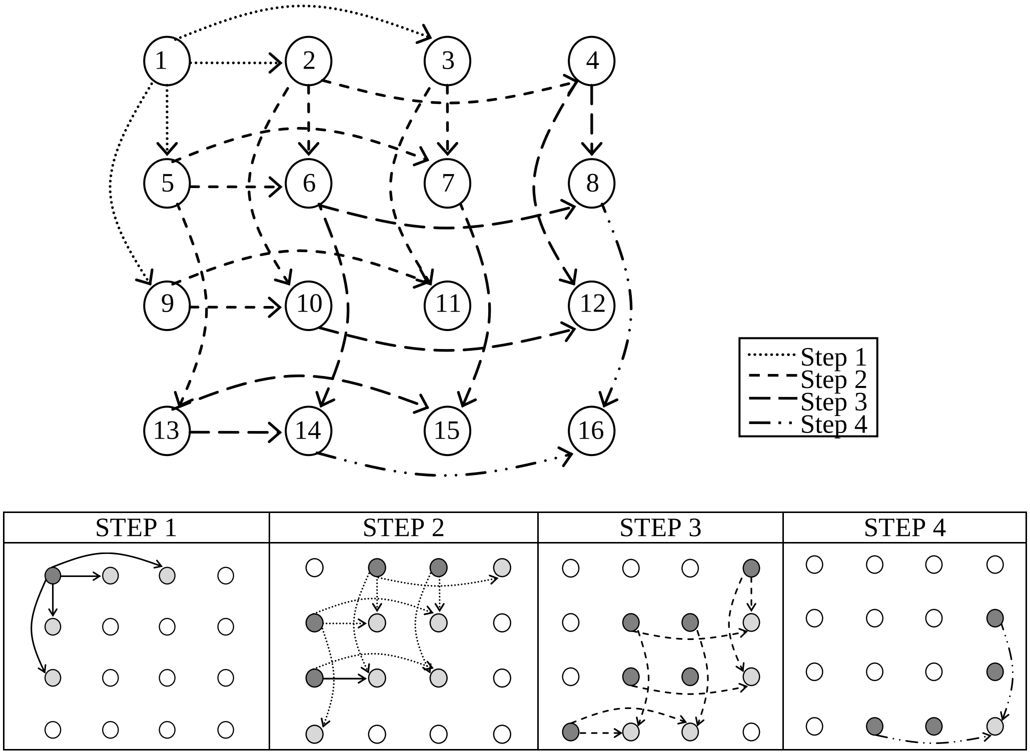


Figure 4.11: Figure shows MoT and one block of MMT network with 4×4 nodes. The flow of data in each step is shown with different line types. The table shows channel involved in communication in each step.

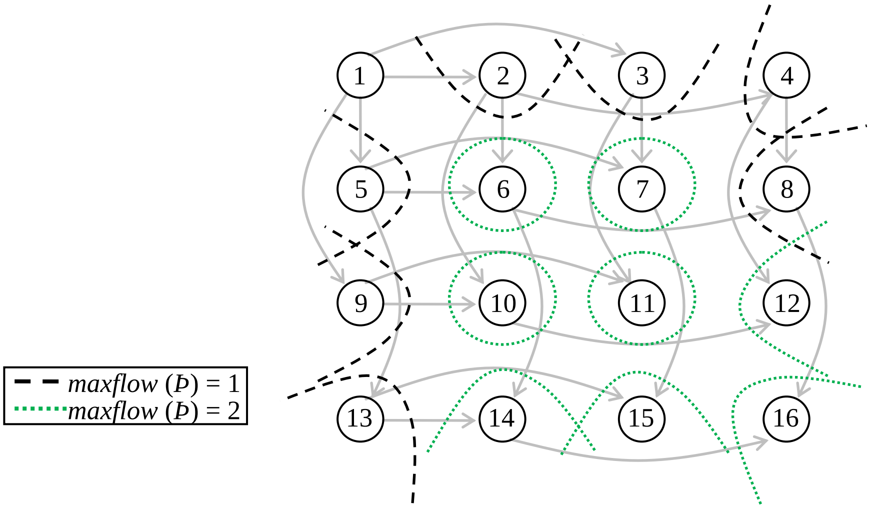


Figure 4.12: 4×4 MoT and one block of MMT network with different values of *maxflow* at different nodes.

In the above figure, MoT and one block of MMT network consists of two different *maxflow* on different nodes. If the statistics of this network is evaluated, seven nodes ($Ꝑ_{2}$, $Ꝑ_{3}$, $Ꝑ_{4}$, $Ꝑ_{5}$, $Ꝑ_{8}$, $Ꝑ_{9}$ and $Ꝑ_{13}$) have *maxflow*= 1; eight nodes ($Ꝑ_{6}$, $Ꝑ_{7}$, $Ꝑ_{10}$, $Ꝑ_{11}$, $Ꝑ_{12}$, $Ꝑ_{14}$, $Ꝑ_{15}$ and $Ꝑ_{16}$) have *maxflow* = 2. The statistics of *maxflow* in these networks is: as *maxflow* increases the number of nodes decreases, which results in similar trend as in RDT network. That is, when *maxflow* = 1, number of nodes = 7; when *maxflow* = 2, number of nodes = 8. This verifies possibility for implementation of network coding on parallel networks. Figure 4.13 shows that if the number of nodes increases then the possibilities of network coded nodes also increase. Using this statics the number of nodes in a huge network can easily be estimated.

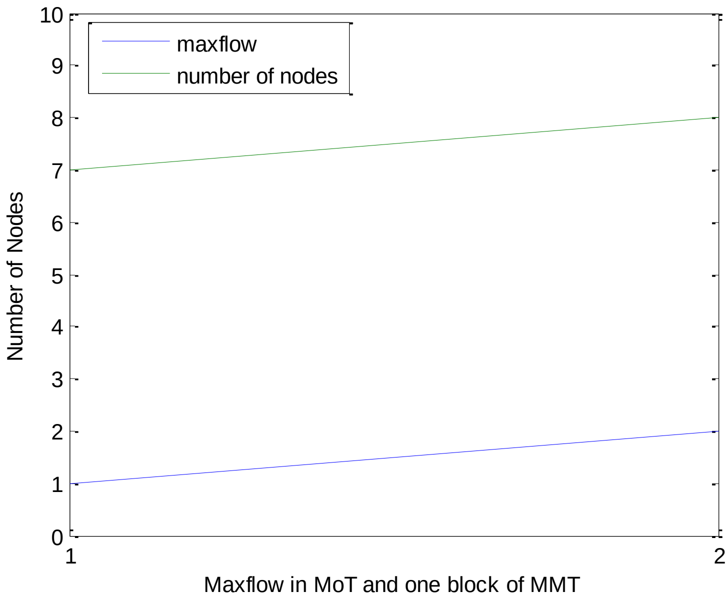


Figure 4.13: Different values of *maxflow* at different number of nodes in MoT and one block of MMT network for 4×4.

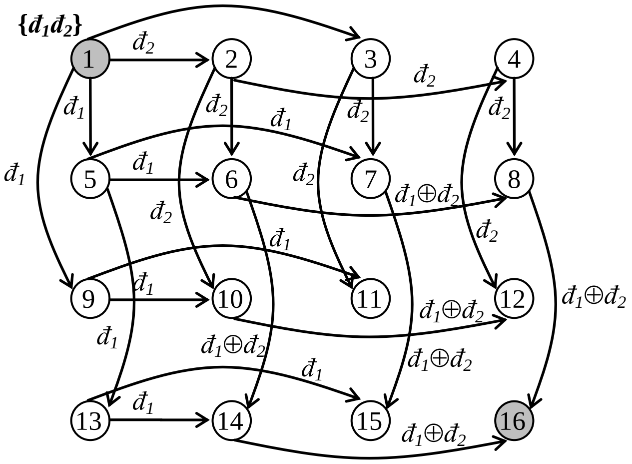


Figure 4.14: MOT and one block of MMT network with network coding. Source node 1 transfers data $đ_1đ_2$ to destination node 16.

Figure 4.14 shows the manner of data transfer from a source node to destination in MoT and MMT network. It shows the nodes where network coding is implemented. Nodes (1, 2, 3, 4, 5, 9 and 13) do not require coding as they receive one input. However, nodes (6, 7, 8, 10 and 14) receive multiple data so these nodes are coded. Remaining nodes (11, 12 and 15) are non-output nodes so network coding is not required on these nodes. The source node 1, transfers 2 bits of data ($đ_1$, $đ_2$) to destination node 16, which receive 1 bit data i.e. $đ_1 \oplus đ_2$. The received data is decoded to get the original information. Network coding has not only simplified the algorithm to transfer data but also reduced the size of data transfer to destination. After coding multiple data receiving nodes in MoT and MMT network, it can be formulated that network coding support to diminish the time complexity of multi-processing environments. Further, only network coding can make sensible parallel processing. Let us reconsider the *maxflow* parameter for 6×6 MoT and one block of MMT for analyzing network coding more evidently on these networks. Assuming that the network size of a block of MMT and entire MoT is 6×6, i.e. 36 nodes in a block of MMT and MoT topology. *Maxflow* in the network is based on classification of nodes with respect to different number of incoming to each node. Figure 4.15 shows different *maxflow* in MMT and MoT network. We considered same data set, which consist of two data bits {$đ_1$, $đ_2$} with node 1, which it multicast to other nodes in the network. Channels {(1, 7) and (1, 13)} carries data $đ_1$ and channels {(1, 2) and (1, 3)} carries data $đ_2$.

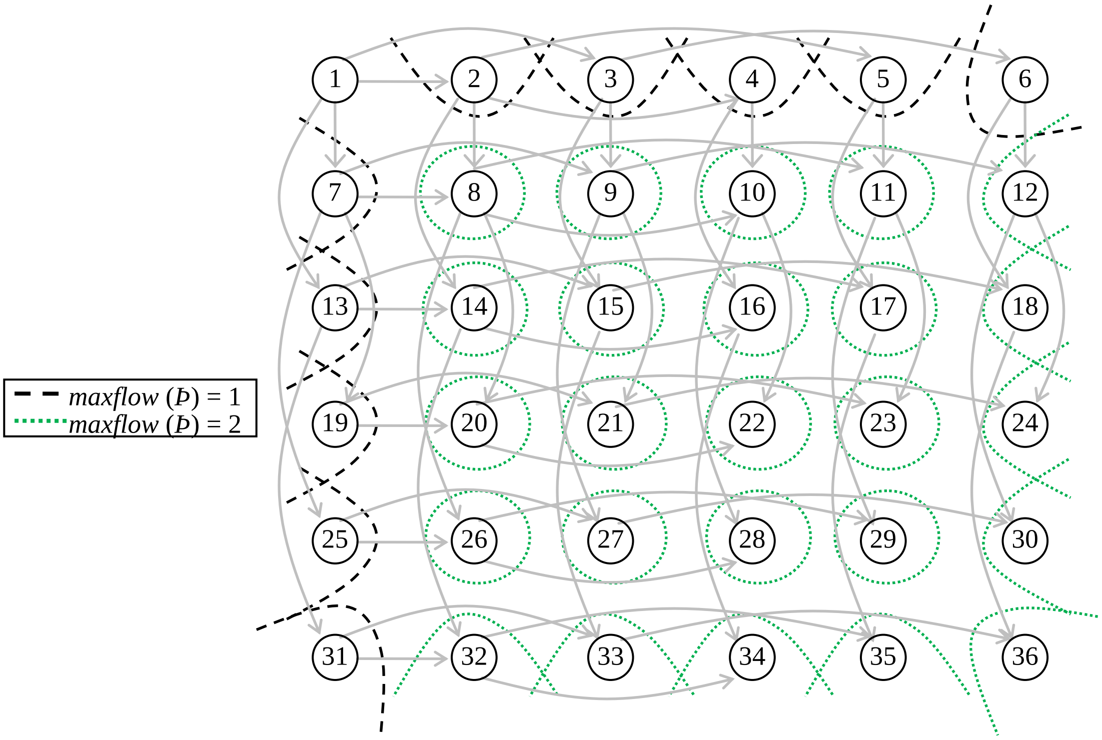


Figure 4.15: 6×6 MoT and one block of MMT network with different values of *maxflow* at different nodes.

According to figure 4.15, MoT and one block of MMT network consists of two different *maxflow* on different nodes. From 36 nodes, ten nodes ($Þ_{2}$, $Þ_{3}$, $Þ_{4}$, $Þ_{5}$, $Þ_{6}$, $Þ_{7}$, $Þ_{13}$, $Þ_{19}$, $Þ_{25}$ and $Þ_{31}$) have *maxflow* = 1; 25 nodes ($Þ_{8}$–$Þ_{12}$, $Þ_{14}$–$Þ_{18}$, $Þ_{20}$–$Þ_{24}$, $Þ_{26}$–$Þ_{30}$ and $Þ_{32}$–$Þ_{36}$) have *maxflow* = 2. Now as *maxflow* increases, the number of nodes decreases which results in similar trend as in RDT network. That is, when *maxflow* = 1, number of nodes = 10; when *maxflow* = 2, number of nodes = 25. The simulation result is shown in figure 4.16. Results of *maxflow* on MoT and MMT are accessed above and vector space is generated. The vector space for MoT and MMT network is used to achieve linear code multicast. We are interested in the channels carrying XOR of data bits ($đ_{1}$, $đ_{2}$) which is encoded using network coding. These channels are the source of complexity estimation in parallel networks. We have formulated the vector space generated using *maxflow* for both 4×4 and 6×6 network sizes.

