End-to-End Reliability in Wireless Sensor Networks through Network Coding

MS Research Dissertation
By

Saleem Zahid
(406-FBAS/MSCS/S08)

Supervisor

Prof. Dr Muhammad Sher
Chairman
Dept. of Computer Science,
IIU, Islamabad.

Co-Supervisor

Dr. Muhammad Zubair
Chairman
Department of Electronic Engineering,
IIU, Islamabad.

Department of Computer Science
Faculty of Basic and Applied Sciences,
International Islamic University Islamabad
2012

- 1 Sensor networks
- 2 Wireless LANs.

A Dissertation submitted to the

Department of Computer Science

International Islamic University Islamabad

As a partial fulfillment of requirements for the award of

the Degree of

MS in Computer Science

Acknowledgment

All praises to almighty Allah Who has all the names, and need no name the most generous, considerate and compassionate Who has blessed mankind with this verdict to think, explore, to learn and discover the hidden secrets of this universe and helped me to broaden the veils of my thought and enabling me to get through the difficulties indulged during this project. Also, admiration to our beloved Prophet Muhammad (PBUH) Who is always a great source of inspiration of divine devotion and dedication to me.

I would cordially pay my special appreciations and whole heartedly considerations to my reverend supervisors Prof. Dr Muhammad Sher and Dr Muhammad Zubair for their endless support, guidance and coordination while conducting this project. I owe them a great respect and honor and I am privileged to work under their supervision. It is their efforts, courage, moral support and endeavoring attitude that helped me to get through my problem and difficulty during each step of this project.

Finally, my beloved parents and family who deserve the credit more than I could ever express for always being completely supportive to me. They have been a constant source of advice, love and devotion to me. From moral to financial they have been blessing me with all the support that I needed up to till now in my life. I express my countless appreciation to all the people who have helped me during achieving this MS degree and hope to have this honor that they would walk along me throughout my life.

Saleem Zahid

Final Approval

This is to certify that we have read the thesis submitted by **Saleem Zahid, Reg# 406-FBAS/MSCS/S08**. It is our judgment that this project is of standard to warrant its acceptance by the International Islamic University, Islamabad, for the Degree of **MS in Computer Science**.

Project Evaluation Committee

External Examiner:
Dr. Mujahid Alam
Chief Scientific (PAEC)
Islamabad

Internal Examiner:
Mr. Muhammad Saqlain
Assistant professor,
Department of Computer Science and Software Engineering
International Islamic University
Islamabad

Supervisor:
Dr. Muhammad Sher
Professor
Chairman
Department of Computer Science and Software Engineering,
International Islamic University
Islamabad

Date Approved: ______________________________

Declaration

I hereby declare that this work, neither as a whole nor as a part has been copied out from any source. It is further declared that I have conducted this research and have accomplished this dissertation entirely on the basis of my personal efforts and under the sincere guidance of my supervisor Dr. Muhammad Sher and Co-supervisor Dr. Muhammad Zubair. If any part of this project is proved to be copied out from any source or found to be reproduction from some other project, I shall stand by the consequences. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Saleem Zahid
(406-FBAS/MSCS/S08)

Abstract

WSNs have limited resources such as energy, computational capabilities and storage spaces. Generally the environment in which sensor nodes are deployed is harsh and fluctuant. This causes rapid changes in wireless links characteristics such as signal strength, interference etc. Beside these, node failure is a common issue in WSN. In such error prone conditions, ensuring reliable communication with lower cost becomes a challenging and important issue. Several approaches such as ARQ, FEC, and Multi-Path Forwarding etc have been found in the literature to handle this issue. However, these traditional approaches are not suitable for WSN due to resources constraints. By considering these specific requirements of WSN, we are using random linear network coding to provide reliable transfer of data with lower cost. The use of random linear network coding with a reasonable amount of redundancy ensures the successful recovery of lost data with high probability of decoding at the destination. The methodology we used and the simulation results confirm our work as a reasonable approach towards reliable data transmission in WSN as compared to the existing approaches.

Table of Contents

List of Figures

Table of Abbreviation

S.No	Abbreviation	Acronyms
1	ADC	Analog to Digital Converter
2	ARQ	Automatic Repeat Request
3	DLNC	Deterministic Linear Network Coding
4	FEC	Forwarded Error correction
5	F_2^s	Finite Field of size 2^s
6	G	Generation Size
8	NC	Network Coding
9	NECO	Network Coding Simulator
7	RLNC	Random Linear network Coding
10	WSNs	Wireless Sensor Networks

Chapter 1

Introduction to Wireless Sensor Networks

Wireless sensor networks are widely used nowadays, because of its broader applications domain. But several constraints restrict its use and need additional measures to be consider. This chapter provides a detail description on different aspects of WSN.

Section 1.1 contains classifications of wireless technologies. Section 1.2 and its subsections provide an overview at different aspects of wireless sensor networks such as communication architecture, hardware constraints of sensor nodes and topologies. Section 1.3 lists some important applications of sensor networks. The environment in which sensor nodes are deployed is highlighted in section 1.4. Section 1.5 is about the dissertation summery.

1.1 Classification of Wireless Networks

Wireless network technologies can be classified into two main categories i.e. single hop and multi hop. In single hop setup, nodes directly interact with the relay agents (sinks) to communicate. While in multi-hop setup, nodes can interact with each other and with relay agents (sinks). Multi-hop interaction has its own advantages over single hop. The classification of different networks on the basis of their interaction is given in figure 1.1.

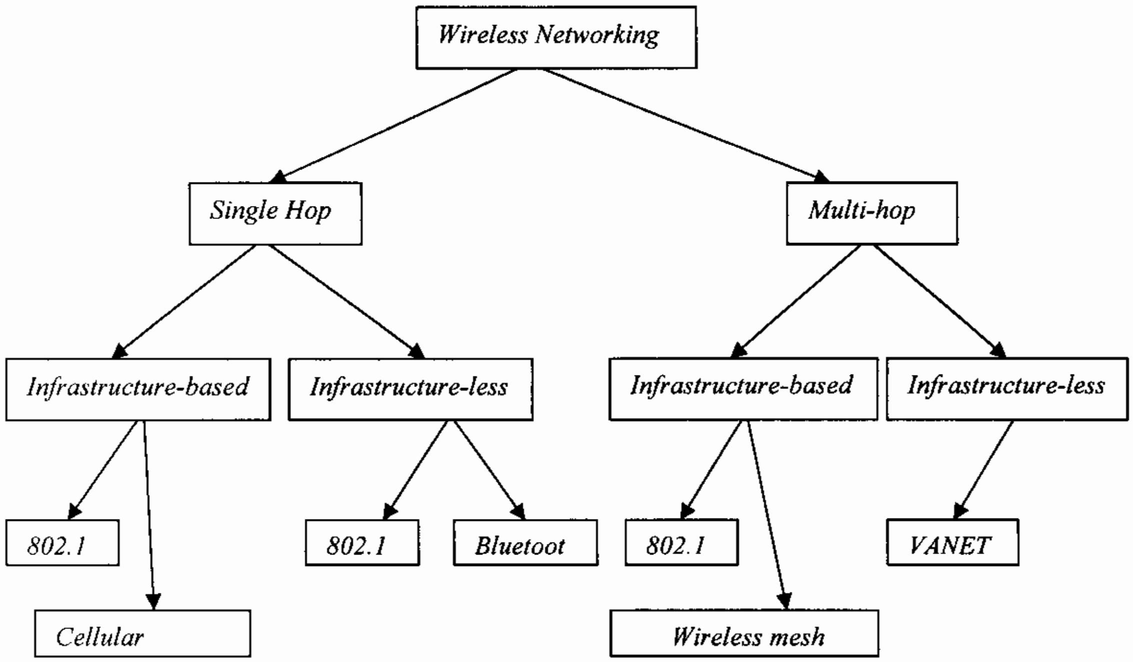

Figure 1.1 Categories of Wireless Networks

1.2 Wireless Sensor Networks

Recent advances in wireless communication and digital electronics have enabled the development of low-cost, low-power sensor nodes. These sensor nodes can communicate with each other through multi-hop broadcast paradigm. Sensor nodes collect data from the environment, and then collected data are routed back to the end user. In sensor networks nodes have no global ID. Sensor networks use attribute-based naming or location-based addressing instead of IP addresses. For example, the location of the nodes where temperature is higher than 70 F^0, is an attribute-based query. Similarly, temperature read by nodes in region A, is an example of location-based naming [1,2].

1.2.1 Sensor Nodes

Generally a sensor node has four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit as shown in figure 1.2 .These may have application dependent additional components such as location finding system, a power generator and a mobilizer. Application dependent components are shown in dotted lines.

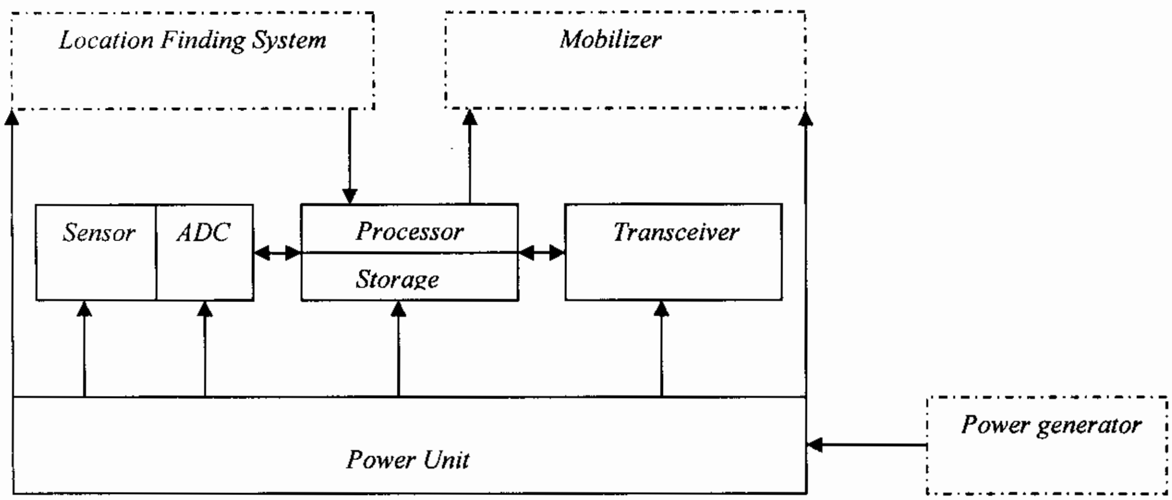

Figure 1.2 Components of a Sensor Node

Sensing units are usually composed of two sub units: sensors and analog to digital converters (ADCs). The analog signals produced by the sensors are converted to digital signals by ADC and then passed to the processing unit. Processing unit is generally associated with a small storage unit. A transceiver unit connects sensor node to network. One of the most important components is the power unit. The power unit may be supported by battery or solar cells. Another important component of a sensor node is the location finding system, because most of the routing techniques require knowledge about the location of sensor nodes for high accuracy. Mobilizer is needed to move sensor nodes when it is required. The required size of all these subunits may be one cubic centimeter [2], However it also depends on the application used for.

1.2.2 Wireless Sensor Network Topology

Generally, hundreds to several thousands of sensor nodes are deployed in the sensor field. It makes topology maintenance a challenging task. The nodes densities may be as high as 20 nodes/m^3 [3].Usually, deployment of nodes are carried out in three phases.

Pre-deployment and deployment phase:

This is the initial phase of nodes deployment. In this phase nodes are deployed in different ways such as dropping from a plane, delivering in a rocket or missile, placing one-by-one either by human or a robot etc. The purposes of initial deployment are to reduce the installation cost, to increase the flexibility of arrangement and to promote self-organization and fault tolerance.

Post deployment phase

After initial deployment, topology changes occur frequently due to nodes position, nodes reachability, available energy and malfunctioning of nodes. So, in post-deployment phase nodes are deployed statically to handle the mentioned issues. Static deployment deploys nodes one by one in the sensor field.

Re-deployment of additional nodes phase

In this phase, Additional sensor nodes can be redeployed at any time to replace the malfunctioning nodes. However, additions of new nodes require re-organization of the network. Re-deployment of additional nodes phase can be used at any time in the life time of the network when needed.

1.2.3 Wireless Sensor Networks Communication Architecture

Sensor nodes are usually scattered in the sensor field as shown in figure 1.3. Each sensor node in the field has the capability to collect data from the environment. The collected data are routed back to the end user by a multi-hop infrastructure less architecture through the sink (relay agent).

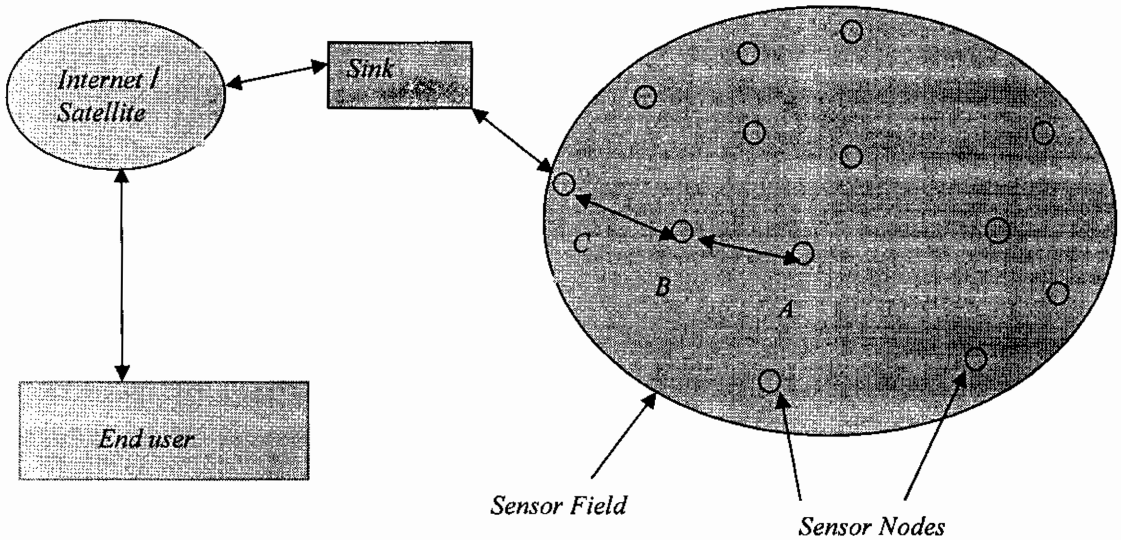

Figure 1.3 Wireless Sensor Network

The protocol stack used by sensor network is shown in figure 1.4. The protocol stack consists of the application layer, transport layer, network layer, data link layer and physical layer. Description of each layer is given below:

Application layer

Application layer protocols are used to administer sensor networks and performing different tasks at sensor nodes. Application software can be built and used at the application layer. This is an unexplored area of sensor networks [1], however there are two well known application layer protocols, sensor management protocol (SMP) and sensor query and data advertisement protocol (SQDDP).Application layer protocols uses attribute based naming or location base naming instead of IP addresses.

Sensor Management Protocol (SMP), System administrator interacts with the sensor network using SMP. SMP is a management protocol, where different operations can be performed by the administrator such as switching on/off sensor nodes, moving sensor nodes, authentication and querying sensor nodes etc. SMP uses attribute-based naming and location-based addressing.

Sensor Query and data dissemination protocol

SQDDP is an important application layer protocol. SQDDP provides an interface to perform different task by the administrator. The major operations to be performed with the SQDDP interface are issue queries, response to queries, collect replies etc. However, functionality of SQDDP and SMP is same but mainly depends on the application requirements.

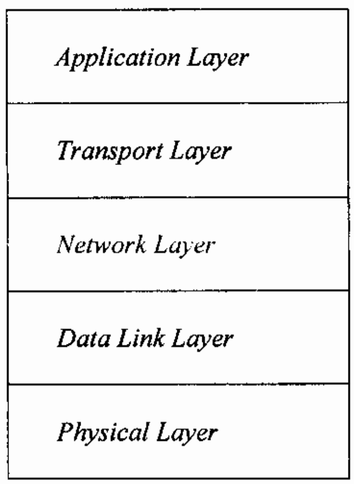

Figure 1.4 Protocol Stack of WSN

Transport Layer

Transport layer is needed especially when the system is planned to be accessed through internet or through other external network. Transport layer protocols use attribute-based naming. Existing transport layer protocols cannot be used for sensor networks because of resource constraints discussed in section 1.2.1. One well-known transport layer protocol is TCP splitting [4]. In this approach TCP connection ends at sink nodes and another transport layer protocol is used to handle communication between sink node and sensor nodes. Communication between sink and sensor nodes may be purely UDP type of protocol, because of limited storage at sensor nodes [1].

Network Layer

From figure 1.3, sensor nodes are scattered in a field either inside or very close to the phenomenon. Sensor networks require special multi-hop wireless protocols. There are some measures to be considered while designing network layer protocols for wireless sensor networks. Such as, power efficiency, attribute-based addressing and location awareness by keeping in the view that most of sensor networks are data centric. There are several approaches to select an energy efficient route. One approach is, maximum available power route (PA). In this approach the routing path having maximum available energy is preferred. The total PA is calculated by adding the PAs of each node along the path. Another approach is to use minimum energy route (ME). In this approach that route is selected between sensor nodes and sinks which consumes minimum energy. Minimum hop count (MH) is another well known technique in which the route with minimum hop count is preferred. So any network layer protocol to be designed must have to be considered these measures. Some popular network layer protocols are, Small Minimum Energy Communication Network (SMECN), Sensor Protocol for Information via Negotiation (SPIN), Sequential Assignment Routing (SAR) and Low Energy Adaptive Clustering Hierarchy (LEACH) etc.

Data Link Layer

The major functionalities of data link layer are multiplexing of data streams, data frame detection, medium access and error control. Data link layer ensures reliable point-to-point and point-to-multipoint connections in wireless sensor networks. One important part of data link layer is medium access control (MAC), MAC protocol in a wireless multi-hop self-organizing sensor network have two major goals. First of all MAC has to create network infrastructure for large number of nodes scattered in the sensor field and to establish communication link for data transfer. This provides a basic infrastructure for wireless sensor network and self-organization ability. Second objective of MAC is to fairly share the communication resources among the sensor nodes. Existing MAC protocols cannot be used for wireless sensor networks due to resource constraints. MAC protocols for wireless sensor networks must have built-in power conservation, mobility management and failure recovery strategies [1].However there are some proposed schemes like fixed allocation and random access version and demand-based MAC in [5],[6].

Physical Layer

Physical layer of sensor networks handles frequency selection, carrier frequency generation, signal detection, modulation and data encryption. All these aspects are based on the underlying hardware. Different modulation and carrier selection schemes are discussed in [3]. Physical layer is also restricted by the resource constraints of sensor networks as discussed in section 1.2.1.

1.3 Wireless Sensor Networks Applications

Sensor networks may consist of different types of sensors which can be used to monitor a wide variety of conditions such as temperature, humidity, noise levels, the presence or absence of certain kind of objects, vehicular movement, lightning condition, pressure, soil makeup, speed, direction and size of objects etc. The property of sensing and wireless connection among the sensor nodes creates a large room for several applications. These applications can be classified into military, environment, health, home and other commercial areas [1],[2].

Military Applications

The rapid deployment, self-organization and fault tolerance properties of sensor networks make them a very useful sensing technique for military purposes. As lower-cost sensor nodes are densely deployed in the battle field so destruction of these nodes by the enemy forces may not affect the overall military operation. The applications of sensor networks in military can be several in ways like monitoring friendly forces and equipments, targeting, battle damage assessment and nuclear, biological, chemical attacks detection.

Environmental Applications:

Environmental applications of sensor networks can be flood detection, forest fire detection, birds or animal movement, environmental conditions that can affect crop and live-stock, geophysical research and pollution study etc. For example thousand to millions of sensor nodes can be densely deployed in a forest. These nodes can monitor the forest and end user can assess the forest conditions.

Health Applications:

Wireless sensor networks have variety of applications in health. The possible applications are diagnosis of different diseases, providing interfaces for disabled, monitoring drug administration, tracking and tele-monitoring doctors and patients in hospitals, monitoring insect movement and internal processes in animals etc.

Home Applications

The most common sensor network application is home automation. For home automation smart sensor nodes are fitted at different locations, so telemonitoring is possible. Sometimes smart sensor nodes are buried in appliances like in micro-wave ovens, refrigerators etc. This allows the end user to manage these devices locally or remotely very easily.

Commercial Applications

There are several important applications of sensor networks in different commercial activities. Common commercial applications can be monitoring product quality, environmental control in office building, machines diagnosis, interactive toys, interactive museums, robot control and guidance in automatic manufacturing environment, vehicles tracking and detection, rotating machinery and managing inventory control etc.

1.4 Sensor Network Environment

Sensor nodes are densely deployed either inside the phenomenon or very close to the phenomenon to be observed. These work unattended in remote geographic locations. The environment of sensor nodes can be in the interior of large machinery, at the bottom of an ocean, in a battle field beyond the enemy lines, attached to fast moving vehicles or attached to animals etc [1]. So, generally sensors nodes have to work in sever conditions. For example, sensor nodes have to work under high pressure in the bottom of an ocean, in harsh environment such as a battle field and extremely noisy environment under intentional jamming.

1.5 Dissertation Outline

The chapter 1 of this dissertation is an introduction to wireless sensor networks, in a quick overview. The chapter 2 is the literature survey; it includes the previous work and corresponding limitations, of researchers who have contributed to this domain. Chapter 3 includes a detail overview of network coding, which is used as technique in the proposed solution to ensure reliability in wireless senor networks. Where chapter 4 contains requirement analysis to provide improved reliability in WSNs, this chapter also describes problem formulation, research objectives and the metrics used to analyze results. Chapter 5 includes the proposed solution and mathematical model to ensure reliability in wireless sensor networks. Chapter 6 contains implementation of the proposed scheme and corresponding result analysis. Current work is based on single source and destination; our future work will focus on multi source/destination pairs. At the end of this dissertation we have added References; here we provide the list of links which have been consulted while carrying out this research.

Chapter 2

Literature Survey

In this chapter reliability is discussed in the light of wireless sensor networks. First, WSN specific requirements, where reliability has to be maintained, are highlighted. Then in the light of these aspects, a review is given over the approaches used in the literature for reliability in wireless sensor network. At the end of the chapter a summary of the proposed solution is given.

2.1 Reliability in Wireless Sensor Networks

The discussion made in chapter 1 (section 1.4) has reflected that, the environments in which sensor nodes are deployed are fluctuant and error occurrence probability is high. So for successful data transmission, reliability is a challenging and important issue. Besides these the wireless medium suffers from signal strength variations and a common issue of node failure, results in data loss.

2.1.1 Reliability and Resource Constraints

Reliability is an important property of any communication system. The reliability requirements and standards vary from application to application. However, there are three important factors whose relation is important with reliability especially in the context of WSN i.e Memory, Computing Capability and Energy of the sensor node. The impact of sensor node memory and computing power on reliability is not crucial, because the advancement in micro-electronics has enabled smart sensor nodes with enough memory and computing power to implement reliability function [1,2]. However, energy is an important and crucial factor and needs to be considered while implementing reliability function in WSN especially. Node's energy consumption can be measured by the number of transmissions the node performs [12]. We have carefully designed a scheme to ensure reliability by keeping in view node's energy constraints. Our approach is simple and no additional transmissions are used. The simulation results confirm our work as an energy conservative. Our scheme consumes less energy as compare to others with corresponding reliability gains.

Several applications of wireless sensor networks are real time and involve mobility which further demands careful schemes for error recovery. There exist several approaches to handle this issue in the literature. But due to the resources constraints discussed above and in section 1.2.1, these approaches are not suitable for efficient error recovery in wireless sensor networks. The approaches used for reliability are given in the subsequent sections with their corresponding limitations particularly with respect to wireless sensor networks.