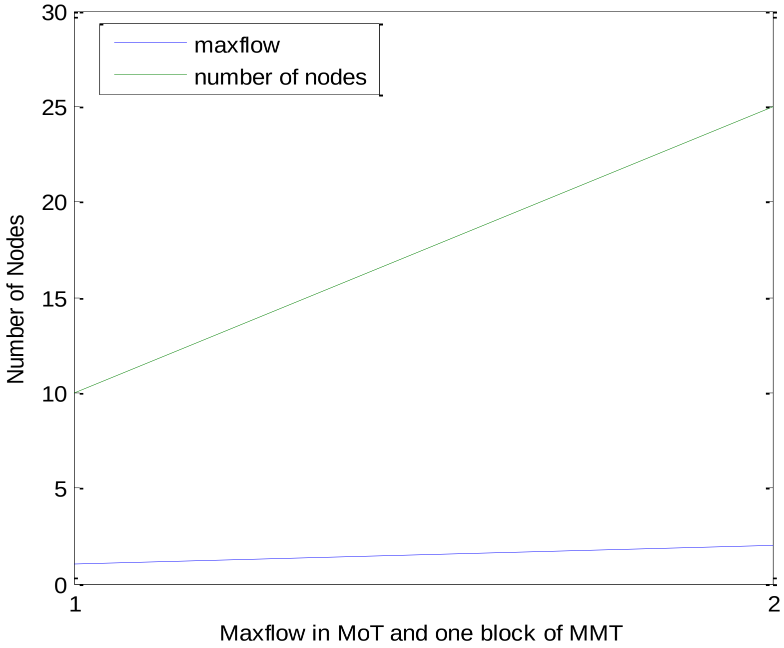


Figure 4.16: Different values of *maxflow* at different number of nodes in MoT and one block of MMT network for 6×6 size.

$$\nu(Þ_{1}Þ_{5}) = \nu(Þ_{1}Þ_{9}) = \nu(Þ_{5}Þ_{6}) = \nu(Þ_{5}Þ_{7}) = \nu(Þ_{5}Þ_{13}) = \nu(Þ_{9}Þ_{10}) = \nu(Þ_{9}Þ_{11}) = \nu(Þ_{13}Þ_{14}) = \nu(Þ_{13}Þ_{15}) = \begin{pmatrix}1\\0\end{pmatrix} \qquad -(\mathbf{4})$$

$$\nu(Þ_{1}Þ_{2}) = \nu(Þ_{1}Þ_{3}) = \nu(Þ_{2}Þ_{4}) = \nu(Þ_{2}Þ_{6}) = \nu(Þ_{2}Þ_{10}) = \nu(Þ_{3}Þ_{7}) = \nu(Þ_{3}Þ_{11}) = \nu(Þ_{4}Þ_{8}) = \nu(Þ_{4}Þ_{12}) = \begin{pmatrix}0\\1\end{pmatrix} \qquad -(\mathbf{5})$$

$$\nu(Þ_{6}Þ_{8}) = \nu(Þ_{6}Þ_{14}) = \nu(Þ_{7}Þ_{15}) = \nu(Þ_{8}Þ_{16}) = \nu(Þ_{10}Þ_{12}) = \nu(Þ_{14}Þ_{16}) = \begin{pmatrix}1\\1\end{pmatrix} \qquad -(\mathbf{6})$$

The above equations are the representation of vector space generated for MoT and one block MMT with network size 4×4. Equation 4 represents the channels which carry $đ_1$ data to other nodes of network. In the same way, equation 5 represents channels with $đ_2$ data. However, equation 6 consists of channels with XOR both $đ_1$, $đ_2$ i.e. $(đ_1 \oplus đ_2)$. The vector space for 6×6 network size is shown below. Equation 9 consists of 30 channels which involve 25 nodes ($Þ_{8}$, $Þ_{12}$, $Þ_{14}$ $Þ_{18}$, $Þ_{20}$ $Þ_{24}$, $Þ_{26}$ $Þ_{30}$, and $Þ_{32}$ $Þ_{36}$), so network coding approach is implemented on these 25 nodes from both networks. Channels with different data bits ($đ_1$, $đ_2$ and $đ_1 \oplus đ_2$) for both 4×4 and 6×6 network size is shown in figure 4.17.

$$\nu(Þ_1Þ_7) = \nu(Þ_1Þ_{13}) = \nu(Þ_7Þ_8) = \nu(Þ_7Þ_9) = \nu(Þ_7Þ_{19}) = \nu(Þ_7Þ_{25}) = \nu(Þ_{13}Þ_{14}) = \nu(Þ_{13}Þ_{15}) = \nu(Þ_{13}Þ_{31}) = \nu(Þ_{19}Þ_{20}) = \nu(Þ_{19}Þ_{21}) = \nu(Þ_{25}Þ_{26}) = \nu(Þ_{25}Þ_{27}) = \nu(Þ_{31}Þ_{32}) = \nu(Þ_{31}Þ_{33}) = \begin{pmatrix}1\\0\end{pmatrix} \qquad -\mathbf{(7)}$$

$$\nu(Þ_1Þ_2) = \nu(Þ_1Þ_3) = \nu(Þ_2Þ_4) = \nu(Þ_2Þ_5) = \nu(Þ_2Þ_8) = \nu(Þ_2Þ_{14}) = \nu(Þ_3Þ_6) = \nu(Þ_3Þ_9) = \nu(Þ_3Þ_{15}) = \nu(Þ_4Þ_{10}) = \nu(Þ_4Þ_{16}) = \nu(Þ_5Þ_{11}) = \nu(Þ_5Þ_{17}) = \nu(Þ_6Þ_{12}) = \nu(Þ_6Þ_{18}) = \begin{pmatrix}0\\1\end{pmatrix} \qquad -\mathbf{(8)}$$

$$\nu(Þ_8Þ_{10}) = \nu(Þ_8Þ_{11}) = \nu(Þ_8Þ_{20}) = \nu(Þ_8Þ_{26}) = \nu(Þ_9Þ_{12}) = \nu(Þ_9Þ_{21}) = \nu(Þ_9Þ_{27}) = \nu(Þ_{10}Þ_{22}) = \nu(Þ_{10}Þ_{28}) = \nu(Þ_{11}Þ_{23}) = \nu(Þ_{11}Þ_{29}) = \nu(Þ_{12}Þ_{24}) = \nu(Þ_{12}Þ_{30}) = \nu(Þ_{14}Þ_{16}) = \nu(Þ_{14}Þ_{17}) = \nu(Þ_{14}Þ_{32}) = \nu(Þ_{15}Þ_{18}) = \nu(Þ_{15}Þ_{33}) = \nu(Þ_{16}Þ_{34}) = \nu(Þ_{17}Þ_{35}) = \nu(Þ_{18}Þ_{36}) = \nu(Þ_{20}Þ_{22}) = \nu(Þ_{20}Þ_{23}) = \nu(Þ_{21}Þ_{24}) = \nu(Þ_{26}Þ_{28}) = \nu(Þ_{26}Þ_{29}) = \nu(Þ_{27}Þ_{30}) = \nu(Þ_{32}Þ_{34}) = \nu(Þ_{32}Þ_{35}) = \nu(Þ_{33}Þ_{36}) = \begin{pmatrix}1\\1\end{pmatrix} \qquad -\mathbf{(9)}$$

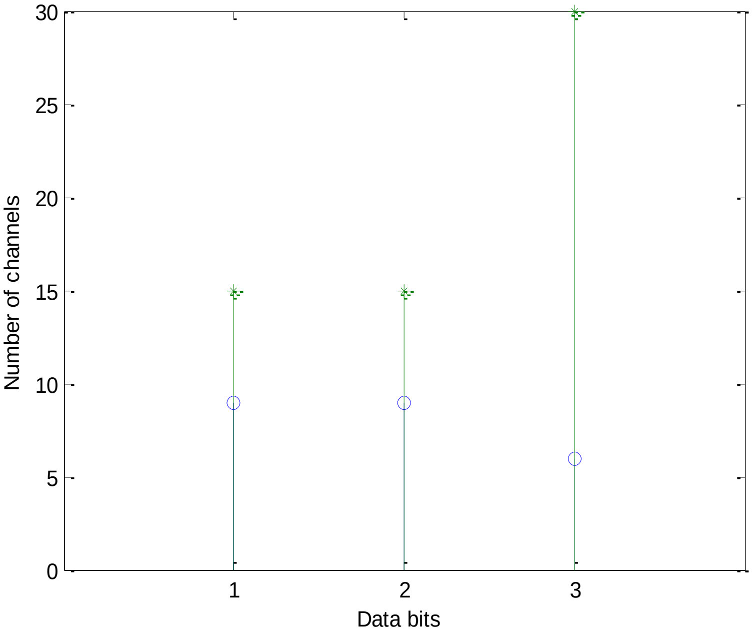


Figure 4.17: Channels with different data bits ($đ_1$, $đ_2$ and $đ_1 \oplus đ_2$). In time domain x-axis captions 1,2 and 3 represents $đ_1$, $đ_2$ and $đ_1 \oplus đ_2$. Both 4×4 and 6×6 network size are represented above.

4.3.3 *Network coding on Multi Mesh and 2D Mesh Network*

A $n \times n$ multi mesh network [118] is a combination of $n^2(n \times n)$ 2D mesh [119]. We have considered both networks for implementing network coding. We have not considered the inter block connectivity between n^2 2D mesh. Rather, we are using one 2D mesh to study the implementation results of network. A 2D mesh consists of n^2 nodes arranged in (*row*×*column*) format (figure 4.18). The diameter of this network is (n-1) with a constant number of channels and channel length. We have considered 4×4 and 6×6 network size to study the variation in results after implementing network coding. Further we initiated with 4×4 network size to find the faulty nodes, as network coding is utmost required on these nodes. For this, assuming that node 1 ($Þ_1$) broadcast data {$đ_1$, $đ_2$} to destination node 16 ($Þ_{16}$). Channel {1, 5} carry $đ_1$ and channel {1, 2} carry $đ_2$. Now, while communicating between different nodes with different data, the receiving nodes can receive either data from different input channels. Such different input receiving nodes are faulty nodes (figure 4.19). These nodes require network coding to eliminate this dilemma. In multi mesh and 2D mesh topology network coding principle is implemented to remove faulty nodes and accomplish data communication between nodes at unit time.

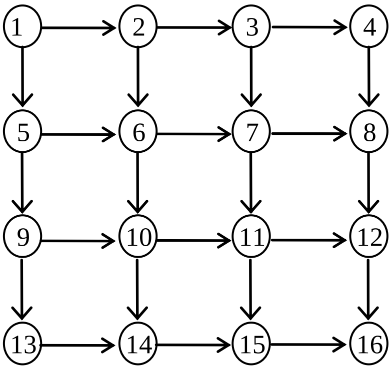


Figure 4.18: 2D Mesh and one block of Multi Mesh network.

For implementing network coding on the 2D mesh and one block of multi mesh it is necessary to find *maxflow* in these networks. *Maxflow* is calculated based on number of incomings to a node. In figure 4.19 we have identified the faulty node with more than one incoming. These non-source nodes with different *maxflow* are shown in figure 4.20 and these are listed in table 4.2.

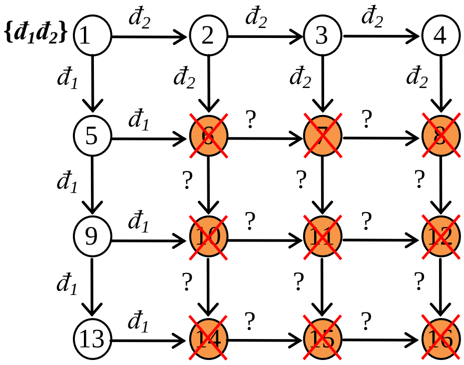


Figure 4.19: 2D Mesh and one block of Multi Mesh network with faulty nodes.