2.2 Automatic Repeat Request (ARQ)

Automatic repeat request also known as automatic repeat query (ARQ) is an error control mechanism that uses acknowledgments and timeouts to achieve reliable data transmission over an unreliable service. The acknowledgments (sent by the receiver) indicate that the data frame or packet has correctly received. Where timeouts (at the sender) are the specified periods of time allowed to elapse before an acknowledgment is to be received by the sender. If the sender does not receive an acknowledgment before the time out, it retransmits the frame/packet until the sender receives an acknowledgment or exceeds a pre-defined number of retransmissions [7]. There are different variations of ARQ protocols such as, Stop-and-wait ARQ, Go-Back-N ARQ and Selective Repeat ARQ.

2.2.1 Stop-and –Wait ARQ

Stop-and-Wait ARQ is the simplest kind of ARQ method and a special case of general sliding window protocol with window size equal to 1. In this method sender sends one frame/packet at a time and then waits for the acknowledgment. Typically the sender adds a redundancy check number (cyclic redundancy/ parity bits etc) to the end of each frame. The receiver uses the redundancy check number to check for possible damage, if there is no error found, the receiver sends an acknowledgement. In case of error detection no acknowledgment is given back. So, the sender waits for a predefine timer value, when timeout occurs then retransmission is performed.

In simplistic stop-and-wait ARQ several problems have to be addressed. One common problem is about the loss of ACK. In this case, the sender does not receive the ACK so it retransmits the frame again considering that the frame was lost. Now receiver has two copies of the same frame and does not know that the second one is a duplicate frame or the next frame of the sequence carrying identical data. Another problem is long latency. In this case the sender resends the frame because of no ACK received. Eventually the receiver has two copies of the same frame and ACK for each one. At this time the sender receives two ACKs which may cause problems if it assumes that the second ACK is for the next frame in the sequence. These problems have been addressed in Go-back-N ARQ and Selective Repeat ARQ.

2.2.2 Go-Back-N ARQ

Go-Back-N ARQ is a special case of general sliding window protocol with the transmit window size of N and receive window size of 1. In this approach the sender continuously sends a number of frames specified by a window size without receiving an acknowledgement from the receiver. However, the receiver process keeps track of the sequence of the next frame it expects to receive and sends that number with every ACK it sends. The receiver will ignore any frame that does not have the exact sequence number it expects. When the sender has sent all of the frames in its window, it will detect that all of the frames since the first lost frame are outstanding, and will go back to sequence number of the last ACK it received. Then the sender fills its window starting with that frame and will continue the process again. So Go-back-N ARQ is efficient than stop-and-wait ARQ, because in Go-back-N ARQ sender sends multiple frames rather than to wait for ACKs. However, In case of frame losses or ACK losses, this method results in sending frames multiple times. To avoid this issue Selective-repeat ARQ can be used [8].

2.2.3 Selective-Repeat ARQ

Selective-Repeat ARQ overcomes the deficiencies of Go-Back-N ARQ. In selective-repeat ARQ the sending process continues to send a number of frames specified by a window size even after a frame loss, where the receiving process will continue to accept and acknowledge frames sent after an initial error. In case of data loss, the receiving process keeps track of the sequence number of the earliest frame it has not received, and sends that number with every acknowledgment it sends, at the same time sending process continues to send subsequent frames until it has emptied its window. Once the sender has sent all the frames in its window, it resends the frame number given by the ACKs, and continues where is has left off. In selective-repeat ARQ, the size of sending and receiving window must be equal to avoid miscommunication in case of data loss [9].

There are several limitations of ARQ in the context of wireless sensor networks. First, ARQ is based on bidirectional traffic, which is not required in wireless sensor networks. Second, ARQ cannot deal with node failure, a common issue in wireless sensor networks. Third, ARQ cannot be used in real time applications, because of its bidirectional nature where ACKs are required for reliable delivery.

2.3 Forwarded Error Correction (FEC)

Forwarded error correction also known as channel coding, is an error control mechanism for data transmission. In FEC, the sender adds redundant data to its original message before forwarding. The redundant data is also known as error-correcting code (ECC). The carefully designed redundancy allows the receiver to detect and correct a limited number of errors occurring anywhere in the message without the need to ask the sender for retransmissions. FEC is generally used in the situations where retransmissions are relatively costly or impossible such as when broadcasting to multiple receivers. A simple example of FEC is to transmit each data bit three times (redundant bits), also known as (3, 1) repetition code. In this case the receiver might see eight versions of outputs as shown in figure 2.1. So, by adding the redundancy this scheme can recover from bit errors. However, by adding redundancy network coding can recover from burst errors because network coding allows mixing the information of different packets by encoding different packets and generating new packets which results into high probability of decoding and recovering data [14].

Triplet Received	*Interpreted as*
000	0(error free)
001	0
010	0
100	0
111	1(error free)
110	1
101	1
011	1

Figure 2.1 (3, 1) Repetition Code

There are two main categories of FEC codes. One is block codes, which work over fixed- size blocks (packets) of data. Another is convolutional codes which work on bit or symbol streams of arbitrary length [10].

However, there are several limitations exist of this approach. First, it is difficult to define a reasonable amount of redundancy in advance because errors are unexpected. Second, FEC cannot be used for high error rates because it can only deal with bit errors not with burst errors.

2.4 Multi-Path Forwarding

In this approach multiple paths are chosen between each source and destination pair for transmission. Where one path is selected as a primary path and others act as backup paths (used when primary path is broken).While in extreme circumstances (high errors rate) multiple paths can be used. Multi-path forwarding approach may be efficient in several scenarios where the data are too important and need to be delivered at any cost. But the cost at which reliability is maintained is very high. Besides these multi-path forwarding is a good candidate in case of node failure especially in wireless sensor networks because node failure is a most common issue due to energy exhaustance [11].

There are several limitations of multi-path forwarding. As, multi-path forwarding reserve multiple paths between the sender and receiver which results in extra network resources consumption i.e. wastage of resources. Beside this, multiple paths selection require additional traffic (exchanging control information among the nodes for paths selection) and computations at sensor nodes, which is not suitable for wireless sensor networks because of the resource constraints as discussed in section 1.2.1 in chapter one.

2.5 Reliability though Network Coding in Wireless Sensor Networks

Network coding is a newly explored area of Information theory and graph theory. Network coding is an in-network data processing technique. Traditionally data are transferred by the intermediate nodes blindly, but network coding provides data manipulation at the intermediate nodes. Detail description over network coding is given in chapter 3.

In the context of reliability in wireless sensor networks, network coding is a good candidate. An effort has been made in [12] to bring reliability in WSN through network coding. The mentioned scheme is based on random linear network coding which is very close to our working domain. There are two schemes, one is clustered based and another is for distributed settings. Both of them are based over normal distribution where a confidence interval is set to compensate the errors. The designed approach is adaptive in its nature i.e by changing the value of confidence interval according to the environment, the redundancy amount calculation also changes accordingly. So, the approach used is dynamic in its functionality. In clustered base settings, first of all cluster head identification and selection is required. Once cluster head is selected, cluster head exchanges control information (test packets) with other nodes. Cluster head estimates errors in the network by calculating statistical quantities such as mean, variance, standard deviation etc. on the basis of feed backs from other nodes. At last, cluster calculates the redundancy amount on the basis of estimated error rate in the network and broadcast it to other nodes. So, each node adjusts the redundancy amount accordingly. In distributed settings, no cluster head is required; however, same task has to be performed by each node. Each node exchange control information (test packets) with other nodes. Each node has to calculate separately statistical quantities such as mean, variance, standard deviation etc. to estimate errors in the network and redundancy amount is calculated accordingly.

However, there are limitations of the mentioned approach which limit its scope. First, the cluster-based settings require cluster head selection which results into additional transmissions (transmission overhead) affecting node's energy and network congestion. Second, redundancy amount calculation requires additional computations (computational overhead) such as mean, variance, standard deviation etc in cluster-based approach as well as in distributed settings etc. which results into extra energy consumptions at sensor nodes. Third, the mentioned transmission and computational overhead of both approaches results in delay which limit its scope and cannot be used for real time applications. Fourth, there are no considerations about the cluster head failure in cluster based settings.

2.6 Proposed Solution

The approaches discussed in this chapter for reliability have their own limitations in the context of wireless sensor networks. However, the work done in [12] is very close to our work. We are focusing this work and have critically analyzed. We are using random linear network coding for reliability purpose. Our scheme is dynamic and adjusts the redundancy amount at each node independently contrast to [12]. There are no pre-calculations and no additional transmission overhead. Our scheme is simple to launch. In our scheme each node adjusts the redundancy based on incoming packets without interacting with other nodes. Each node analyzes the number of packets it received and calculates the redundancy amount accordingly. We are also ensuring network congestion by providing upper bound limits over outgoing packets.

CHAPTER 3

Network Coding

Network coding is an in-network data processing technique where each intermediate node encodes incoming packets and produces newly encoded packets from the received packets. This chapter presents a detail description over network coding. Section 3.1 has discussed network coding and its interaction with different areas as well as canonical examples to explain network coding. Section 3.2 has a comprehensive discussion over network coding and its functionality. Section 3.3 has discussed types of network coding with real packets examples and their corresponding limitations and benefits. Sections 3.4, 3.5 and 3.6 have highlighted several issues in theoretical network coding and their corresponding solutions in practical network coding i.e real world usage. Network coding applications are presented in section 3.7.

3.1 Network Coding Overview

Network coding is a newly explored area which is the field of graph theory and information theory. Traditionally data are transferred by the intermediate nodes blindly, but network coding provides data manipulation at the intermediate nodes. Network coding is actually the mixing of data at the intermediate network nodes.

Network coding has its own significance in the context of wireless communication, because network coding exploits the broad cast nature of wireless networks efficiently and improve throughput [16]. It has its uniqueness in multi-casting as well. Network coding has several applications in wired networks as well as in wireless networks. Network coding and its possible applications in different areas are shown in figure 3.1. However, in wireless sensor networks network coding can be used in several dimensions like routing, security, reliability etc.