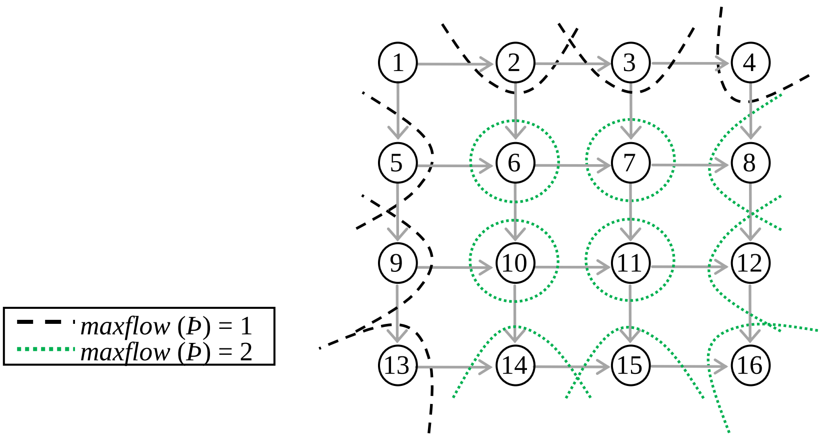


Figure 4.20: (4×4) 2D Mesh and one block of Multi Mesh network with different values of *maxflow* at different nodes.

Table 4.2: *Maxflow* in 2D Mesh and one block of Multi Mesh network.

Nodes Receiving Single Data Input	**Nodes Receiving Two Data Inputs**
maxflow (Þ_{2}) = 1	*maxflow* (Þ_{6}) = 2
maxflow (Þ_{3}) = 1	*maxflow* (Þ_{7}) = 2
maxflow (Þ_{4}) = 1	*maxflow* (Þ_{8}) = 2
maxflow (Þ_{5}) = 1	*maxflow* (Þ_{10}) = 2
maxflow (Þ_{9}) = 1	*maxflow* (Þ_{11}) = 2
maxflow (Þ_{13}) = 1	*maxflow* (Þ_{12}) = 2
	maxflow (Þ_{14}) = 2
	maxflow (Þ_{15}) = 2
	maxflow (Þ_{16}) = 2

The above figure 4.20 shows different *maxflow* on these networks. The tendency of *maxflow* in these networks evaluates that six nodes (Þ_{2}, Þ_{3}, Þ_{4}, Þ_{5}, Þ_{9} and Þ_{13}) have *maxflow* = 1; and nine node (Þ_{6}, Þ_{7}, Þ_{8}, Þ_{10}, Þ_{11}, Þ_{12}, Þ_{14}, Þ_{15} and Þ_{16}) have *maxflow*= 2. So, network coding is required for nodes having *maxflow* = 2, where more than one input is received. Calculation of *maxflow* is required to identify nodes having a different number of input channels. The simulation result of this prospect is given in figure 4.21.

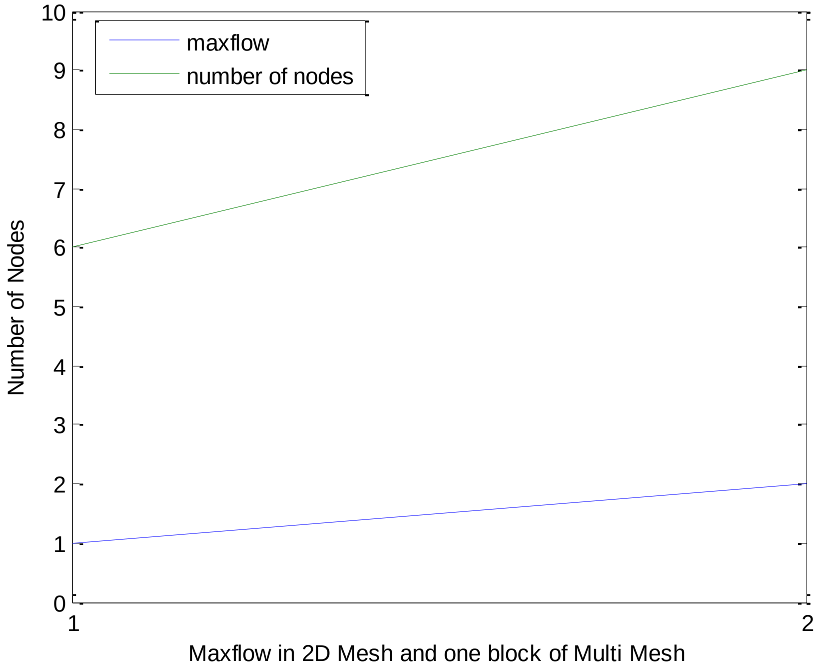


Figure 4.21: Different values of *maxflow* at different number of nodes in 2D Mesh and one block of Multi Mesh network for 4×4 size.

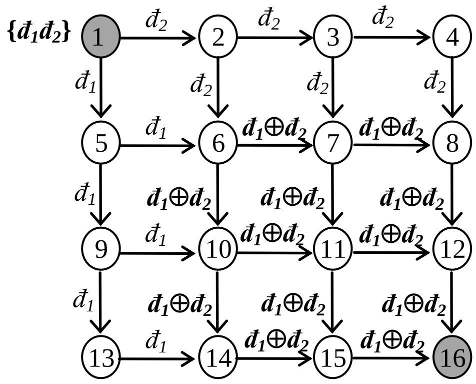


Figure 4.22: 2D Mesh and one block of Multi Mesh network with network coding. Source node 1 transfers data $đ_1đ_2$ to destination node 16.

Figure 4.22 shows the data transfer in the 2D mesh and multi mesh network from a source node to destination. Nodes (1, 2, 3, 4, 5, 9 and 13) do not require coding as they receive one input. However, nodes (6, 7, 8, 10, 11, 12, 14, 15 and 16) receive multiple data so these nodes are coded. The source node 1, transfers 2 bits of data ($đ_1$, $đ_2$) to destination node 16, which receive 1 bit data, i.e. $đ_1 \oplus đ_2$. The received data is decoded to get the original information.

Further we reconsidered the *maxflow* parameter for (6×6) 2D mesh and multi mesh network for examining network coding. Figure 4.23 shows different *maxflow* in 2D mesh and multi mesh network. We have considered same data set consisting of two data bits {$đ_1$, $đ_2$} with node 1 which it multicast to other nodes in the network.

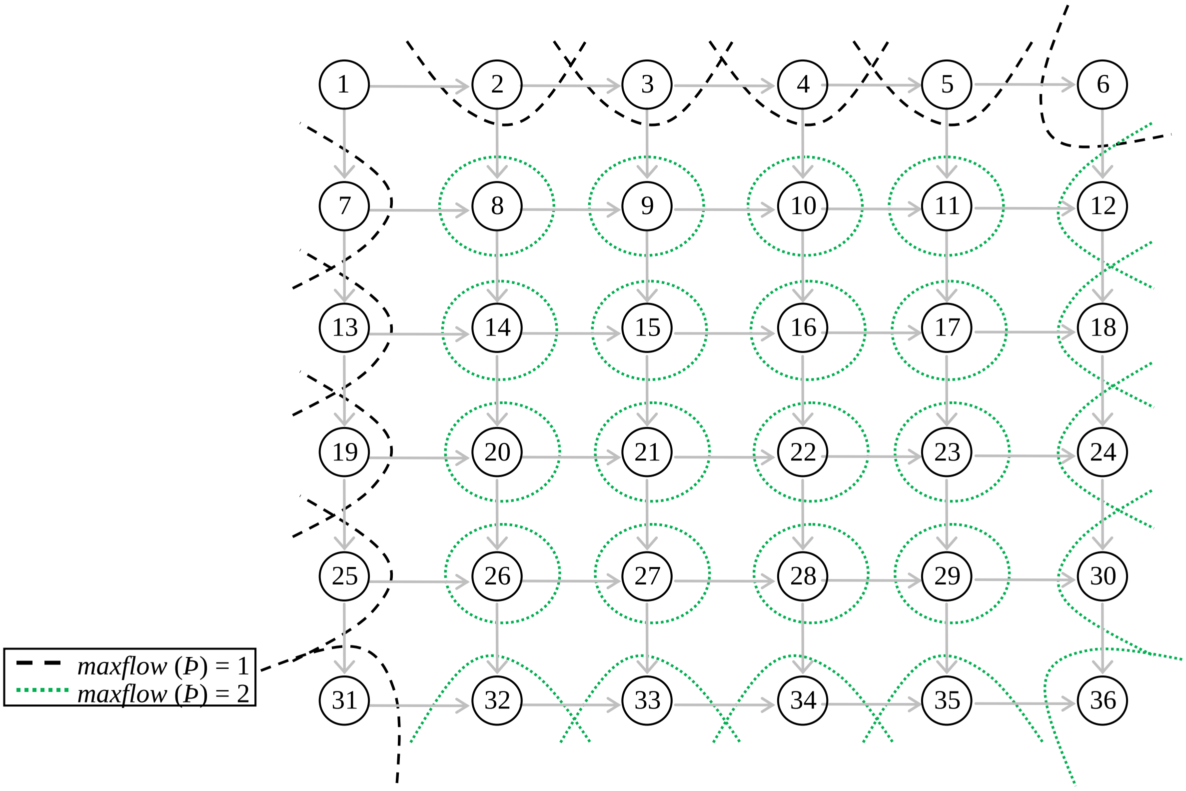


Figure 4.23: (6×6) 2D Mesh and one block of Multi Mesh network with different values of *maxflow* at different nodes.

The above figure illustrates that 2D mesh and one block of multi mesh network consist two different *maxflow* on different nodes. Out of 36 nodes, ten nodes ($Þ_2$, $Þ_3$, $Þ_4$, $Þ_5$, $Þ_6$, $Þ_7$, $Þ_{13}$, $Þ_{19}$, $Þ_{25}$ and $Þ_{31}$) have *maxflow*= 1; 25 nodes ($Þ_8$–$Þ_{12}$, $Þ_{14}$–$Þ_{18}$, $Þ_{20}$–$Þ_{24}$, $Þ_{26}$–$Þ_{30}$ and $Þ_{32}$–$Þ_{36}$) have *maxflow* = 2. Now, *maxflow* in these networks of 36 nodes increases, which results in similar trend as in RDT, MMT and MoT networks. That is, when *maxflow* = 1, number of nodes = 10; when *maxflow* = 2, number of nodes = 26. The simulation result of this observation on MoT and MMT is shown in figure 4.24.

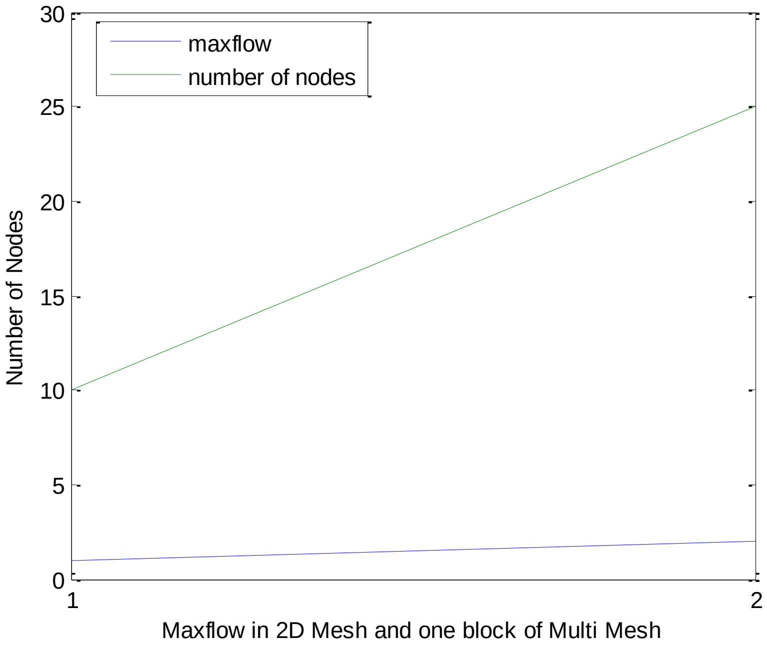


Figure 4.24: Different values of *maxflow* at different number of nodes in 2D Mesh and one block of Multi Mesh network for 6×6 size.

Outcome of *maxflow* on the 2D mesh and one block of multi mesh is reviewed above and vector space is generated. The vector space is used to achieve linear code multicast on these networks. We are concerned about the channels which carry an XOR of data bits ($đ_1$, $đ_2$) which necessitate network coding. These channels are the source of complexity estimation in parallel networks. Now we formulate the vector space generated using *maxflow* in these networks for both 4×4 and 6×6 network size.

$$\nu(Ᵽ_1Ᵽ_5) = \nu(Ᵽ_5Ᵽ_6) = \nu(Ᵽ_5Ᵽ_9) = \nu(Ᵽ_9Ᵽ_{10}) = \nu(Ᵽ_9Ᵽ_{13}) = \nu(Ᵽ_{13}Ᵽ_{14}) = \begin{pmatrix}1\\0\end{pmatrix} \qquad -\mathbf{(10)}$$

$$\nu(Ᵽ_1Ᵽ_2) = \nu(Ᵽ_2Ᵽ_6) = \nu(Ᵽ_2Ᵽ_3) = \nu(Ᵽ_3Ᵽ_4) = \nu(Ᵽ_3Ᵽ_7) = \nu(Ᵽ_4Ᵽ_8) = \begin{pmatrix}0\\1\end{pmatrix} \qquad -\mathbf{(11)}$$

$$\nu(Ᵽ_6Ᵽ_7) = \nu(Ᵽ_6Ᵽ_{10}) = \nu(Ᵽ_7Ᵽ_8) = \nu(Ᵽ_7Ᵽ_{11}) = \nu(Ᵽ_8Ᵽ_{12}) = \nu(Ᵽ_{10}Ᵽ_{11}) = \nu(Ᵽ_{10}Ᵽ_{14}) = \nu(Ᵽ_{11}Ᵽ_{12}) = \nu(Ᵽ_{11}Ᵽ_{15}) = \nu(Ᵽ_{12}Ᵽ_{16}) = \nu(Ᵽ_{14}Ᵽ_{15}) = \nu(Ᵽ_{15}Ᵽ_{16}) = \begin{pmatrix}1\\1\end{pmatrix} \qquad -\mathbf{(12)}$$

The above equations are representations of vector space generated for 2D mesh and one block of multi mesh with network size 4×4. Equation 10 represents the channels which carry $đ_1$ data and equation 11 represent channels with $đ_2$ data to other nodes of network. Equation 12 represents channels with XOR of both $đ_1$, $đ_2$ i.e. ($đ_1 \oplus đ_2$). The vector space for 6×6 network size is shown below. Equation 15 consists of 40 channels which involve 25 nodes ($Ᵽ_8$ $Ᵽ_{12}$, $Ᵽ_{14}$ $Ᵽ_{18}$, $Ᵽ_{20}$ $Ᵽ_{24}$, $Ᵽ_{26}$ $Ᵽ_{30}$, and $Ᵽ_{32}$ $Ᵽ_{36}$), so network coding approach is implemented on these 25 nodes. Channels with different data bits for both (4×4) and (6×6) network size are represented in figure 4.25.

$$\nu(Ᵽ_1Ᵽ_7) = \nu(Ᵽ_7Ᵽ_8) = \nu(Ᵽ_7Ᵽ_{13}) = \nu(Ᵽ_{13}Ᵽ_{14}) = \nu(Ᵽ_{13}Ᵽ_{19}) = \nu(Ᵽ_{19}Ᵽ_{20}) = \nu(Ᵽ_{19}Ᵽ_{25}) = \nu(Ᵽ_{25}Ᵽ_{26}) = \nu(Ᵽ_{25}Ᵽ_{31}) = \nu(Ᵽ_{31}Ᵽ_{32}) = \begin{pmatrix}1\\0\end{pmatrix} \qquad -(\mathbf{13})$$

$$\nu(Ᵽ_1Ᵽ_2) = \nu(Ᵽ_2Ᵽ_3) = \nu(Ᵽ_2Ᵽ_8) = \nu(Ᵽ_3Ᵽ_4) = \nu(Ᵽ_3Ᵽ_9) = \nu(Ᵽ_4Ᵽ_5) = \nu(Ᵽ_4Ᵽ_{10}) = \nu(Ᵽ_5Ᵽ_6) = \nu(Ᵽ_5Ᵽ_{11}) = \nu(Ᵽ_6Ᵽ_{12}) = \begin{pmatrix}0\\1\end{pmatrix} \qquad -(\mathbf{14})$$

$$\nu(Ᵽ_8Ᵽ_9) = \nu(Ᵽ_8Ᵽ_{14}) = \nu(Ᵽ_9Ᵽ_{10}) = \nu(Ᵽ_9Ᵽ_{15}) = \nu(Ᵽ_{10}Ᵽ_{11}) = \nu(Ᵽ_{10}Ᵽ_{16}) = \nu(Ᵽ_{11}Ᵽ_{12}) = \nu(Ᵽ_{11}Ᵽ_{17}) = \nu(Ᵽ_{12}Ᵽ_{18}) = \nu(Ᵽ_{14}Ᵽ_{15}) = \nu(Ᵽ_{14}Ᵽ_{20}) = \nu(Ᵽ_{15}Ᵽ_{16}) = \nu(Ᵽ_{15}Ᵽ_{21}) = \nu(Ᵽ_{16}Ᵽ_{17}) = \nu(Ᵽ_{16}Ᵽ_{22}) = \nu(Ᵽ_{17}Ᵽ_{18}) = \nu(Ᵽ_{17}Ᵽ_{23}) = \nu(Ᵽ_{18}Ᵽ_{24}) = \nu(Ᵽ_{20}Ᵽ_{21}) = \nu(Ᵽ_{20}Ᵽ_{26}) = \nu(Ᵽ_{21}Ᵽ_{22}) = \nu(Ᵽ_{21}Ᵽ_{27}) = \nu(Ᵽ_{22}Ᵽ_{23}) = \nu(Ᵽ_{22}Ᵽ_{28}) = \nu(Ᵽ_{23}Ᵽ_{24}) = \nu(Ᵽ_{23}Ᵽ_{29}) = \nu(Ᵽ_{24}Ᵽ_{30}) = \nu(Ᵽ_{26}Ᵽ_{27}) = \nu(Ᵽ_{26}Ᵽ_{32}) = \nu(Ᵽ_{27}Ᵽ_{28}) = \nu(Ᵽ_{27}Ᵽ_{33}) = \nu(Ᵽ_{28}Ᵽ_{29}) = \nu(Ᵽ_{28}Ᵽ_{34}) = \nu(Ᵽ_{29}Ᵽ_{30}) = \nu(Ᵽ_{29}Ᵽ_{35}) = \nu(Ᵽ_{30}Ᵽ_{36}) = \nu(Ᵽ_{32}Ᵽ_{33}) = \nu(Ᵽ_{33}Ᵽ_{34}) = \nu(Ᵽ_{34}Ᵽ_{35}) = \nu(Ᵽ_{35}Ᵽ_{36}) = \begin{pmatrix}1\\1\end{pmatrix} \qquad -(\mathbf{15})$$

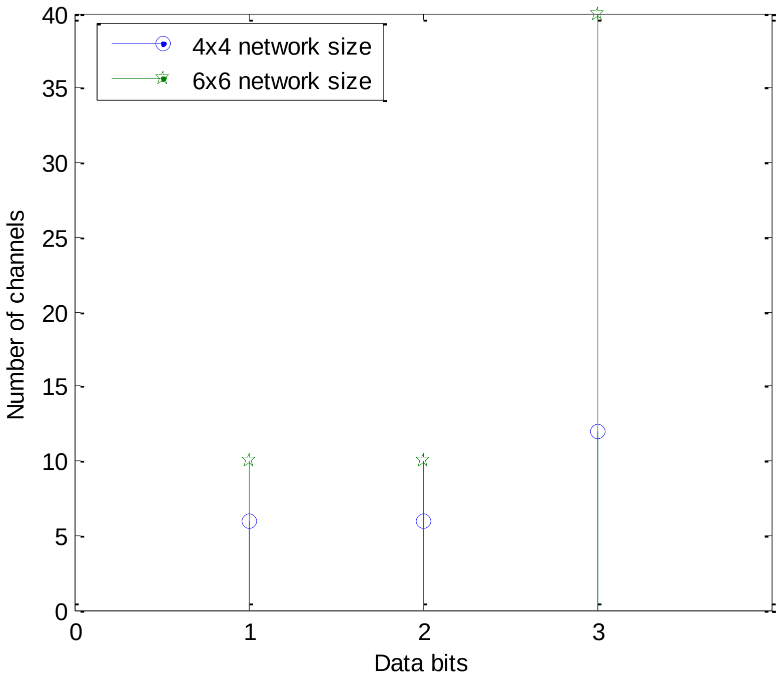


Figure 4.25: Channels with different data bits ($đ_1$, $đ_2$and $đ_1 \oplus đ_2$). In time domain x-axis captions 1, 2 and 3 represents $đ_1$, $đ_2$and $đ_1 \oplus đ_2$. Both (4×4) and (6×6) 2D Mesh and one block of MMT network size are represented above.

4.4 *Benefits of Linear Network Coding on Parallel Networks*

Network coding suggests considerable capacity gains for networks with special structures [127]. Advantages offered by network coding for different network scenarios are different. For parallel architectures, network coding offers some generic solutions to a set of problems. In this section, we considered three generic solutions to the above specified parallel architectures (Recursive Diagonal Torus (RDT), Multi Mesh of Trees (MMT), Mesh of Trees (MoT), Multi Mesh (MM) and 2D-Mesh networks) in which linear coding is particularly useful.

4.4.1 *Removal of Faulty Nodes*

In parallel networks, large numbers of nodes are involved and all nodes act as sender and receiver at different part of communication. Several nodes in this communication may receive diverse data from different nodes. However, a node can only receive one bit data at a particular time. So, either the information sent from multiple sources is lost or incomplete information is received by the node. Consider the problem of sending two bits of data from a source node to destination node in RDT network (figure 4.5). For simplicity, we analyzed only one node (Þ_{10}) which receive two diverse data (đ_1 and đ_2) from two channels. Now, either đ_1 or đ_2 can be received or complete information is lost. This node is now acting as a faulty node. This setback can only be removed by using network coding. When network coding is implemented this faulty node will XOR the data (đ_1 and đ_2) to ($\text{đ}_1 \oplus \text{đ}_2$). So the chance of information failure at this node is reduced. Now this node will further transfer XOR data to other connected nodes and vice versa. To identify the faulty nodes in a parallel network, *maxflow* is calculated.

Evaluation of *maxflow* not only identifies faulty nodes but also analyze number of incomings to this node. Based on the level of *maxflow*, network coding is implemented. According to previous section in these parallel networks, network coding is suggested based on the levels of *maxflow* evaluated. While analyzing *maxflow* a generic trend is originated. This trend suggests that as the number of node increases, *maxflow* decreases. Figure 4.26 shows the trend in these parallel networks.

4.4.2 *Reduced information size*

Another scenario in which network coding can be advantageous, is reduced information size. We have compared communication without and with network coding, in which the size of information increases based on the input received by the nodes. In parallel networks a core criterion is, to reduce the size of information flow so that communication complexity is reduced. This setback makes practical implementation of parallel networks unfeasible.

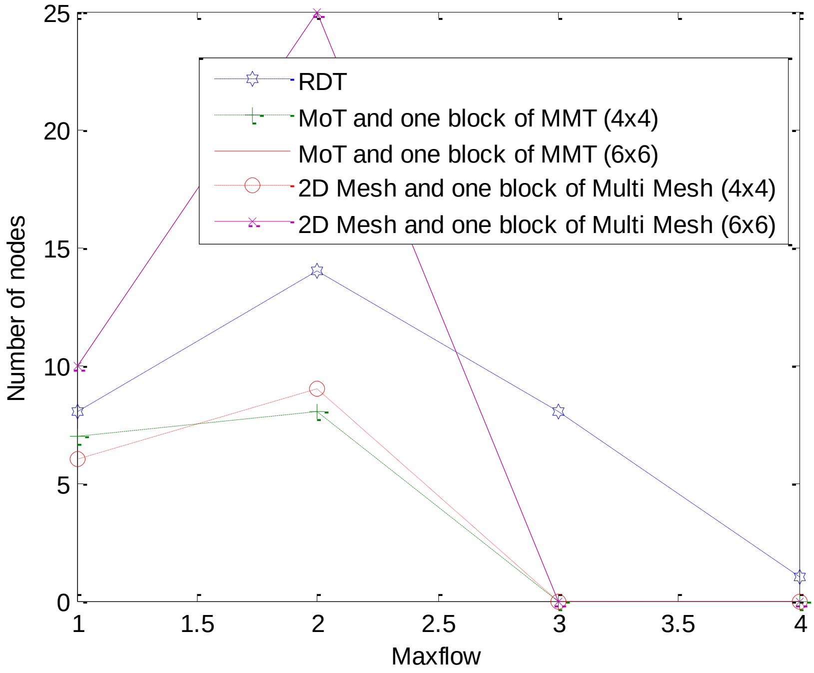


Figure 4.26: Trend of *maxflow* in above parallel networks (Recursive Diagonal Torus (RDT), Multi Mesh of Trees (MMT), Mesh of Trees (MoT), Multi Mesh (MM) and 2D-Mesh networks).

We have applied network coding with random communication with the following parameters: number of nodes n = 4, number of source = 1, number of receiver = $(n \times n) - 1$, number of destination = 1, steps of communication involved = 6. To distinct approach with and without network coding, assume that each node consist of different one bit data size. Let us examine communication in this network using both approaches:

4.4.2.1 *Communication without network coding*

In first step, the source node transmits data to the immediate connected node (see figure 4.27). The data size at receiving nodes will become two bits each. These nodes will act as sender node for step 2. In step 2, two receiving node get data of size three bits i.e., (2+1) and one get data size five i.e., (2+2+1). Similarly, till step 5 the data size will become thirty four bits of each node. The destination node receives data of size sixty eight with its previous data of size one bit; combined buffer size of this node is sixty nine. The simulation based on the size of data in each step without network coding is shown in figure 4.28.