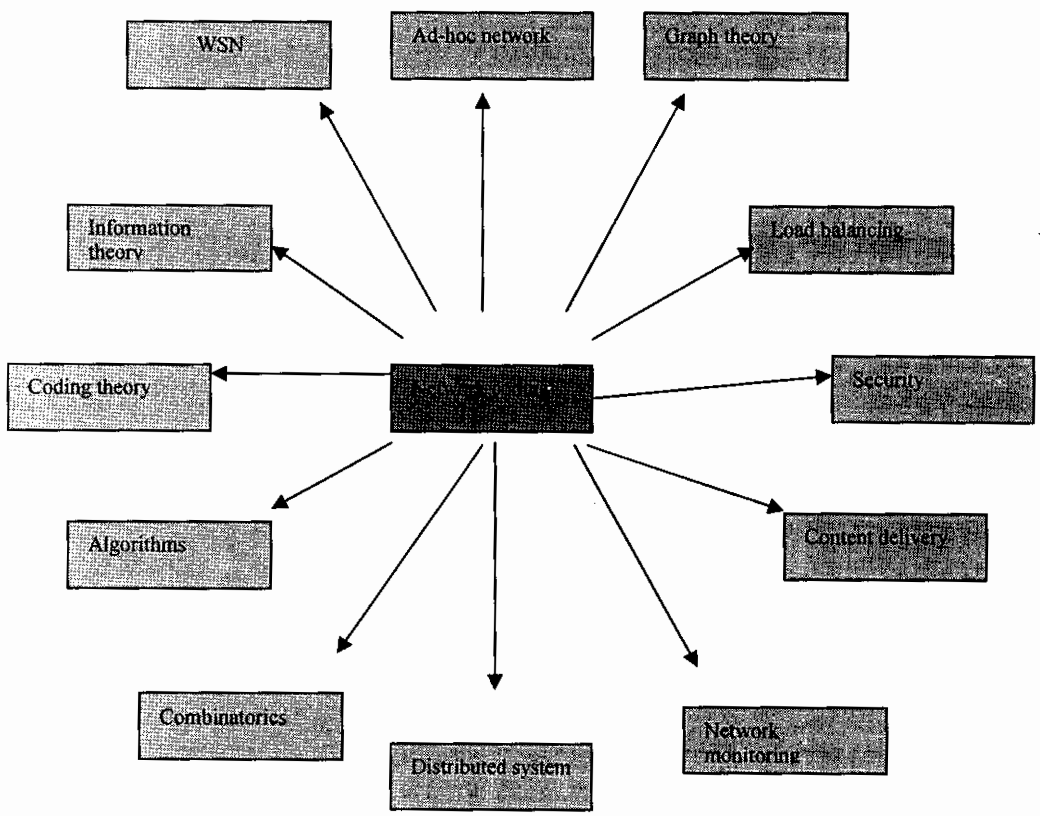

Figure 3.1: Network coding and its interaction with other areas

3.1.1 Canonical Examples of Network Coding

Example 1: Butter-fly network

Source A has packet X_1 and B has packet X_2, where E and F are destinations as given in figure 3.2. In the figure A is directly connected to E but has no direct connection to F so path (A, C, D, F) has to be used. Similarly, node B is directly connected to F but has no direct connection to E so path (B, C, D, E) has to be used. Where the communication paradigm is wireless broadcast. The transmission sequence is numbered. In traditional method as in figure 2.2(a), node A broadcasts X_1 to E and C in step (i), at the end of this step destination E has X_1 but not X_2. In step (ii), node B broadcasts X_2 to C and F, so now destination F has the value of X_2 but not X_1. In step (iii) and (iv) node C broadcasts first X_1 then X_2 to D respectively. In step (v) and (iv) node D broadcasts first X_1 to E and F then X_2 to E and F respectively. Now each destination node has both X_1 and X_2. So the total number of transmissions required is six. Where network coding needs only four transmissions, as given in figure 3.2 (b). In step (1), source A broadcasts X_1 to C and F. In step (2), source B broadcasts X_2 to C and F. Now, destination E has value X_1 but still needs X_2 and destination F has value X_2 but still needs X_1. However, Node C has both values X_1 and X_2, so by using simple network coding node C performs XOR operation and the resulted value (X_1+X_2) is broadcasted to node D in step (3). Node D broadcasts (X_1+X_2) to destinations E and F. At the end, both destinations nodes perform again XOR operation with the corresponding values to get the required value. So that, each destination node has both values X_1 and X_2 in four transmissions (increases throughput) but the processing cost for XOR operation. Further details are summarized in table 3.1 and table 3.2.

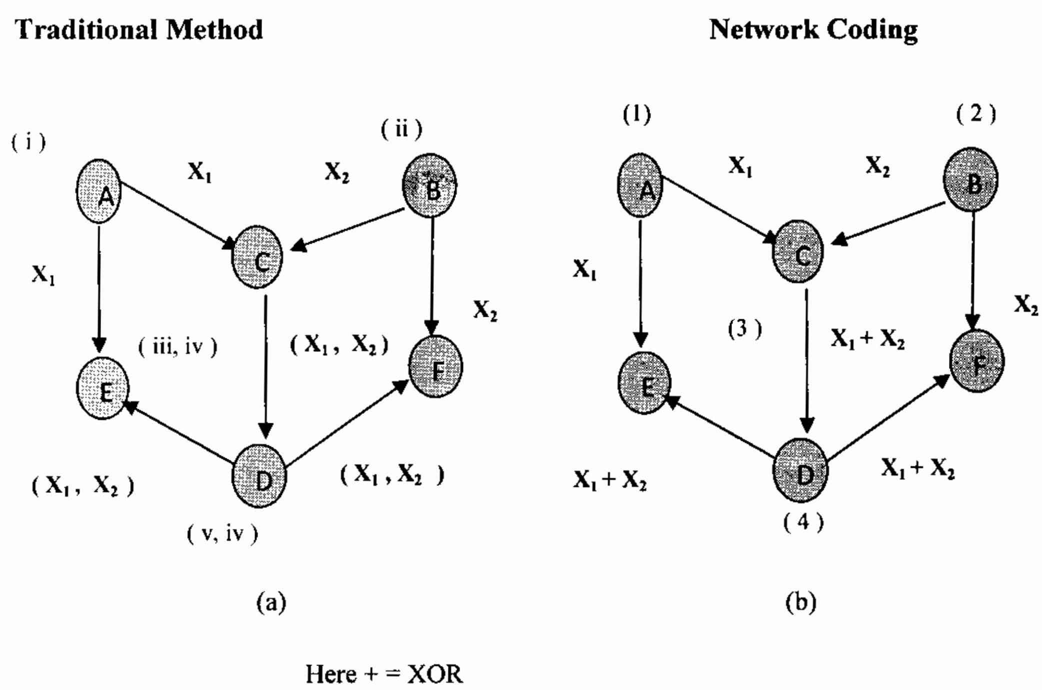

Figure 3.2 Traditional Method (a) vs Network Coding (b)

In traditional method six transmissions are required where in network coding only four transmissions are required. (Throughput increases).

Sequence #	Source	Destination	Value
I	A A	C E	X1 X1
II	B B	C F	X2 X2
III	C	D	X1
IV	C	D	X2
V	D	E	X2
VI	D	F	X1
Total Transmissions	**6**		

Table 3.1 Conventional Transmission Sequence of Example1

Sequence #	**Source**	**Destination**	**Value**
I	A A	C E	X1 X1
II	B B	C F	X2 X2
III	C	D	X1 ***XOR*** X2
IV	D D	E F	X1 ***XOR*** X2 X1 ***XOR*** X2
Total Transmissions	**4**		

Table 3.1 Network Coding Transmission Sequence of Example1

Example 2

A and B are wireless nodes and node S is a relay agent as shown in figure 3.2(a,b). Here, A wants to send "a" to B and B wants to send "b" to A, where A and B are out of rang to each other. Both, traditional and network coding methods are given with sequence numbers. In traditional method shown in figure 3.3 (a), in step i) node A broadcasts "a" to S. In step ii) node B broadcasts "b" to S. The relay agent S broadcasts first "a" to A,B in step iii), then broadcasts "b" to A,B in step iv). So total number of transmissions needed are four. At the other hand, network coding only needs three transmissions as shown in figure 3.3 (b). In step 1) A broadcasts "a" to S and B broadcasts "b" to S in step 1) and 2) respectively. In step 3), S performs XOR operation and the resulted value i.e. a+b is broadcasted to A and B. Now each node has the value ***aXORb***. So, each node gets the corresponding value. So that, network coding only needs three transmissions (throughput gains).

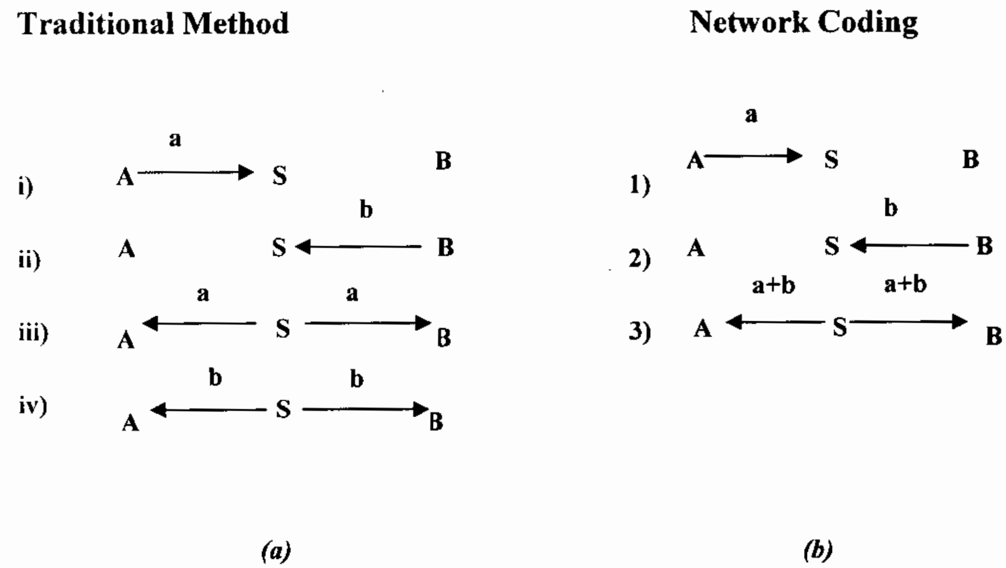

Figure 3.3 Traditional Method VS Network Coding

The examples given above are based only on XOR operation which is the simplist form of network coding, however in practice network coding is different. For practicall network coding the case is a little bit different.The rest of the chpter is dedicated to practical network coding, where network coding can be applied to packets in real networks

3.2 Network Coding

Network coding allows in-network data mannipulation besides blindly forwarding data by the nodes. Here nodes along the path encode the incoming data (packets) into an equal or greater number of outgoing packets. The encoding process involves the combination of incoming packets to produce out going packets. The combination is based on a certain finite field $\mathbf{F_2}^s$. where s denotes the number of bits in a packet which are to be represented by a single number in the finite field. The corressponding value in the finite field is called encoding vector or encoding coefficient. Each node along the path forward the encoded packets and corressponding vectors to next node. Before starting the network coding process all the packets must be of the same size, smaller packets should be padded with zeros [14]. Each node along the way would repeat this process of encoding .For recovery of the orignal packets decoding is needed only at the distination .For decoding the reciever collects the incoming packets into matrix, where Guassian elimination is performed and orignal packets are recovered. To make the encoding and decoding process efficient the packets are grouped together into generations.Only packets from the same generation are encoded and decoded, based on a certain finite field. From the work in [14,15], the matrix size at the destination is an important issue, where large matrix size results in buffere overflow and difficult to manage the decoding pocess. To solve this issue matrix ranks are used as well as the innovative packets concept. To maitain the matrix size manageable only those packets are kept in decoding matrix whose are innovative i.e those packets who can increase the rank of matrix. Further more by keeping the generation size reasonable, the matrix size issue can be handled effiently. As concern to the wireless sensor networks the decoding process is only performed at the sink, which is of course an ideal scenario for efficient performance using network coding.

Further more, there are two important concepts i.e local encoding vectors and global encoding vectors. To decode the encoded packets at the destination successfully each encoded packet transmitted must have its own vectors which are taken from the finite field at the time of encoding. Each node calculates the relavent vectors and forward it to next node along the path. However the concept of encoding vectors depend on the network coding type used. The detail is given in the subsequent sections.

Let the source have packets (M_1, M_2, M_3 . . . . . M_n), when source performs encoding, it will generate (Y_1,Y_2,Y_3 . . . . . . Y_m) encoded packets. The encoding process is given as:

$$Y_i = \sum g_{i,j} \cdot M_j \quad (j = 1,2,3\ldots\ldots\ldots\ldots n,\ \ i = 1,2,3,\ldots\ldots\ldots\ldots m)$$

Here, $g_{i,j}$ is the encoding vector and is taken from $\mathbf{F_2^s}$.

Each node along the path will encode the received packets and broadcast. However, in case of random linear network coding each node will have to calculate the global vectors from the receiving local vectors and send these vectors with the coded data. The encoding process is performed recursively at each node but decoding is only performed at the sink.

Decoding process through matrices:

The relation $Y_1 = g_1M_1 + g_2.M_2 + G_3.M_3 + \ldots\ldots\ldots\ldots g_n.M_n$, is written in the matrix form and solved to find the value of M through Gaussians elimination (like solving linear equation).The matrix is of the form:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ . \\ Y_n \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ . & . & \cdots & \\ g_{n1} & g_{n2} & & g_{nn} \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ . \\ M_n \end{pmatrix}$$

Here the matrix formed by the encoding vectors must be fully ranked (delay).