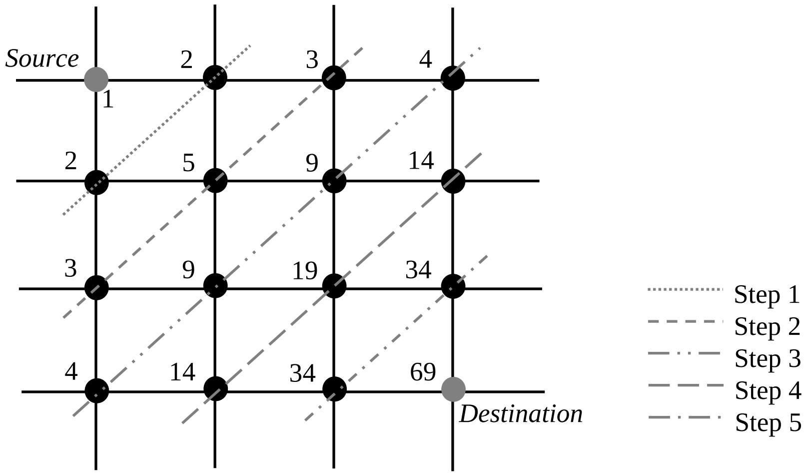


Figure 4.27: Steps of communication without network coding.

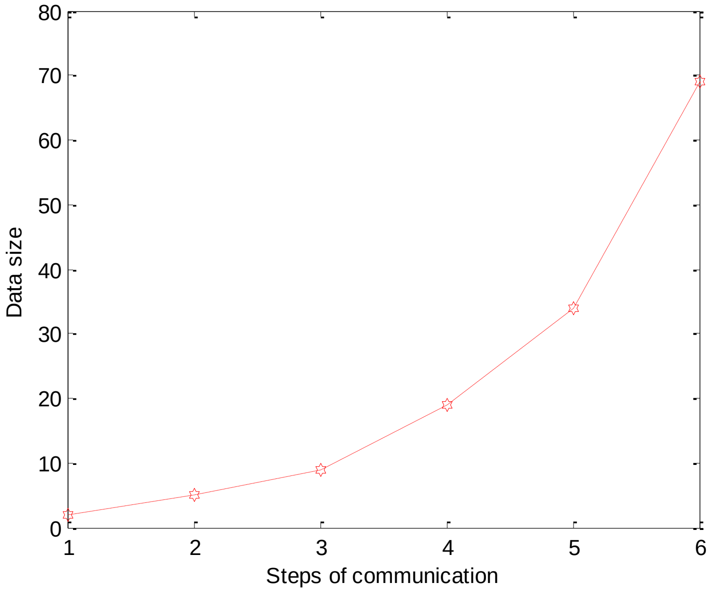


Figure 4.28: Communication in 2D mesh network without network coding.

Figure 4.27 shows that without using the network coding in 2D mesh the data size in each step grows exponentially. This is a limitation of parallel architecture, which can be reduced using network coding. These simulations do not attempt to quantify accurately the differences in performance and overhead of linear network coding with parallel networks, but are useful as a preliminary suggestion.

4.4.2.2 *Communication with network coding.*

We have considered above example of 2D mesh architecture and evaluated results by implementing network coding approach. The parameters and constraints are assumed similar as stated above. The steps of communication in the 2D mesh network will also remain unchanged. So the communication involves six steps from source to destination (figure 4.29). The source node transfers data to immediate connected nodes, the receiving node encodes the data received. Now this XOR data is one bit size and is transferred further to other nodes. Similar encoding is performed at other nodes in step 2. This process is performed until the complete information is received at destination node which will finally hold one bit information in the buffer of the destination nodes.

Irrespective of time complexity involved at each node to encode data, the size of data will remain same. Due to encoding at each node the data is XOR to form single information. This will eliminate the problem of excess data size associated with parallel architectures at each node buffer. Network coding provides an alternative to reduce buffer size of each node in parallel networks. While communication in 2D mesh network with network coding, the data size remains constant. This is a revolutionary advantage in the field of parallel communication, which suggests a better technique to implement high-scale parallel architecture.

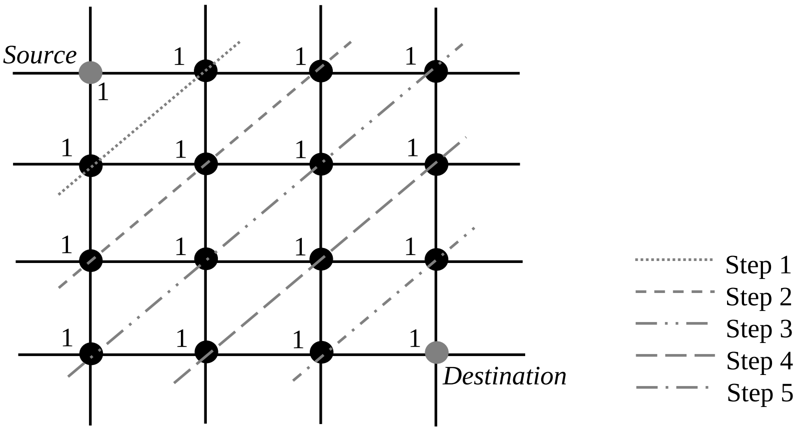


Figure 4.29: Steps of communication in 2D mesh network with network coding.

4.4.3 *Reduced algorithmic time complexity*

In previous part of this section, we have shown that using network coding; the buffer size of each node in a parallel topology can be reduced exponentially.

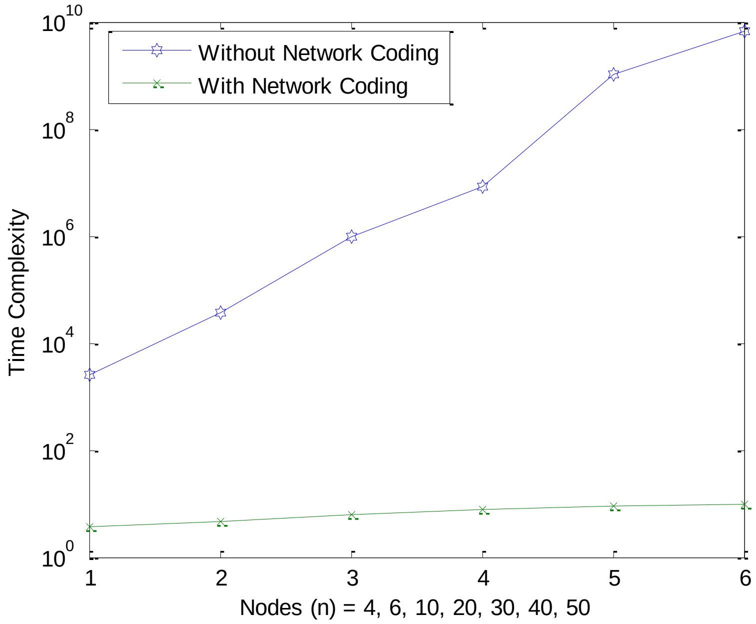


Figure 4.30: Time complexity variation for different network size (x-axis) with and without network coding.

Now, due to this reduced data size the complexity of data communication between different nodes is reduced. We have considered the previous example of 2D mesh network (figure 4.27), where the communication is accomplished in six steps. Let us assume that each data element is of size n. After step 1, time involved to communicate data is $logn$. The complexity of communication after step 2 is $nlogn$. Similarly, after step 6 the time complexity will become n^5logn. This is the complexity to communicate data from source to destination in six steps. Now, by using network coding it is reduced to $6logn$. At each step the data size remains constant ($logn$) so it is $6logn$ after entire communication. This shows a vast difference in terms of time complexity in parallel networks (shown in figure 4.30). This approach resolves three inter-linked problems of parallel communication.

4.5 *Chapter Outline*

This chapter presents linear network coding on parallel architecture. This multisource multicast parallel networks approach shows that: faulty nodes, information size and time complexity of communication decreases with code length. Further this approach also achieves capacity asymptotically as given by max-flow min-cut bound. As a result a generic coding methodology for different parallel networks is proposed. By means of different network scenarios it is shown that linear network coding effectively reduces the probability of errors. Finally, it is shown that LCM−PA benefits over routing approaches and the robustness of linear network coding offers a significant advantage in practical exploitation of parallel architectures.

CHAPTER 5

NETWORK CODED CONTOUR APPROACH TO BROADCAST IN PARALLEL ARCHITECTURES

Parallel communication suffers from a variety of distinctive problems such as low information rate, high space requirement and high data size. Recently, network coding evolved as an approach which can attain information flow in the network. In previous chapters, it is proved that network coding approach provides efficient broadcasting scheme for parallel networks. This chapter considers the problem of high data size in parallel networks and have used 2D-Mesh network to study this problem. By using the network coding approach in a contour manner in parallel networks, it is proved that this approach reduces the data size at each communication step by performing coding on the data set. Finally, an approach of network coding in parallel communication is proposed which reduces the storage requirement at each node and thus solves the problem of high data size to communicate with 2D-Mesh network.

5.1 *XOR in Parallel Architectures*

Parallel algorithms aspires optimally utilize processors of parallel architectures. As involvement of processors increases computation complexity and it needs increasingly high storage at each transmission. We have shown that network coding provides an advantage over store and forward mechanism, so it is interesting to know whether this approach is prolific for parallel communication (broadcasting and multicasting). Network coding is capable of utilizing the network capacity and provides polynomial-time solution for broadcast, which is termed as a special case of multicast [1, 69, 120-123]. We have implemented the perception of linear network coding over parallel architecture to provide energy-efficient broadcasting. This reduces the computation at each node receiving multiple data and limits the requirements of high storage at each node during such communication. Another problem with parallel communication is non-uniform traffic [128-132]. Parallel network consists of several processing nodes and each node is sending and receiving information at different time gaps.

So it turns out to be hard to utilize the network capacity with least time complexity. We discussed this problem by shaping traffic of parallel communication by using the network coding principles [133, 134]. Lastly, we resolved the problem to distinguish between more than one data received at any node.

Let us consider a example of the 2D-Mesh network of size 4 × 4 (see figure 5.1), where aim is to send information of all nodes to destination node (node 16). It achieves this aim by initiating the active processor [120] (node 1) and then combines remaining nodes (see figure 5.2). In figure 5.1, step 1, 2 … 5 shows participating nodes in respective steps. At each step the data size increases with the increase in the number of nodes. Let us explain the computing scenario in details. Node 1 sends data to node 2 and 5. Both 2 and 5 either clubs the data received and send them to further connected nodes (i.e., nodes 3, 6, and 9 which increases the computation time as the data size at node 2 and 5 increases) or can store either of the data and forward to other nodes 3, 6 and 9 (which increases the communication time). At each node receiving multiple data, the computation time and the requirement of storage also increases for each transmission. This approach of communication makes parallel architectures more complex and unfeasible in practical implementation. As the computation and communication time is increasing exponentially therefore, energy required to perform broadcast is high. These networks can become more efficient and feasible when problem of computation and communication reduces. We have reduced both problems by suggesting energy-efficient broadcasting using the network coding for parallel networks.

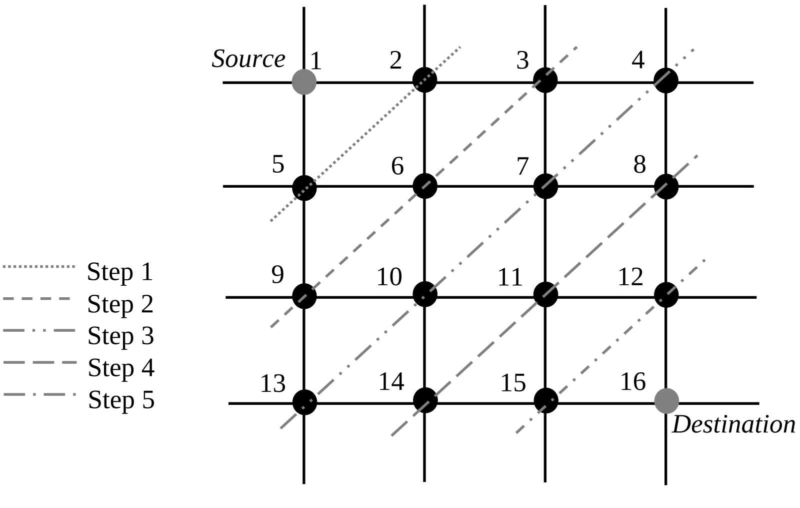


Figure 5.1: 2D-Mesh Architecture.