Note: Here the + sign shows XOR operation.

3.3 Types of Linear Network Coding

Linear coding takes linear combination of received packets. Linear network coding is sufficient both in terms of functionality and efficiency, especially for wireless sensor networks where computational and storage resources are limited. For communication and networks applications the only proper choice is linear network coding to minimize the efforts in managing communications and networks. There are two types of linear network coding. The detail is given next.

3.3.1 Deterministic Linear Network Coding

Deterministic network coding takes linear combination of packets at each node remains the same throughout the network i.e. each node along the path encode the incoming packets with the same combination as done by others. In other words, the encoding coefficients (encoding vectors) remain constant throughout the network [13,18]. This means that every node in the network take same coefficients (encoding vectors) from the finite field. This results into several advantages but at the same time several limitations exist.

Deterministic Network Coding Example

Let the data packets are X_1, X_2, X_3 where

X_1=1001 X_2=1000 X_3=1100

The corresponding finite field is F_2^4, where s=4. So F_2^4= {1,2,3.... 16}

In deterministic network coding each node along the path take same combinations. The encoding vectors are taken from the finite field.

Lets the encoding vectors at $node_1$ are g_1, g_2, g_3 taken from F_2^4 , where in binary form the corresponding vectors are

g_1= 111, g_2= 101, g_3= 110

Lets the encoded packets at $node_1$ are Y_1, Y_2, Y_3 .

The encoding process is

$Y\mathbf{i} = \sum g_{i,j} \cdot X_j$ So that

Y1=1(1001) + 1(1000) + 1(1100)

Y1=1101

Y2=1(1001) + 0(1000) + 1(1100)

Y2=0101

Y3=1(1001) + 1(1000) + 0(1100)

Y3=0001

The encoding process has completed at node1. Node1 will forward the encoded packets to next node where encoded vectors are not required to forward in deterministic network coding. The same process is repeated at each node until the destination node.

Decoding process

The destination node will keep the incoming encoded packets in matrix form and through Gaussian elimination original packets are recovered. By rearranging and Performing row operations to reduce the matrix in reduced echelon form.

$$\begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} . \begin{pmatrix} 1101 \\ 0101 \\ 0001 \end{pmatrix}$$

By R3+R1

$$\begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} 1101 \\ 0101 \\ 1100 \end{pmatrix}$$

By R2+R3

$$\begin{pmatrix} X2 \\ X1 \\ X3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1101 \\ 1001 \\ 1100 \end{pmatrix}$$

Now by R1+R2 and R1+R3

$$\begin{pmatrix} X2 \\ X1 \\ X3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1000 \\ 1001 \\ 1100 \end{pmatrix}$$

So, X1=1000, X2=1000 and X3=1100 are successfully decoded by the sink.

Advantages of Linear Deterministic Network Coding

Since, linear Deterministic network coding uses same encoding vectors. So each node along the path uses the same encoding vectors along the whole path. There are several advantages of deterministic network coding. First, Encoding vectors are known in advance to each node, so no need to append vectors with the coded data. This property of deterministic network coding brings its own advantages such as, decoding matrix size is fixed, also results in to lower packet size because there is no need to forward encoding vectors with the coded data. Furthermore, deterministic network coding also results in lower computational cost because no global vectors calculations are required contrast to random linear network coding.

Disadvantages of Linear Deterministic Network Coding

Deterministic network coding has its own limitations especially in the context of wireless sensor networks. First, deterministic network coding cannot be used for security concerned applications because each node perform encoding with same encoding vectors throughout the whole network. Second, this approach is not flexible where each node is bound to perform same combination for encoding. Third, it is not suitable for fluctuant and wireless environment.

3.3.2 Random Linear Network Coding

Random linear network coding is an efficient and flexible as compare to deterministic network coding. In random linear network coding, the encoding process is performed randomly and independently by each node. Each node along the transmission path performs encoding independently without the knowledge of topology, so can deal with topology changes. In this technique each node has its own local encoding vectors which are randomly picked from a finite field $\mathbf{F}_2^s$. After performing encoding at a specific node, the node has to calculate its global vectors on the basis of local vectors received. The global vectors are placed in the packet's header and forwarded next. The process remains the same for all the nodes along the path. However, decoding is done only at the destination (sink) through matrix inversion. The destination node keep the received packet data and corresponding global vectors in matrix form and then by performing Gaussian elimination the original packets are recovered [13].

Calculation for Global Vectors

When linear random network coding are used, then there is a need to calculate global encoding vectors from the corresponding local vectors at each node along the path. After calculating global vectors, the node appends these vectors in the encoded data and forwards it.

Calculation for global vectors:

$\mathbf{K_i = \sum h_{i,j} \cdot g_j}$

Where **g** shows the received vectors with each packet, **h** shows the local vectors taken from $\mathbf{F_2^s}$ and **K** denotes the global vector calculated (which has to be forwarded with the encoded data). The whole process has explained in the corresponding example.

For random linear network coding same procedure has to be followed as in the example of deterministic network coding. The only difference is the global coding vectors calculation. Every time independent vectors are taken from the finite field by each node along the path.

Advantages of Random Linear Network Coding over Linear Deterministic Network Coding

Random linear network coding has several advantages over deterministic linear network coding. First, in random network coding the choice of encoding vector is independent at each node, which results into flexibility and can deal with topology changes. So random network coding is a better choice for wireless environment where topology changes occur frequently.

3.4 Matrix Size Issue

Random linear network coding uses different vectors at each node along the path, which results into dynamic matrix size. Large matrix size results into memory wastage but better performance (redundancy/reliability), this is overcome by generations. Packets are grouped into sets are called generation. Packets belonging to the same generation are (combined) encoded. If a new incoming packet increases the rank of a matrix then his encoded vectors are added to the vector matrix (innovative packet) otherwise not added because the packet is already available in the matrix (non-innovative).

3.5 Scope of Network Coding

Here the term scope means that at which level network coding can be applied. So there are two levels for using network coding. The first choice for using network coding is at bit level. The bit level usage of network coding can be applied at the physical layer. The bit level usage can results in faster processing. The second choice to use network coding is at packet level. The packet level usage of network coding can be at the higher layers of the communication architecture such as at network layer or at the transport level. At the network layer network coding can be used for routing purposes, while at the transport layer network coding can be for reliability or for security purposes. These choices of using network coding results in broader applications. That's why we are using this technique for error recovery (reliability). Here we are using network coding at the transport layer, which is a best choice (layer) for implementing reliability.

3.6 Theoretical vs. Practical Network Coding

As mentioned earlier in this chapter, network coding has its roots in graph theory and in information theory. Theoretically, network coding works with theorems and assumptions. However, practical networks have their own constraints and properties different form theory. In theory symbols flow synchronously in a graph (network), all edges (links) have equal capacity and complete topology knowledge is known in advance which is used to perform encoding and decoding process etc. Whereas, In practice several things are needed to manage such as Information flows asynchronously in the form of packets, packets transmissions involve random delay and loses, edges capacity are also unknown and time varying, no advance knowledge of topology etc. So need to solve these problems.

We can reduce the gap between theory and practice through different ways. Such as by using buffering asynchronous and random delay issues can be solved. Another way to deal with asynchronous is to use packets and generations, where packet's header carries information about the topology. Beside these, random network coding also allows topology changes. However, opportunistic network coding can also be used to overcome asynchronous and delay issues.

3.7 Applications of Network Coding

The applications of network coding discussed in [13,14,17] are listed in the subsequent sections.

3.7.1 P2P Networks

In point-to-point networks, the most well-known application using network coding is avalanche [17]. Generally, in a peer to peer content distribution network, a server divides large files into smaller blocks. In avalanche, the blocks sent out by the server are random linear combinations of all original blocks. Where, peers sent out random linear combinations of all the blocks available to them. Where clients download files and distribute among themselves. This results in reducing downloading time by 20% to 30% [17].

3.7.2 Wireless Networks

In wireless environment multiple streams and bidirectional traffic are used. Where, analysis has shown that network coding can increase throughput [14]. Network coding exploits the broad cast nature of wireless networks and increase throughput. In multi-hop interaction each router along the path simply *xor* the received packets and forwards to next router. At the destination node original packets can be recovered through decoding process. Similarly, network coding is a good candidate for asynchronous transmissions and where wireless channels are lossy having random delay. Network coding uses opportunistic approach i.e when there is an opportunity to forward packets so packets are forwarded. Moreover, explicit acknowledgements can be replaced by using network coding and errors are detected through the decoding process.

3.7.3 Sensor Networks

Network coding has several applications in wireless sensor networks. One important application is data gathering. Using network coding, data can be gathered and spread over all the nodes which results easy retrieval for the sink node but expensive in memory required at each node. Beside this, network coding can be used in several dimensions in wireless sensor networks like in routing and reliability etc [14].

3.7.4 Network Monitoring

Network monitoring is an important aspect where network is properly monitored to find out about the node failures or data losses. Two approaches are used for this purpose called active monitoring and passive monitoring. Active monitoring (sending probes messages to nodes periodically) is expansive, which can be replaced by Passive monitoring (analyzing packets to find out about the failure nodes etc). However, network coding is a good candidate for network monitoring [13]. In network coding each packet carries information about the others so network assessment is easy and cheaper. Network coding can replace active monitoring because encoding vectors of each packet contain information about the topology and other data packets. Data losses and topology changes can be easily detected. So that, there is no need to send probes messages periodically. This also results into reducing additional network traffic and saving energy. Similarly, network coding also replaces the conventional passive technique. Destination node can easily estimate the remaining data packets by only analyzing the encoded vectors.

3.7.5 Network Security

Generally, network security can be distinguish as security against an eavesdropper that attempts to recover part of data, security against a malicious attacker that attempts to misinform the receivers by modifying data packets and security against jamming attacks. Network coding has the capability to deal with the mentioned issues. An eavesdropper can be protected by using network coding because network coding spreads data into multiple packets so difficult to recover the original data. Beside it for successful decoding to recover the original data, eavesdropper must have all the corresponding encoding vectors. Network coding can also deal with man-in-the-middle attack. By using network coding, an attacker cannot control the outcome of the decoding process at the destination without all other coded packets the destination will receive. So man-in-the-middle-attacks are more difficult. Similarly network coding can deal with jamming attacks. Network coding is actually the mixing of data at the intermediate nodes so in case of packets losses original data can be recovered at the destination, which can reduce the damage of jamming attacks. The use of network coding results into data replicas, so false data can be easily found out and data integrity can be guaranteed [13].

CHAPTER 4

Theory and Model

This chapter presents a detail description of our main objectives and the methodology we are using to achieve these targets. All the required parameters to perform simulation and results analysis have been discussed.

4.1 Reliability Requirements

WSNs are a subset of wireless networks; these are today's widely acceptable and progressive networks. These are becoming mature day by day but their practical implementation is still quite difficult due to challenges in several areas like reliability, routing, energy conservation etc. One of the challenging issue is reliability because the environment in which sensors nodes are deployed are fluctuant where topology changes occur frequently. Besides this node failure is a common issue in WSN, so in these circumstances to ensure data delivery is an important issue.

The reliability goals or objectives those we are considering for WSNs can be thought as an extension to the objectives those are set for traditional networks. According to these, the reliability should be provided in terms of data loss detection and recovery. Whereas, WSNs impose its own constraints in term of energy and processing resources, so these aspects also need to be considered while ensuring reliability. Under the given constraints and reliability requirements our research is a challenging task. The term Reliability can be defined as "That property of a communication system (network) which ensures the successful delivery of data from one point (sender) to another point (receiver)." Where as in the case of data loss during transmission the scheme used to ensure reliability should be capable to recover the lost data. So reliability is a basic property of any communication system, especially in those environments where error rates are high like in wireless sensor networks.

4.2 Problem Formulation

We are going to use random linear network coding to ensure reliability in WSNs. For successful decoding at sink node, the sink must have at least generation size independent encoded packets. Since while using network coding source divides the packets into different generations of same size, so at destination same generation size independent packets are needed. If there is error along the path so some packets may loss. Let there are **N** nodes along the path from source to destination, by considering error i.e. P packets are lost during transmitting from $\mathbf{N_i}$ to $\mathbf{N_{i+1}}$, to decode the packets at sink we simply need to forward **(P+ G)** packets to compensate the error occurrence during transmission. However the lost packets i.e. **P** is unknown and can't be predicted, where **G** is the generation size. So here we need a reasonable amount of redundancy to compensate the error. Therefore we are using a hit and try approach to adjust the redundancy amount at each node independently without any pre calculations. This approach reduces the transmission overhead and computational overhead and also helpful to increase network life time. The concept given above is depicted in figure 3.1.

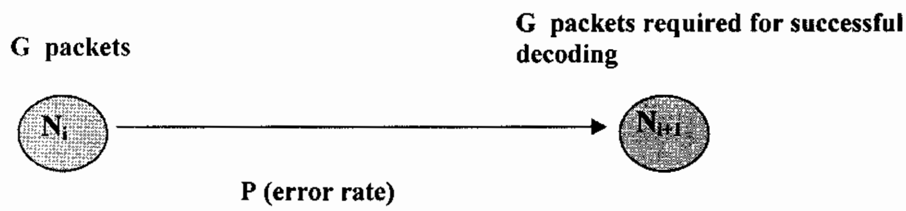

Figure 4.1: Problem formulation (diagrammatically)

Here, we simply need to forward **(P + G)** packets to compensate error amount but errors are unknown and unexpected, so we need to define a reasonable amount of redundancy to ensure reliability. The redundancy amount calculation and other details are given in chapter five.

4.3 Research Objectives

According to the different problematic situations discussed in chapter 1 and in chapter 3, we have set certain goals of this research which have built in solutions for the problems already discussed. The targets and objectives of this research with underlined methodologies are discussed in the subsequent sections.

4.3.1 Ensure Reliability

The major goal of our work is to ensure reliable communication in wireless sensor networks. We are focusing over end-to-end reliability by using random linear network coding. Here random linear network coding is used as a technique. However, we have designed a procedure to calculate reasonable amount of redundancy at each node independently. The proposed scheme is dynamic in nature where redundancy amount is calculated at each node to compensate errors. The redundancy amount does not need any pre-calculations or additional traffic. Further detail has given in chapter 5 of this dissertation.

4.3.2 Consuming Less Energy

Another important aspect of WSNs is their resource constraints like lower processing capabilities and limited energy at sensor nodes. So our objective is not only to ensure reliable communication but at lower cost as well.

4.4 Performance Metrics

As the problem statement focuses over the redundancy amount to ensure reliability in WSN. So the main objective is reliability with the considered redundancy amount. So to analyze reliability of the proposed scheme and comparison with the schemes used in the literature, we are using two performance metrics to analyze reliability and power consumptions by the proposed schemes as in [12]. For reliability analysis the metric is success ratio and for power consumption the metric used is number of transmissions.

Here success ratio is the ratio between total numbers of packets (data) at source and total numbers of packets (data) received and successfully decoded at sink. Success ratio with value 1 is considered as 100% reliability.

Success Ratio = Total number of packets at source / Total number of packets at sink)

Another important aspect of WSN is the transmission overhead or network life time or energy consumption. The transmission overhead or network life time can be measured by the number of transmissions [12]. Here we are considering only the power consumption for transmission i.e. transmission overhead while the computation overhead at nodes is negligible because sensor nodes have to perform only encoding which is a straight forward operation. In the proposed scheme there are no pre-calculations and additional traffic are required which minimize computational overhead and transmission overhead. Where the schemes used in [12] need several pre-calculations such as mean, variance, square root etc to adjust the redundancy amount which further increases transmission and computational overhead, while in the proposed scheme only a single control flow is required to launch the scheme over the network.

Number of Transmissions = (Total number of transmission) / (Total number of packets at source x Average number of hop counts from source to sink)

Here, the total number of transmissions includes all hop-by-hop packet transmissions and retransmissions from source to sink. However, here ***Number of Transmissions*** shows all the additional traffic which results into transmission overhead and additional energy consumption.

CHAPTER 5

Proposed Solution

5.1 Proposed Solution

The literature survey has discussed several approaches to ensure reliability in wireless sensor networks. But due to the resource constraints (energy, processing) of wireless sensor networks, all these conventional approaches are not appropriate. We are using network coding (NC) to provide end-to-end reliability in wireless sensor networks. Similar work has been done in [12],[15]. But our work is completely different in terms of reliability (end-to-end) and redundancy amount calculation.

In our proposed solution, each sensor node along the path between source and destination (sink), encodes the received packets through random network coding to generate an equal or greater number of outgoing packets. For successful decoding, the receiver (sink) must have the same number of packets sent by sender. In case of packet losses we need reasonable amount of redundancy to compensate the errors occurrence during transmission. Our solution provides that reasonable amount of redundancy.

By using network coding different incoming packets are combined to generate equal or greater number of outgoing packets. Similarly, global coding vectors are calculated at each node independently. Each node along the path repeats the same procedure. Decoding is only performed at the destination.

Let the source have packets, $x_1, x_2, x_3 \ldots\ldots\ldots\ldots x_n$. All the packets at sender are divided into groups called generations. Each packet has its own generation id. Packets are grouped together by each node according to their generation ids. Encoding and decoding are performed on generation basis. Packets belonging to the same generation are chosen for encoding and decoding The size of generation is represented by **G.** when source performs encoding, it will generate $y_1, y_2, y_3\ldots\ldots\ldots\ldots y_m$ encoded packets. The encoding is done as:

$$Yi=\sum g_{i,j} .X_j \quad (j=1,2,3\ldots\ldots\ldots\ldots n ,i=1,2,3,\ldots\ldots\ldots\ldots m)$$

Here, $g_{i,j}$ is the encoding vector ,which is taken from F_2^s.

For successful decoding, $m >= n$

To compensate the error rate on the transmission path, we are defining certain amount of redundancy.

$$\text{Redundancy} = \lceil (\text{Incoming Packets} + G)/2 \rceil$$

$$\text{Redundancy} >= G$$

Here, **m > = Redundancy** , So the node has to create **redundancy** amount of encoded packets. Here Redundancy is not **G** but the node has to encode at least new G number of packets.

Calculation of global vectors is performed as:

$$K_i=\sum h_{i,j}\, g_j$$

Where **g** shows the received vectors and $\mathbf{h}_{i,j}$ represents the local vectors picked from $\mathbf{F}_2^s$ and $\mathbf{K}_i$ denotes the global vector calculated. The global vectors are forwarded with encoded packets. Decoding is done only at the sink, where through Gaussian elimination original packets sent can be recovered.

5.2 Proposed Scheme/Algorithm

The proposed scheme is illustrated in the following steps.

Assumptions:

- The routing path is provided by any lower layer routing protocol.
- Generation size is known to every node along the path.

Source Node Behavior:

The source node generates packets, first divide these packets into sets (generations) having same size, where smaller packets can be padded with zeros [13]. Each packet header contains a generation id, which shows the generation with which that packet belongs and encoding vectors. The source performs random linear network coding where local encoding vectors are picked randomly from a certain finite field. But only packets belonging to the same generation are encoded.

Intermediate Node Behavior:

Now, each intermediate node along the routing path performs random network coding, calculating new global vectors from received local vectors, appending these with the packets headers as done at source.

For all the intermediate nodes the redundancy amount is the average of (G + incoming packets) , then a ceiling function is used if the resultant value is not an integer value , so the newly encoded packets must be greater than or equal to the generation size.

Redundancy amount= ⌈ (no. of received packets + gen. size)/2 ⌉

Resultant value = > G

Each intermediate node keeps the receiving packets in a buffer according to their generation id, where redundancy amount is calculated based on the received packets. Newly encoded packets are created and forwarded based on the redundancy amount. All the mentioned operations are based over generation level, where different generations are differentiated through their corresponding generation id.

The same process is repeated at every intermediate node.

Sink Node Behavior:

At sink (destination), the sink buffers the receiving packets and polling the packet according to their generation id. Sink keeps the incoming packets into matrix form organized by generation id. When sink node receives packets equal to generation size in a certain generation. Then decoding is performed through Gaussian elimination. If the sink decodes successfully and recover the original packets sent by the sender, so data has been received successfully without any additional transmissions and energy loses.

In the proposed solution, we have used generation sizes, and packets from the same generation are encoded at each sensor node. The division of packets into generations keeps the matrix size limited and results into efficient decoding.

5.3 Proposed Scheme Testing Scenario

Proposed scheme is illustrated with the help of figure 5.1. There are only three steps i.e source node behavior, Intermediate node behavior and sink node behavior.

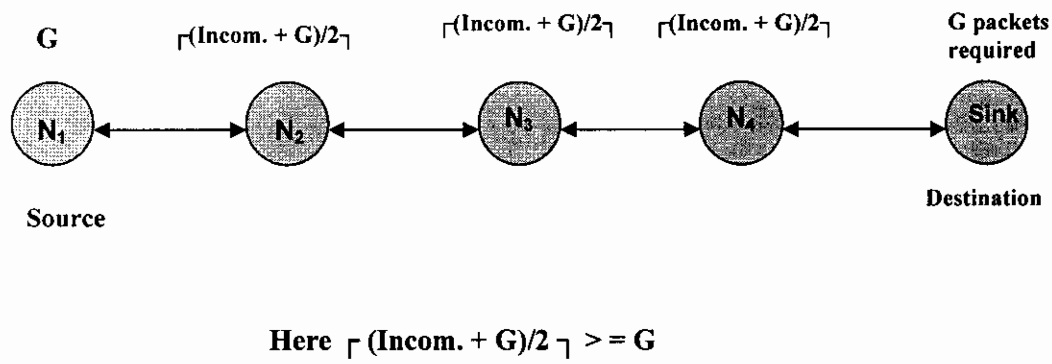

Figure 5.1 Proposed Scheme Testing Scenario

Here Incoming packets mean, those received packets whose belong to same generation. Steps in short are:

Source node behavior:

i) Creates packets and divide into generations of equal sizes. Each generation has its own generation id. Smaller packets can be padded with zeros.
ii) Source creates G packets in a certain generation.
iii) Perform RLNC according to local vectors.
iv) Forward the encoded data and corresponding vectors.

Intermediate nodes behavior:

i) Keep incoming packets into buffer according to generation id.
ii) Calculate redundancy amount as:
   $\lceil (\text{Incoming packets} + G) / 2 \rceil \geq G$
iii) Perform RLNC and calculate global vectors
iv) Append global vectors in packet header and forward.
v) Repeat step i to iv for each intermediate node

Sink behavior:

i) Keep the incoming packets into matrix form according to generation id.

ii) If sink node receives G (generation size) of packets in a specific generation, go to step (iii).

iii) Perform decoding through Gaussian Elimination

5.3 Mathematical Model

To ensure reliability and to recover from losses, the redundancy amount used in this scheme is the average value of the total incoming packets in a certain generation by a node and the generation size. The calculated redundancy amount should also be greater than or equal to generation size as discussed earlier. So that, each node along the path has to produce redundancy amount equivalent packets.

Let there are **N** nodes along the path from source to sink, where nth node is the sink, Let **r** and **r'** represents number of received and outgoing packets in a certain generation of size G respectively, at a node $\mathbf{N_i}$. So each intermediate node $\mathbf{N_i}$ will have to calculate the redundancy amount as the average of incoming packets in a certain generation and generation size. Now, according to the proposed scheme:

$$r_1' = G \quad \text{(source node)}$$
$$r_2' = (r_2 + G)/2$$
$$r_3' = (r_3 + G)/2$$
$$\vdots$$
$$r_n = (r_n + G)/2$$

By adding both sides:

$$r_1' + r_2' + r_3' \ldots r_n' = G + (r_2 + G)/2 + (r_3 + G)/2 \ldots (r_n + G)/2 \quad \textbf{Eq (1)}$$

By assuming that with the given amount of redundancy, each node has G packets in a certain generation. So

$r_1' = r_2$

$r_2' = r_3$

$r_3' = r_4$

.