5.2 *Efficient Broadcasting using Network Coding*

The aim of this section is to reduce the computation and communication time required to send data between different nodes of parallel networks. For the understandability of readers, we considered 2D-Mesh architecture of $n \times n$ dimension. All the nodes in this architecture are sources and receivers. Each node in 2D-Mesh can broadcast information to its closest connected neighbouring nodes. So, communication in 2D-Mesh is like square grid (see figure 5.3). Based on this communication manner, data from node 9, 14, 16 and 21 will reach node 15. This means that after every transmission, data of four nodes reaches to the center node. The data is then XOR to result in one data, i.e., after receiving data from 9, 14, 16 and 21, the information at the center node 15 will be; $y_{15} = g_9 x_9 + g_{14} x_{14} + g_{16} x_{16} + g_{21} x_{21}$ where, y is the information field and $y_{15} \in y$ and y is associated with an encoding vector g, where $\{g_9, g_{14}, g_{16}, g_{21}\} \in g$ and x_i is the data symbol from source node i. So, according to network coding principle [1], the decoding matrix at node 15 will be;

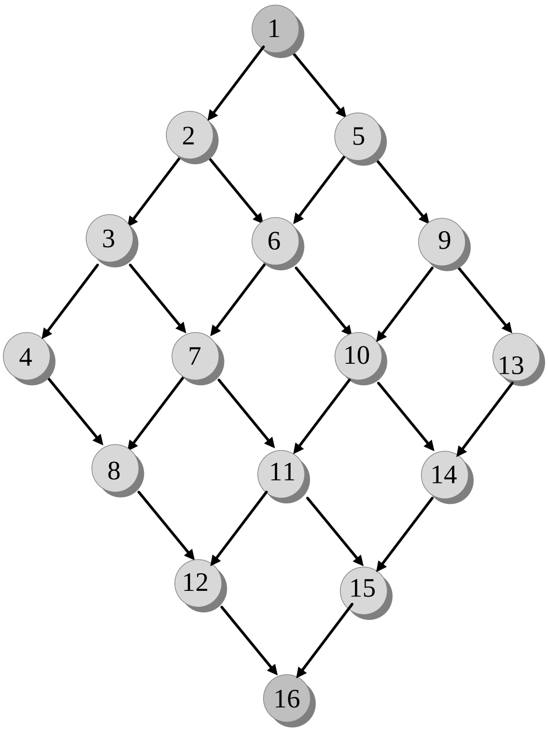


Figure 5.2: Graph representation of nodes of 2D-Mesh.

Assuming, $g_9 \rightarrow g'_1$, $g_{14} \rightarrow g'_2$, $g_{16} \rightarrow g'_3$, and $g_{21} \rightarrow g'_4$. Similarly, $y_9 \rightarrow y'_1$, $y_{14} \rightarrow y'_2$, $y_{16} \rightarrow y'_3$, and $y_{21} \rightarrow y'_4$ and $x_9 \rightarrow x'_1$, $x_{14} \rightarrow x'_2$, $x_{16} \rightarrow x'_3$, and $x_{21} \rightarrow x'_4$.

$$\begin{bmatrix} g_1' & 0 & 0 & 0 \\ 0 & g_2' & 0 & 0 \\ 0 & 0 & g_3' & 0 \\ 0 & 0 & 0 & g_4' \end{bmatrix} \begin{matrix} y_1' \\ y_2' \\ y_3' \\ y_4' \end{matrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} x_1' \\ x_2' \\ x_3' \\ x_4' \end{matrix}$$

Assuming that the data at $g_1' = [1 \quad 0 \quad 0 \quad 0]$, $g_2' = [0 \quad 1 \quad 0 \quad 0]$, $g_3' = [0 \quad 0 \quad 1 \quad 0]$, and $g_4' = [0 \quad 0 \quad 0 \quad 1]$. So, the data received by the destination node g_{15} is

$$\begin{array}{rcccc} & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \\ + & 0 & 0 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

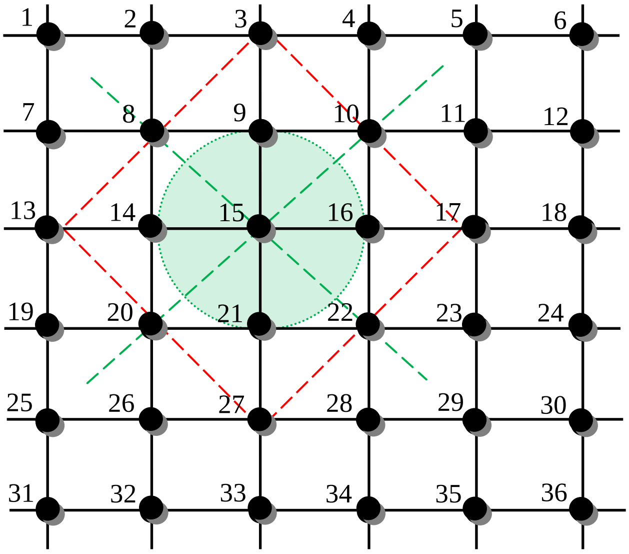


Figure 5.3: Communication in 2D-Mesh network.

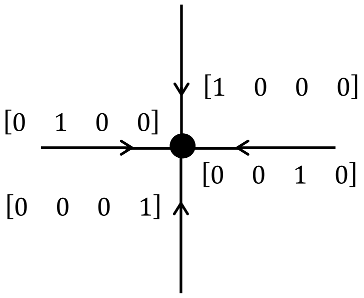


Figure 5.4: Radius nodes receiving data from their respective contour nodes.

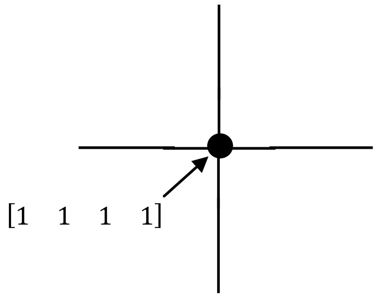


Figure 5.5: XOR operating at radius node.

The final XOR data is of same size which the destination node contains previously, i.e. using the network coding broadcasting information between different nodes reduces the issues of data size (figure 5.4 and 5.5). We have considered the traditional approach to communicate data set between same nodes (i.e., nodes 9, 14, 16 and 21 results in an array having four data values at different locations). The size of this array increases with the increases in data communication. For each communication step data storage increases with the increasing data size (see figure 5.6).

0	1	2	3
[1 0 0 0]	[0 1 0 0]	[0 0 1 0]	[0 0 0 1]

Figure 5.6: Increase in size of array at each data receiving nodes.

According to the traditional approach of communication in parallel networks, the communication and computation time depends on the size of data (see figure 5.7). Widmer and et. at. [135] have proposed that for wireless rectangular network, the total number of transmission required to broadcast one information unit to all nodes using network coding $(T_{nc}) \geq \frac{n^2}{4}$ and without network coding $(T_w) \geq \frac{n^2}{3}$. So,

$$\frac{T_{nc}}{T_w} = \frac{n^2/4}{n^2/3} = \frac{3}{4} = 0.75$$

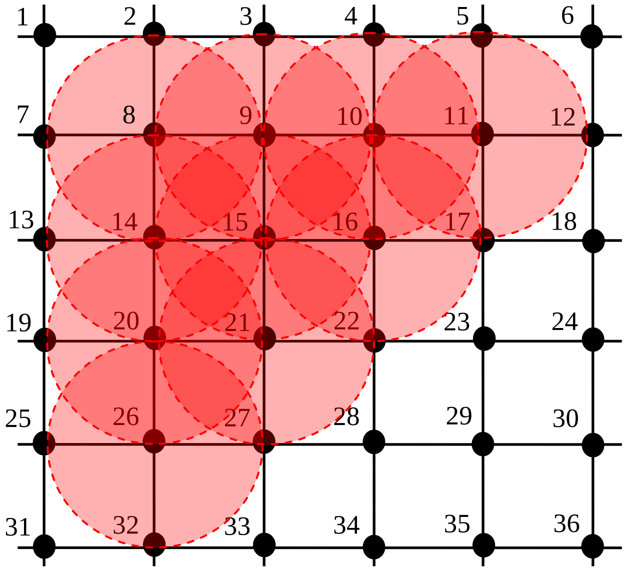


Figure 5.7: Size of data affects communication and computation time in parallel networks.

Figure 5.8 performs data communication using network coding in the form of contour approach. The data from node 2, 5, 6, 7 and 10 reaches to node 6. Similarly, the data from node 4, 7, 8 and 12 reaches to node 8 and the data from node 10, 13, 14 and 15 reaches node 14. While node 6, 8 and 14 act as radius nodes in respective contours. Network coding performs at these radius nodes where it collects the data from neighboring nodes. The data received by these nodes are as below;

$$node\ 6 = 2 + 5 + 6 + 7 + 10;$$

$$node\ 8 = 4 + 7 + 8 + 12;$$

$$node\ 14 = 10 + 13 + 14 + 15;$$

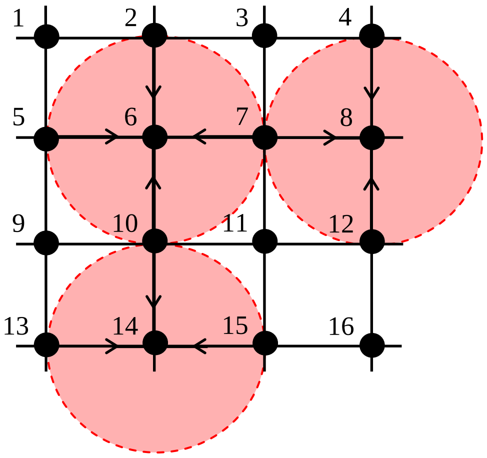


Figure 5.8: 4×4 2D-Mesh with transmission towards node 6, 8, 14 from neighboring nodes.

After receiving the data, the radius nodes re-transmit data to respective neighboring nodes. The data content at these neighboring nodes is XOR with the previous data (as in figure 5.8). The re-transmission is conducted as in figure 5.9.

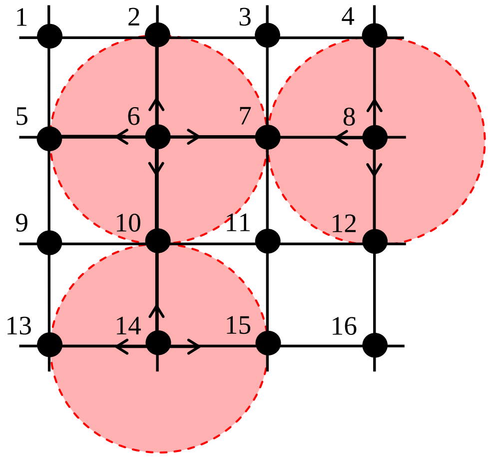


Figure 5.9: Re-transmission of data from radius nodes to other nodes of respective contours.

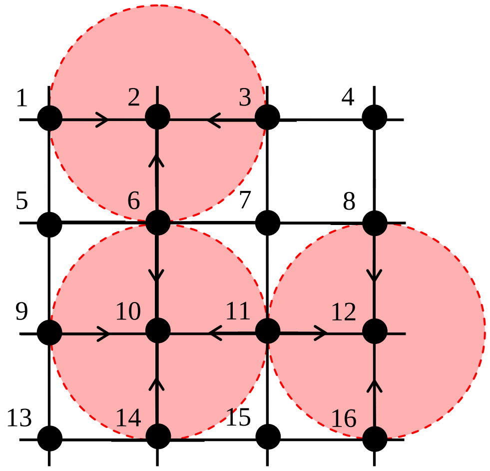


Figure 5.10: Formation of contour in 4×4 2D-Mesh with different radius nodes.

The data at the neighboring nodes 2, 4, 5, 7, 10, 12, 13, 15 can be calculated as:

$node\ 2 = 2 + 5 + 6 + 7 + 10;$

$node\ 4 = 4 + 7 + 8 + 12;$

$node\ 5 = 2 + 5 + 6 + 7 + 10;$

$node\ 7 = 2 + 4 + 5 + 6 + 7 + 8 + 10 + 12;$

$node\ 10 = 2 + 5 + 6 + 7 + 10 + 13 + 14 + 15;$

$$node\ 12 = 4 + 7 + 8 + 12;$$
$$node\ 13 = 10 + 13 + 14 + 15;$$
$$node\ 15 = 10 + 13 + 14 + 15;$$

After receiving data, XOR operation is performed on the data of each neighboring node. Each node updates its information for further communication. To completely transfer the information between every other node of 2D-Mesh network, the contour is formed with other nodes of the network (see figure 5.10). To enable complete transfer of information within this network the contour position is changed. This change is performed until complete data is received by each node. According to the above figure, data transfers from the radius nodes 2, 10 and 12 to the neighboring nodes of each contour. The data is collected at the radius nodes and XOR operation is performed. Node 2 receives data from 1, 3 and 6; node 10 receives data from 6, 9, 11 and 14, and node 12 receive data from 8, 11 and 16. The data received by these nodes are as below;

$$\begin{aligned} node\ 2 &= 1 + \ddot{2} + 3 + \ddot{6}; \\ &= 1 + \left(2 + 5 + 6 + \dot{7} + 10\right) + 3 + \left(2 + 5 + 6 + \dot{7} + 10\right); \\ &= 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 10 + 12; \end{aligned}$$

$$\begin{aligned} node\ 10 &= \ddot{6} + 9 + \ddot{10} + 11 + \ddot{14}; \\ &= \left(2 + 5 + 6 + \dot{7} + \dot{10}\right) + 9 + \left(13 + \dot{14} + 15\right) + 11 + (10 + 13 + 14 + 15); \\ &= 2 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15; \end{aligned}$$

$$\begin{aligned} node\ 12 &= \ddot{8} + 11 + \ddot{12} + 16; \\ &= \dot{4} + \dot{7} + 8 + 11 + \left(2 + 4 + 5 + 6 + 7 + 8 + \dot{10} + 12\right) + 16; \\ &= 2 + 4 + 5 + 6 + 7 + 8 + 10 + 11 + 12 + 13 + 14 + 15 + 16; \end{aligned}$$

Similarly, as in previous case, the XOR operation is performed on the data received by neighboring nodes. These nodes, updates the information and accumulate data for further transmission. Now, the radius nodes 2, 10 and 12 send this information to respective neighboring nodes (see figure 5.11).

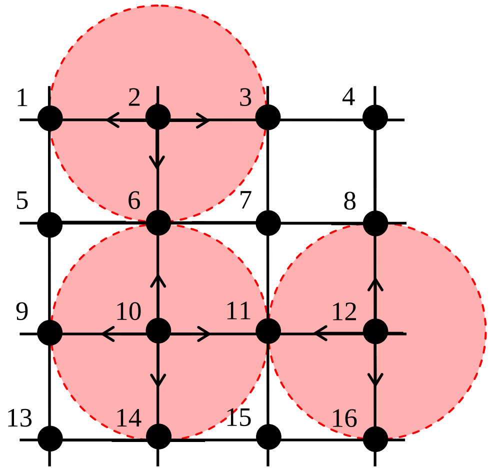


Figure 5.11: Re-transmission of data from radius nodes to other nodes of respective contours.

The data at the neighboring nodes 1, 3, 6, 8, 9, 11 and 16 are as below:

$node\ 1 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 10 + 12;$

$node\ 3 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 10 + 12;$

$node\ 6 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15;$

$node\ 8 = 2 + 4 + 5 + 6 + 7 + 8 + 10 + 11 + 12 + 13 + 14 + 15 + 16;$

$node\ 9 = 2 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15;$

$node\ 11 = 2 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16;$

$node\ 16 = 2 + 4 + 5 + 6 + 7 + 8 + 10 + 11 + 12 + 13 + 14 + 15 + 16;$

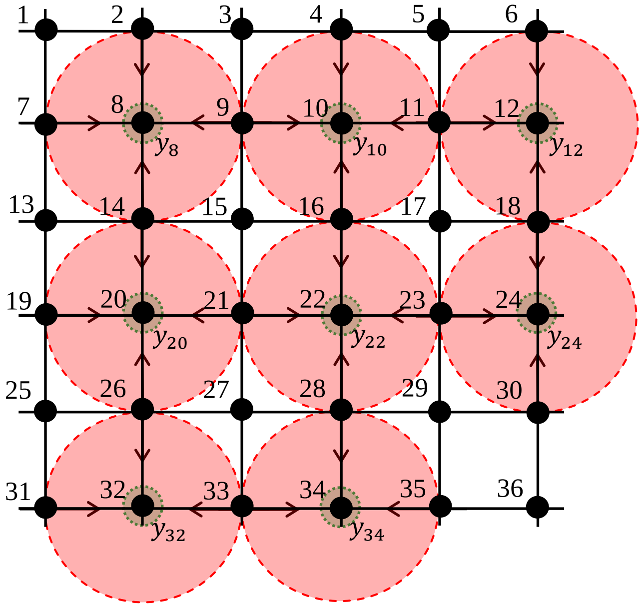


Figure 5.12: Contours in a 6×6 2D-Mesh network.

This shows the way to perform XOR operation on each node of 4×4 2D-Mesh network. Further we have developed the complete network coding by involving information field and encoding vector on 6×6 2D-Mesh network. Figure 5.12 shows network of 36 nodes in a 2D-Mesh with four complete and four semi-contours. The information field associated with encoding vector for radius nodes comprises XOR of all the incomings towards the radius nodes. Combining such nodes is given below. The matrix formed to evaluate the network coding is also given consecutively.

$$y_8 = g_2x_2 + g_7x_7 + g_9x_9 + g_{14}x_{14}.$$
$$y_{10} = g_4x_4 + g_9x_9 + g_{11}x_{11} + g_{16}x_{16}.$$
$$y_{12} = g_6x_6 + g_{11}x_{11} + g_{18}x_{18}.$$
$$y_{20} = g_{14}x_{14} + g_{19}x_{19} + g_{21}x_{21} + g_{26}x_{26}.$$
$$y_{22} = g_{16}x_{16} + g_{21}x_{21} + g_{23}x_{23} + g_{28}x_{28}.$$
$$y_{24} = g_{18}x_{18} + g_{23}x_{23} + g_{30}x_{30}.$$
$$y_{32} = g_{26}x_{26} + g_{31}x_{31} + g_{33}x_{33}.$$
$$y_{34} = g_{28}x_{28} + g_{33}x_{33} + g_{35}x_{35}.$$

Suppose, the combination of all information fields associated with encoding vectors is denoted by y_c.

$$\begin{aligned} y_c = {} & g_2x_2 + g_4x_4 + g_6x_6 + g_7x_7 + g_9x_9 + g_{11}x_{11} + g_{14}x_{14} + g_{16}x_{16} + g_{18}x_{18} \\ & + g_{19}x_{19} + g_{21}x_{21} + g_{23}x_{23} + g_{26}x_{26} + g_{28}x_{28} + g_{30}x_{30} + g_{31}x_{31} \\ & + g_{33}x_{33} + g_{35}x_{35}. \end{aligned}$$

Similarly, the data of radius nodes is sent back to respective neighboring nodes. The data received is of same size as in previous transmission. Thus, this approach not only reduces the storage requirements but also enables fast data processing in such networks. Further, encoded data includes security aspect in parallel communication. Even when transaction fails, the loss can be recovered by coding data at previous stage. So, network coding approach is feasible and practical for parallel communication. It provides high reduction in data storage problem of massive parallel communication.

$$\begin{bmatrix}
g_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & g_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & g_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & g_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & g_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & g_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & g_{14} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{18} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{19} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{26} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{28} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{30} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{31} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{33} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{35}
\end{bmatrix}
\begin{matrix} y_2 \\ y_4 \\ y_6 \\ y_7 \\ y_9 \\ y_{11} \\ y_{14} \\ y_{16} \\ y_{18} \\ y_{19} \\ y_{21} \\ y_{23} \\ y_{26} \\ y_{28} \\ y_{30} \\ y_{31} \\ y_{33} \\ y_{35} \end{matrix}
\rightarrow
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{matrix} x_2 \\ x_4 \\ x_6 \\ x_7 \\ x_9 \\ x_{11} \\ x_{14} \\ x_{16} \\ x_{18} \\ x_{19} \\ x_{21} \\ x_{23} \\ x_{26} \\ x_{28} \\ x_{30} \\ x_{31} \\ x_{33} \\ x_{35} \end{matrix}$$

5.3 *Numerical Analysis of Proposed Approach*

With the help of examples we perform network coding on 2D-Mesh and reevaluate our results. Further we considered nodes 6, 8 and 14 as radius nodes of 4×4 2D-Mesh network (figure 5.8). Node 6 communicates with node 2, 5, 7 and 10 similarly, node 8 communicates with node 4, 7 and 12, and node 14 communicates with nodes 10, 13 and 15. For 4×4 2D-Mesh network, assume that all nodes consist of autonomous data before the communication begins. Data communicated using contour approach is show in figure 5.8. Data at each node is defined as below:

$node\ 1 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0].$

$node\ 2 = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0].$

$node\ 3 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0].$

$\vdots$

$node\ 16 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1].$ (for n nodes the matrix will have data at n^{th} location)

Consider node 6, which receives data from node 2, 5, 7 and 10 while it also contains its own data. This complete data is received at node 6 and XOR operation is performed to capitulate on the following result.

$$\begin{array}{r}
0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
+\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
\end{array}$$

Similarly, node 8 receives data from 4, 7 and 12 nodes with its own data. The XOR operation is performed to give the below given result. In same manner, node 14 receives data from 10, 13, 15 and its own data. The XOR operation for node 14 is also defined below.

For node 8,

$$\begin{array}{r}
0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
+\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
\end{array}$$

For node 14,

$$\begin{array}{r}
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \\
+\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 \\
\hline
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0 \\
\hline
\end{array}$$

Now, this approach is accepted for second contour where the radius nodes are 2, 10 and 12. In figure 5.11, node 2 communicates with node 1, 3 and 6. Similarly, node 10 communicates with nodes 6, 9, 11, 14, and node 12 communicates with nodes 8, 11 and 16. The XOR operation is performed on these radius nodes, and the result is stored on the respective nodes for further communication. The operations given below show the situation of data on 2, 10 and 12 nodes.

For node 2,

$$\begin{array}{r}
1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
+\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
\hline
\end{array}$$

For node 10,

$$\begin{array}{rl} & 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ + & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \\ \hline & 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \\ \hline \end{array}$$

The approach of network coding reduces the storage requirements at each node to an enormous extent. The factor to deal with is coefficient vector for encoding and decoding at each node. However, this factor is still less complex than the issue of the storage requirements of parallel network. We discuss the results in next section. After entire communication the final data set at each node 4×4 2D-Mesh network is a 16×16 matrix with each element is one.

5.4 *Results*

Without network coding in parallel communication the requirement is to keep track of data for the safety of communication failure. For this each node need to store the information of data present at that stage with it and the information of the next communicating node and the data. We consider an example which performs data communication in a 4×4 mesh network without using the network coding (see figure 5.13).

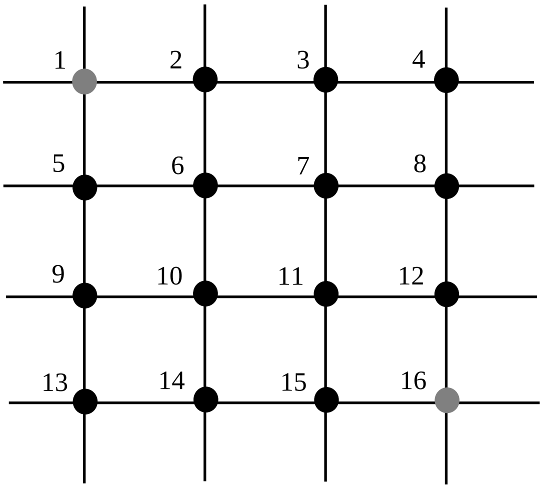


Figure 5.13: 4×4 Mesh network.

We have considered row and column broadcast in the mesh network and studied the reduced steps using the approach of network coding. At time t node 4 transfers its data to node 3. After receiving data from node 4, node 3 now consists of data of 3 and 4. Node 2 further receives this information and node 1 which receives data from node 2. So the data at each node of mesh network appears similar to the following:

$$\underset{(4)}{Node\ 4} \rightarrow \underset{(3,4)}{Node\ 3} \rightarrow \underset{(2,3,4)}{Node\ 2} \rightarrow \underset{(1,2,3,4)}{Node\ 1}.$$

$$\underset{(8)}{Node\ 8} \rightarrow \underset{(7,8)}{Node\ 7} \rightarrow \underset{(6,7,8)}{Node\ 6} \rightarrow \underset{(5,6,7,8)}{Node\ 5}.$$

$$\vdots$$

$$\underset{(16)}{Node\ 16} \rightarrow \underset{(15,16)}{Node\ 15} \rightarrow \underset{(14,15,16)}{Node\ 14} \rightarrow \underset{(13,14,15,16)}{Node\ 13}.$$

At the end of this step, node 1 received data from all nodes from its row. Similarly, node 5, 9 and 13 receives all the information from respective rows. To transfer data in a row involves three steps of communication. Now for column broadcast at time $t+\tau$ the data transfer is performed in each column at a time (in parallel). The column broadcast appears as below:

$$\underset{(13)}{Node\ 13} \rightarrow \underset{(9,13)}{Node\ 9} \rightarrow \underset{(5,9,13)}{Node\ 5} \rightarrow \underset{(1,5,9,13)}{Node\ 1}.$$

$$\underset{(14)}{Node\ 14} \rightarrow \underset{(10,14)}{Node\ 10} \rightarrow \underset{(6,10,14)}{Node\ 6} \rightarrow \underset{(2,6,10,14)}{Node\ 2}.$$

$$\vdots$$

$$\underset{(16)}{Node\ 16} \rightarrow \underset{(12,16)}{Node\ 12} \rightarrow \underset{(8,12,16)}{Node\ 8} \rightarrow \underset{(4,8,12,16)}{Node\ 4}.$$

After the column broadcast, node 1 receives data from node 13, 9 and 5. In the same way, node 2, 3 and 4 receives data from respective columns. Subsequently, we formulate both these steps of row and column broadcast in a tabular form (table 5.1). This table shows the communication at each node for both these steps. Based on this the complexity of row and column broadcast over 4×4 mesh network is formulated without using network coding. The above table shows that the communication complexity is increasing with the increase in steps of communication. Now, at any step if communication fails, then it is very difficult to keep track of communication link. Furthermore, with this approach, it is very complex to optimize the storage requirement at each node. The storage at each node increases with the increase in communication steps. Now, let us re-consider the approach of network coding on the mesh network as in section 5.3.