.

.

$r_{n-1}' = r_n$

$r_n' = r_{n+1}$

Now put these values in equation (1):

Eq (1) =>

$r_2 + r_3 + r_4 \ \ldots \ r_n + r_{n+1} = G + (r_2 + G)/2 + (r_3 + G)/2 \ \ldots \ (r_n + G)/2$

$2(r_2 + r_3 + r_4 \ldots r_n + r_{n+1}) = 2.G + (r_2 + G) + (r_3 + G) \ \ldots \ (r_n + G)$

$2(r_2 + r_3 + r_4 \ldots r_n + r_{n+1}) = 2.G + (r_2 + G) + (r_3 + G) \ \ldots \ (r_n + G)$

As packets at source, $r_1 = G$ so

$2(r_2 + r_3 + r_4 \ldots r_n + r_{n+1}) = r_1 + G + (r_2 + G) + (r_3 + G) \ \ldots \ (r_n + G)$

$2(r_2 + r_3 + r_4 \ldots r_n + r_{n+1}) = (n.G) + (r_1 + r_2 + r_3 + \ \ldots \ r_n)$

$2(r_2 + r_3 + r_4 \ldots r_n + r_{n+1}) - (r_1 + r_2 + r_3 + r_n) = n.G$

$-r_1 + r_2 + r_3 + \ldots \ r_n + 2\, r_{n+1} = n.G$

As packets needed at sink (received by (N+1)th) = G

$-r_1 + r_2 + r_3 + \ldots \ r_n + 2.G = n.G$

$-r_1 + r_2 + r_3 + \ldots \ r_n + 2.G = n.G$ As $r_1 = G$

$-r_1 + r_2 + r_3 + \ldots \ r_n + 2r_1 = n.G$

$r_1 + r_2 + r_3 + \ldots \ r_n = n.G$

According to the proposed scheme source divides packets into generations of equal sizes, and each node along the path calculates redundancy amount as an average of incoming packets and generation size. So the resultant value is generally equal or greater than generation size G (very close to generation size), where for reliability only G packets are required so:

$$r_1 \approx r_2 \approx r_3 \approx . \ldots \approx r_n$$

$$\sum r_i = n.G \qquad i = 1, 2, 3 \ldots n$$

$$n.r_n = n.G$$

$$r_n = G$$

$$r_n = G$$

Hence proved, that with the given amount of redundancy the sink node has generation size, G independent packets for successful decoding. Here n shows the number of nodes along the path.

5.4 Benefits of the proposed solution

We have chosen random network coding to ensure reliability in WSNs that's why; all the benefits of random linear network coding are inherited by our proposed scheme. Beside these, following are the additional benefits as compare to work done in [12].

Our proposed scheme is simple and dynamic in nature. The redundancy amount calculation is too straight forward, because each node has to only analyze incoming packets according to their generation id. Then node has to simply calculate redundancy and produces encoded packets according to the redundancy amount. So there are no complex calculations and interactions with other nodes, which further reduces computational and transmission overhead. The proposed scheme is also dynamic in its nature; each intermediate node will have to calculate redundancy again and again until its resultant value becomes not equal to generation size G.

Each intermediate node along the path has to calculate redundancy amount independently without interactions with other nodes contrast to work in [12]. Similarly, there is no need about the topology knowledge for redundancy calculation.

The scheme given in [12] is centralized in its nature, where each cluster head is responsible for redundancy amount calculation. The cluster head shares this information with its neighbors. In case of cluster head failure, the whole system breaks down. However, our proposed scheme has no cluster head concept. Here, each node independently finds its own redundancy amount and encodes packets accordingly.

CHAPTER 6

Simulation and Results Analysis

Implementation and results analysis is an essential part of any comparative research strategy. We are following same convention to analyze our results and to make comparison with other related work. We have implemented our proposed solution in NECO simulator. NECO is the only open source simulator available, fully dedicated to Network Coding based protocols [19, 20].

6.1 Simulation Details

Chapter 4 "Theory and Model" has discussed research objectives and corresponding performance metrics to analyze our results. We are using two performance metrics i.e success ratio and number of transmission. Success ratio demonstrates reliability, where number of transmissions represents energy consumption or transmission overhead. We are using these performance metrics to make comparison with the work done in [12].

Success Ratio = (Total number of packets at source / Total number of packets at sink)

Number of Transmissions = (Total number of transmission) / (Total number of packets at source x Average number of hop counts from source to sink)

6.1.2 Simulation Settings

Table 6.1 lists simulation details including topology, number of nodes, routing protocol, traffic generator, packet size, finite field size, simulation time, error rates, etc used.

S.No	Parameter	Description
1	Topology	Path Graph
2	Number of nodes	20
3	Routing Protocol	Dikjistra Algorithm
4	Traffic Generator	Deterministic
5	Packet size	50 bits
6	Finite Field Size	F_2^8
7	Error rates	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12, 0.13
8	Simulation time	1000 seconds
9	Total number of packets at source	31
10	Protocol	Proposed Scheme
11	Simulator	NECO Simulator
12	Operating System	Linux (Ubuntu)
13	Scripting Language	Python

Table 6.1 Simulation Details

6.1.3 Assumptions

Two assumptions are made in the proposed solution. It is assumed that the routing path has been provided by the network layer and generation size is known by each node along the path. However, we have used Dikjistra routing algorithm (shortest path) for simulation purpose.

Since the proposed scheme works over transport layer and the performance metrics are environment independent. So the simulations made are supposed to be on transport layer where lower layers are assumed to pass data to transport layer. That's why this abstraction is supposed while performing simulation and implementation. Our main purpose is to carefully analyze reliability and transmission overhead of the proposed scheme. All other relevant issues have been put into our future work.

6.2 Reliability Analysis

We are using success ratio as a performance metric to analyze reliability of the proposed scheme and to make comparison with other schemes. Table 6.2 lists simulation results for success ratio against the corresponding error rates. Here the error rates indicate the packet loss rate or link erasure probability. The link erasure probability is a built-in feature of the simulator used. Success ratio has the maximum value one, which shows 100% successful transmission.

While performing simulation source node creates 31 packets and sent these packets to destination by using the proposed scheme. Simulations are performed against every rate and corresponding success ratio results are listed in the table. Snapshots are given of the simulation results in the subsequent section.

S.No	Error Rate	Success Ratio
1	0.01	0.96774
2	0.02	0.93548
3	0.03	0.96774
4	0.04	0.94900
5	0.05	0.93548
6	0.06	0.93548
7	0.07	0.90323
8	0.08	0.88001
9	0.09	0.90323
10	0.10	0.90323
11	0.11	0.87097
12	0.12	0.87097
13	0.13	0.83871

Table 6.2 Success Ratio vs Error Rate

For each and every error rate, simulations are performed twice. Simulation results in table 6.2 are taken as average values for the corresponding error rates.

Figure 6.1 presents a reliability graph which shows a relationship between the proposed solution and work done in [12]. Both schemes use the same performance metric i.e. success ratio for reliability measurement with similar scale.

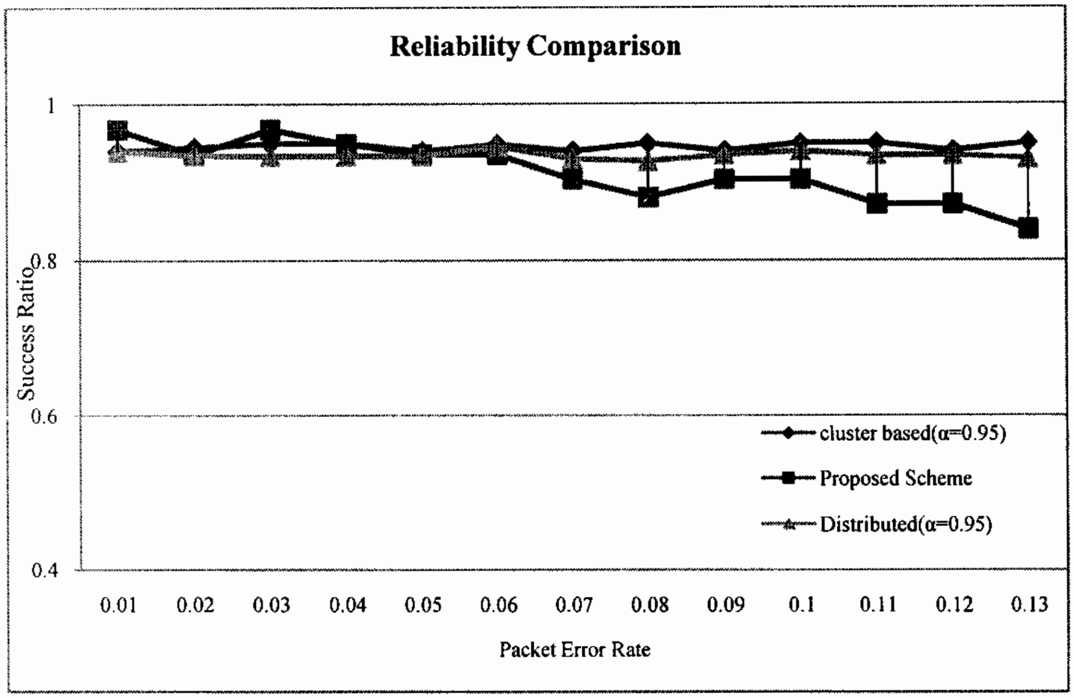

Figure 6.1 Reliability Comparison

We can evaluate the comparative performances of the schemes with the help of figure 6.1. Reliability gains of the proposed scheme are consistent and significant up to error rate .06. It shows that our proposed scheme is more efficient than others up to error rate .06. However, further increase in error rate results into a decline in reliability gains. This is because we are not compromising over transmission overhead, which results into increasing network life time. This can be observed by considering figure 6.1 and figure 6.3 simultaneously.

However, reliability gains of schemes in [12] are consistent about all the way. But, we have found that the given distributed and cluster settings are maintaining reliability with the increasing cost of transmission overhead. So, there exist a trade-off between reliability and transmission overhead. But, proposed scheme also has the corresponding gains in transmission overhead. Under the constraints of WSN, our proposed scheme is much better than others and exactly fulfilling WSN's specific requirements.