Table 5.1: Data with each node at steps of communication.

STEPS	Node/Data	Node/Data	Node/Data	Node/Data
Step 1	Node 1 1, 2, 3, 4	Node 2 2, 3, 4	Node 3 3, 4	Node 4 4
	Node 5 5, 6, 7, 8	Node 6 6, 7, 8	Node 7 7, 8	Node 8 8
	Node 9 9, 10, 11, 12	Node 10 10, 11, 12	Node 11 11, 12	Node 12 12
	Node 13 13, 14, 15, 16	Node 14 14, 15, 16	Node 15 15, 16	Node 16 16
Step 2	Node 1 (1, 2, 3, 4), 5, 9, 13	Node 2 (2, 3, 4), 6, 10, 14	Node 3 (3, 4), 7, 11, 15	Node 4 (4), 8, 12, 16
	Node 5 (5, 6, 7, 8), 9, 13	Node 6 (6, 7, 8), 10, 14	Node 7 (7, 8), 11, 15	Node 8 (8), 12, 16
	Node 9 (9, 10, 11, 12), 13	Node 10 (10, 11, 12), 14	Node 11 (11, 12), 15	Node 12 (12), 16
	Node 13 (13, 14, 15, 16)	Node 14 (14, 15, 16)	Node 15 (15, 16)	Node 16 (16)

In figure 5.14, the amount of space required is compared with storage required at each node with and without network coding. The data used in this figure is declared in table 5.2. The data received by using network coding and without using is same, but the data sent from the respective nodes of contour 1 and 2 decreases when this approach is exploited.

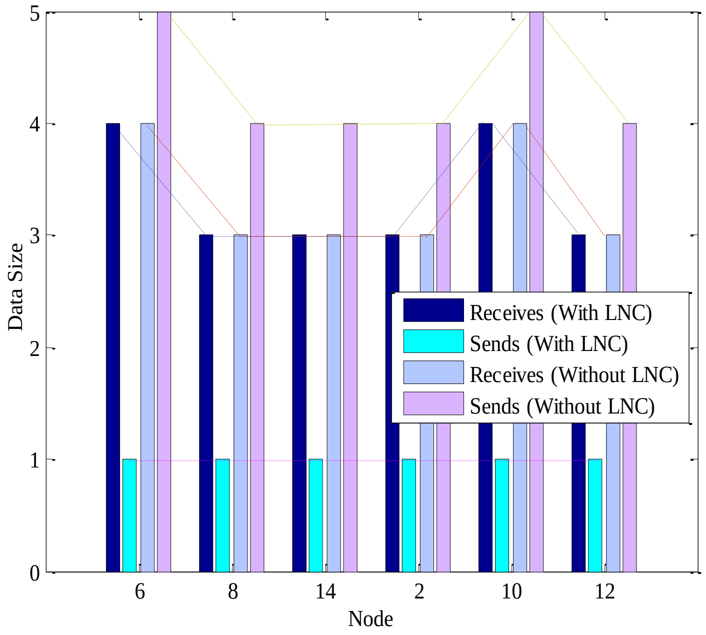


Figure 5.14: Nodes in Mesh network receiving and sending data with and without network coding. The figure denoted nodes of contour 1 and 2. Nodes 6, 8, 14 are of contour 1 while nodes 2, 10, 12 are of contour 2.

While in case of communication without contour approach or without network coding, the main drawback is the rate of information communication which turns out slower. Furthermore, in the traditional store and forward method of parallel communication the storage size increases with the increase in the steps of communication. We have examined how the network coding approach is advantageous over this traditional approach of communication in parallel networks.

Table 5.2: Size of data at each node.

Contour	**Nodes**	**Amount of data received and send with LNC**		**Amount of data received and send without LNC**	
		Receive	Send	Receive	Send
Contour 1	Node 6	4	1	4	5
	Node 8	3	1	3	4
	Node 14	3	1	3	4
Contour 2	Node 2	3	1	3	4
	Node 10	4	1	4	5
	Node 12	3	1	3	4

In 4×4 mesh network, which forms one full contour and two half contours. One full contour involves five nodes with one radius node, and these nodes receive/send data from four other nodes of respective contours. While one half contour involves four nodes, with one radius node and it receives/sends data from/to three other nodes. Similarly, the 6×6 mesh network involves four full and four half contours. Now, for 4×4 mesh network, nodes communicating at first communication step are,

$$= 3\, nodes\ (in\ half\ contour) + 4\, nodes\ (in\ full\ contour) \\ + 3\, nodes (in\ half\ contours).$$

$$= 10\, nodes.$$

For second communication step, the number of communicating nodes $= 3\, nodes$. As these three nodes are the radius nodes of three different contours, at this step each contour node contains information of the other nodes of respective contour. Similarly for 6×6 mesh network, the number of communicating nodes at first communication step,

$$4\ nodes(in\ full\ contour) + 4\, nodes(in\ full\ contour) + 3\, nodes\ (in\ half\ contour) + \\ 4\, nodes(in\ full\ contour) + 4\, nodes(in\ full\ contour) + 3\, nodes(in\ half\ contours) + \\ 3\, nodes(in\ half\ contours) + 3\, nodes(in\ half\ contours). \\ = 28\, nodes.$$

Similarly, for second communication step the number of communicating nodes in 6×6 mesh network$= 8\,nodes$. Further we considered the case of communication without contour approach or without network coding. For 4×4 mesh network, number of nodes communicating at first communication step of row broadcast (for example) are $= 1+1+1+1+1+1 = 6\,nodes$(one node from each row) and for second communication step the nodes communicating are $= 1+1+1+1+1+1 = 6\,nodes$ (second node from each row). We compared these results and observe that without using the network coding approach it is not feasible to perform communication in the parallel networks (figure 5.15). After establishing results for the rate of communication in 2D-Mesh network by using network coding, we examined the storage requirements at each node in this network. For 4×4 mesh network, the first communication step involves 10 communicating nodes and the radius nodes store data from these communicating nodes. Data received by these three radius nodes are data from {1, 3, 6} node is stored at node 2; data from nodes {6, 9, 11, 14} is stored at node 10 and data from nodes {8, 11, 16} is stored at node 12.

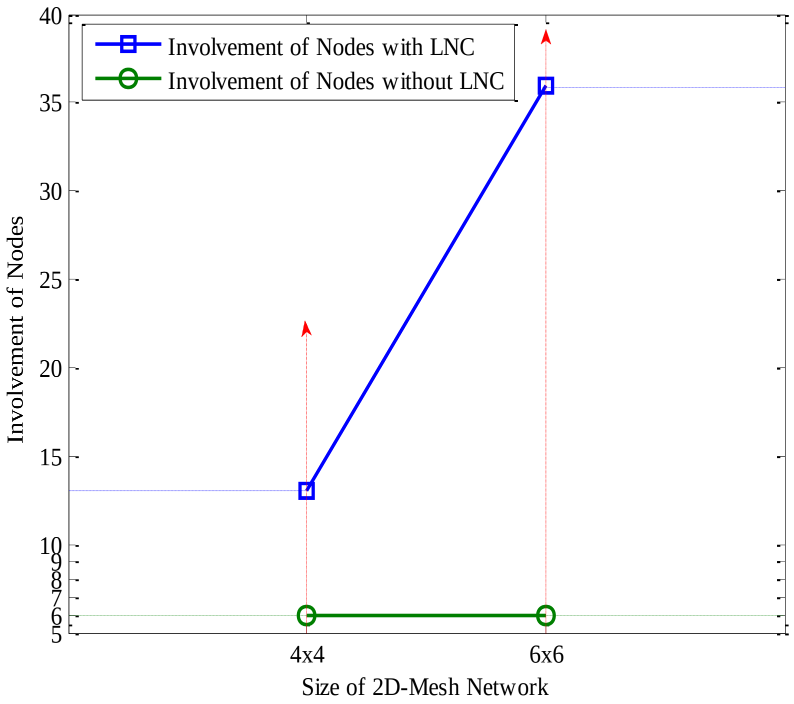


Figure 5.15: Rate of information transfer by using network coding and by using traditional approach for 4×4 and 6×6 mesh network.

When these radius nodes 2, 10 and 12 receive data, they perform XOR on this data and the size of final data at these radius nodes reduces to 1 bit (if all nodes consist of 1 bit data). Thus, the size of data remains the same as it was at its previous position. At second communication step, the radius nodes 2, 10 and 12 send the XOR information back to each node in respective contours. The data received by these nodes is again 1 bit. Now, irrespective of data size each node receives complete information of respective contour.

While in case of traditional approach of data communication, at first communication step the data is transferred in each row. The node 4 send its data to node 3, similarly node 8 send its information to node 7 and so on. Now the data size at node 3, 8, 12 and 16 becomes 2 bits. At second communication step, the data from node 3 is sent to node 2, similarly data from node 7 is sent to node 6 and so on. Now the data size at receiving nodes 2, 6, 10 and 14 becomes 3 bits. This shows that as the communication steps increase the data size using the conventional approach of communication also increases. Figure 5.16 shows how the data size at each node decrease using the network coding approach with respect to traditional approach.

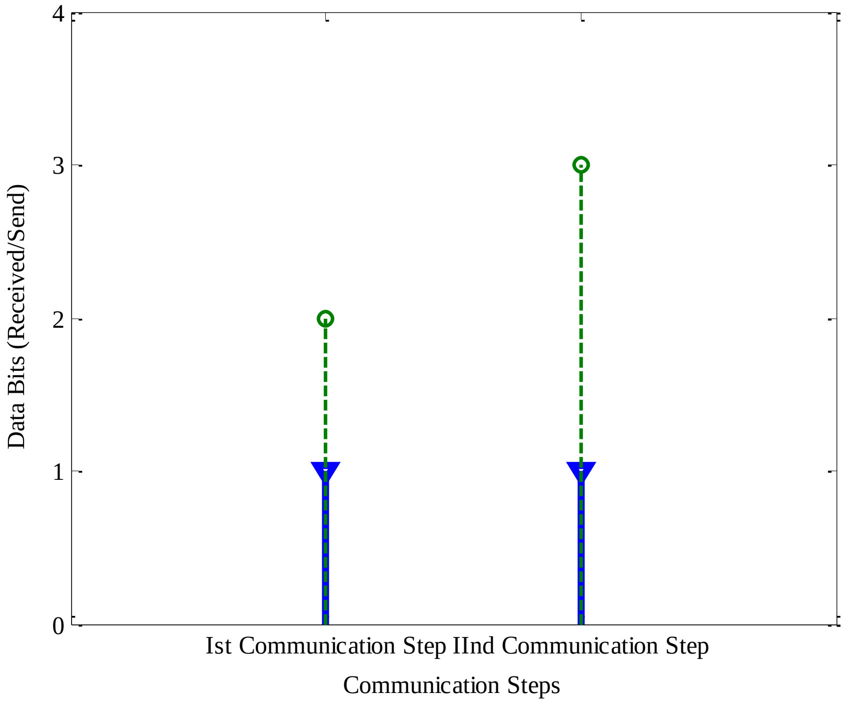


Figure 5.16: Data storage requirements at each communication step for 4×4 and 6×6 mesh network with and without network coding.

This shows that communication rate using the network coding approach increases with the increase in the size of network. It also shown that information transferred to each node moreover increases without increasing the size of information. The only factor with network coding approach is that, for each communication, the XOR is performed and the encoding and decoding factors are generated. However, it also proved consecutively in [136] that time required to compute the encoding and decoding factor is less than the time and spaces required to store and forward the data during each communication. Parallel networks are certainly a satisfactory way to perform high computation in lesser time. It clearly works, but it is difficult to reduce the data size at each communicating node for such networks. The time and space requirements for such networks increase exponentially with the increase in communication. To actually formulate parallel networks practical in use, it is important to overcome these issues.

5.5 *Chapter Outline*

This chapter presents the problem of high data size during communication and proposes network coding approach as a solution. To examine this problem 2D-Mesh network is considered. Further, the issues of information rate and the number of nodes involved during parallel communication are studied using this approach. Further in this chapter it is proved that this approach (network coding approach in a contour manner in parallel networks) reduces the data size at each communication step by performing the XOR operation on the data set. In addition, it is proved that using network coding with a contour communication manner the involvement of nodes in communication increases. Furthermore, due to the node involvement per step of communication, the rate of information transfer also increases.