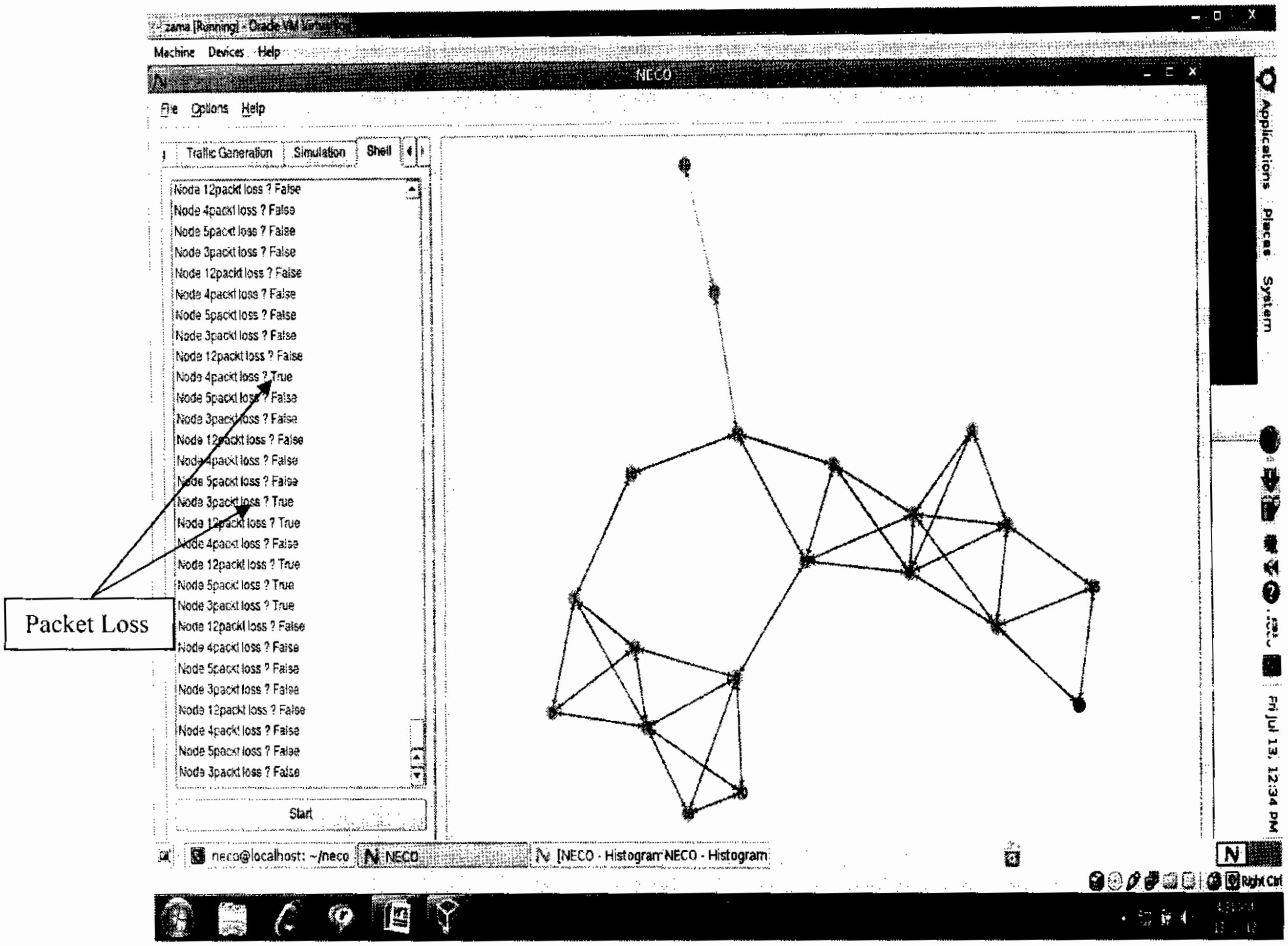

Figure 6.2 Reliability Simulations and Packet Lost Screenshot

Figure 6.2 highlights the packet losses according to the error rates during simulation. However, each entry in the shell area is interpreted as a single transmission. So this is further consider in the calculation of performance metrics i.e reliability and transmission overhead (energy consumption).

The above discussion leads us to conclude that our work is a reasonable approach towards reliable data transmission in WSN by using network coding. However, there are possibilities to make it more efficient for high gains in terms of reliability with lower cost.

6.3 Overhead Analysis / Energy Consumption

We are using Number of Transmissions as a performance metric to analyze transmission overhead or energy consumption of the proposed scheme and also to make comparison with the schemes used in the literature [12]. Table 6.3 lists simulation results for Number of Transmissions against the corresponding error rates.

S.No	Error Rate	Number of Transmissions
1	0.01	0.92742
2	0.02	0.90000
3	0.03	0.87903
4	0.04	0.80645
5	0.05	0.82258
6	0.06	0.83870
7	0.07	0.82258
8	0.08	0.77419
9	0.09	0.74194
10	0.10	0.71774
11	0.11	0.72581
12	0.12	0.70968
13	0.13	0.66935

Table 6.3 Number of Transmissions vs Error Rates

Figure 6.3 shows transmission overhead comparison among our proposed solution and approaches used in the literature. We can easily conclude that the proposed scheme is consistent as compare to others. Proposed scheme line goes down gradually as error rate increases contrasting to other scheme. The main reason is that our scheme has no additional transmissions as error rate increases. The redundancy amount we have used is never based on additional traffic. While the schemes used in the literature require more additional traffic to estimate redundancy as error rate increases. That's why transmission overhead of the scheme used in [12] increases as error rate increases, which further results into additional energy consumption and ultimately affects network life time. Another interesting behavior we can observe about our proposed scheme. That is, as error rate increases transmission overhead decreases as an inverse relationship. This can also be seen in case of reliability as in figure 6.1, Because as packets losses starting to occur our reliability graph also goes down, which ultimately reduces traffic i.e packet lost. So there is a reasonable tradeoff between reliability and transmission overhead. However, collectively our scheme has significant gains up to error rate .06, both in terms of reliability and transmission overhead. Whereas, after error rate .06 reliability performance falls but corresponding overhead gains are still significant as compare to others.

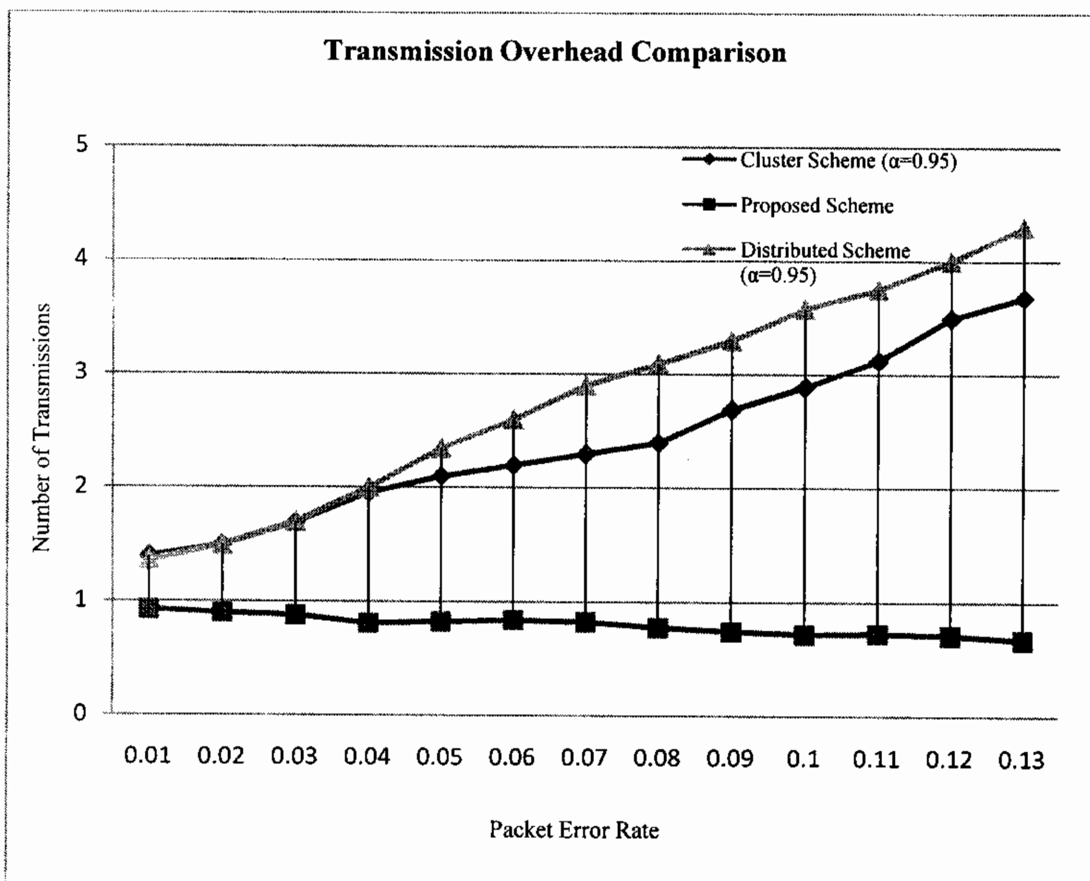

Figure 6.3 Transmission Overhead Comparisons

6.4 Conclusion

We have found that from the last few years among the wireless technologies, WSNs have gained much of the popularity. But, WSNs are subjected to various kinds of constraints like energy, computational capabilities and fluctuant environmental conditions etc. So, there is a crucial of such protocols and schemes which are able to perform well under these conditions. We have design a reliability scheme by considering all these constraints and not compromising over corresponding goals of reliable communication. The proposed scheme is simple and dynamic in its nature. The simulation is made in NECO simulator. It is observed that our scheme consumes less energy with the corresponding reliability gains as compare to the schemes used in the literature. The simulation results confirm our work as a reasonable approach in the domain of WSN. Section 6.2 and section 6.3 have proved our work as a good condidate towards reliable data transmission in WSN. We have come to know that energy conservation is an important and critical issue. Our scheme consumes less energy than others with the corresponding reliability gains. So here we can conclude that the use of random linear network coding in WSN results in to efficient communication in terms of reliability and power consumption.

6.5 Future Enhancements

Enhancements are always well come in any research work. In future, we have a plane to bring enhancements to our work by considering several aspects. The main planed enhancements include implementation over real test beds and to redesign the work done for multi-source destination pairs.

References

[1] I.F Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayyirci. "Wireless Sensor Networks: A Survey" , Computer Networks 38, pp 393-422, Elsevier Science, 2002.

[2] G.J. Pottie, W.J Kaiser, "Wireless Integrated Networks Sensors", Communication of ACM, 43 (5), pp 551-558, 2000.

[3] E.Shih, A. Sinha, A. Wang, S. Cho "Physical Layer Driven Protocol and Algorithm Design for Energy Efficient Wireless Sensor Networks", Mobicom' 01, pp. 272-286, ACM 2001.

[4] A. Bakre, B.R, Badrinath "Indirect TCP for Mobile Nodes" Proceeding of 15th international conference on distributed computing systems, pp. 136-143 ACM 2000.

[5] K.Sohrabi, J. Gao, G.J pottic, "Protocols for Self-Organization of a Wireless Sensor Network" IEEE personal communication, pp. 16-27, 2000.

[6] A. Woo, D. Culler. "A Transmission Control Scheme for Media Access in Sensor Networks", Proceeding of ACM mobiCom'01, Rome, Italy , pp. 221-235, 2001

[7] Peterson and Davie, " Computer Networks: A System Approach " , 3rd edition 2003.

[8] Kurose, James F, kieth W. Ross. "Computer Networking: A Top-Down Approach".P.232.

[9] Tanenbaum, Andrew S. "Computer Networks, p.223, 2003.

[10] Charles Wang, Dean sklar, Diana Johnson "Forward Error-Correction Coding", The aerospace magazine of advances in aerospace technology. 2001

[11] Johny Chen, peter Druschel, Devika Subramanian, "An Efficient Multipath Forwarding Method", pp. 1418-1425 vol.3, Infocom, IEEE1998.

[12] Ting-Ge, Chih-Cheng, Cheng-Fu Chou. " On Reliable Transmissions by Adaptive Network Coding in WSN ", IEEE ICC 2009 .

[13] C Fragouli ,J . Y,L.Boudec and J. widmer " Network Coding: An Instant Primer" International Conference on Computational Theory, IEEE 2001.

[14] R. Ahlswede, N.cai ,S.R.Li and R.W Yeung "Network Information Flow" international conference over wireless communication,IEEE 2000.

[15] Zheng guo, Peng xie, Bing Wang "On Applying Network Coding to Under Water Sensor Networks" 3rd international conference over wireless communication ACM, Seatle USA Aug 2006.

[16] Dong Nguyen,Thinh Nguyen, Bella Bose. "Wireless Broadcasting using Network Coding" Computer Networks 56, pp. 2512-2538, Elsevier,ACM 2008.

[17] Avalanche: File Swarming with Network Coding. http://research.microsoft.com/pablo/avalanche.aspx

[18] Nicholas J.A Harvey, David R. Karger, Kazuo Murota "Deterministic Network Coding by Matrix Completion" International conference over theoretical computing, IEEE, 2004.

[19] Diogo Ferreira, Luisa Lima, Joao Barros, "NECO: Network Coding Simulator", Network Architecture and Design – Network Communications; C.2.2 [Computer Communication Networks] : [Simulation and Modeling]; General, ACM 2010.

[20] "NECO: NEtwork COding Simulator," http://www.dcc.fc.up.pt/ neco/wiki.