Adaptive Backbone and Link Correlation based Data Transmission Schemes for Wireless Sensor Networks

by

Thilina Nuwan Weerasinghe

Supervised by
Professor Frank Yong Li

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information and Communication Technology

Department of Information and Communication Technology
Faculty of Engineering and Science
University of Agder

Grimstad, May 26, 2015

Abstract

One of the main challenges faced by modern wireless sensor networks (WSN) is rapid energy depletion of individual sensor nodes. In many applications, sensor nodes are deployed in outdoor environments and it is difficult to replace or recharge node batteries. Depending on network topology and the transmission schemes implemented, certain nodes could have higher energy consumption compared with other nodes. In highly unbalanced load distributions this could lead to early total energy depletion of a critical node in the topology resulting in what is known as an energy hole in the network. Considering these factors it is vital that the transmission schemes employed in WSNs are designed to address these energy constraints. At the same time transmission schemes should be able to achieve the expected levels of reliability and latency requirements of the applications.

This thesis proposes one basic scheme (BS) and an advanced scheme (AS) considering aforementioned requirements for efficient and reliable data transmissions. AS scheme would be employed if 100% reliability is required in the network while BS could be employed otherwise. The proposed schemes utilize the concept of wireless link correlation to minimize the number of transmissions and hence to reduce the energy consumption. The BS proposes 3 main components, backbone selection criteria, hop count based back-off algorithm and the selective re-transmission (SR) phase . The backbone selection method extends the ideas from connected dominating set (CDS) to achieve more balanced load distribution and the hop count based back-off algorithm aims to reduce the number of intermediate re-transmissions. In addition,the selective re-transmission (SR) helps to improve the reliability of BS.

The proposed schemes are implemented along with two other transmission schemes in MATLAB based environment and extensive simulations are carried out for performance evaluation and comparison. The results show that BS and AS are capable of achieving higher level of reliability with comparatively low levels of energy consumption and number of transmissions. Furthermore, BS and AS have better performance in weak correlations than their counter parts like Collective Flooding (CF). The AS is capable of reaching 100 % reliability in all scenarios for a slightly higher energy consumption compared with BS. The results reveal also the trade off between energy consumption and reliability in a WSN. Overall, the proposed schemes could contribute towards the energy conservation of the network while providing a higher reliability when required.

Keywords: Wireless sensor networks, Link correlation, Data transmission schemes, Reliability, Energy consumption, Implementation and simulations, Performance comparison.

Preface

This report is based on the work done for IKT590 Master thesis, which is a 30 ECTS credit module on the Master program in Information and Communication Technology at the Faculty of Engineering and Science, University of Agder, Norway. The research work started from the second week of January 2015 and the thesis was submitted on 26th May 2015.

The completion of this thesis would not be possible without the kind support of several people. I would like to specially thank my supervisor professor Frank Yong Li for his guidance and patient support through out the thesis work. I also take this opportunity to thank Professor Morten Goodwin for his coordination of the thesis project. My sincere gratitude goes to Mr. Indika Anuradha who is a PhD Research Fellow at the department of ICT in University of Agder, for his support to complete this thesis. At last but by no means the least, I would like to thank my wife, my parents and my sister for their love and support through out my Master's studies.

Thilina Nuwan Weerasinghe
Grimstad
May 26, 2015

Contents

List of Figures

List of Tables

List of Abbreviations

ACK	Acknowledgment
AS	Advanced Scheme
BS	Basic Scheme
CDS	Connected Dominating Set
CF	Collective Flooding
CP	Coverage Probability
CSMA	Carrier Sense Multiple Access
DCF	Distributed Coordination Function
ER	Explicit Re-transmission
LC	Link Correlation
LCT	Link Correlation based Transmissions
MANETs	Mobile Ad Hoc Networks.
NLCT	Non-Correlation based Transmissions
OR	Opportunistic Routing
PRR	Packet Reception Ratio
SD	Standard Deviation
SR	Selective Re-transmission
TDMA	Time Division Multiple Access

TE	Transmission Effectiveness
UML	Unified Modeling Language
WSN	Wireless Sensor Networks

Chapter 1

Introduction

This thesis focuses on proposing novel data transmission schemes for wireless sensor networks (WSN). In this chapter, we initially provide some background information about the WSNs and the motivation for the current thesis. It is followed by the problem statement and research approach describing the research methodology that was followed. After that, an outline of the thesis chapters is provided.

1.1 Background and Motivation

WSN is a collection of sensor nodes which are capable of collectively monitor and transmit information about a variety of phenomena of interest. Sensor nodes are battery powered devices that are capable of sensing physical quantities. In addition some sensor nodes have the capability of storage, processing and aggregation of data . Sensor nodes observe its environment and takes periodic, or action triggered data and measurements. These information is forwarded to one or more sink nodes that have the duty of collecting data and interacting with the outside world. Sensor nodes can have wide range of applications including environmental, civilian , military and commercial areas [1]. The research interest on WSNs has been active for a longer time. It still has a huge interest due to ever increasing demand for sensor network implementations [2]. More sophisticated sensor nodes that are capable of meeting demanding requirements, and the affordability for mass production at a simpler cost has contributed towards this rise in implementations.

The principals of traditional wireless network design can not be directly applied for WSNs. The traditional communication networks are built to support a variety of users with different objectives. Hence, a multi-service paradigm is used in traditional networks. However, due to following characteristics, the traditional approach is not suited for WSNs [1].

- WSNs normally consist of larger density of nodes which needs sensors that are cheap and easy to deploy.
- Due to different applications there is the need for application specific sensor nodes.
- WSNs have limitations in energy, processing power and memory which needs to be addressed when designing protocols.
- In WSNs, all nodes collaborate towards achieving a common objective.

Hence, WSNs needs specific protocols to address the above requirements. When considering the data transmission process, it is vital to have transmission schemes that addresses the energy limitations of a WSN. In most applications sensor nodes are deployed in remote places. Due to this reason, nodes need to operate for a longer period of time without routine access or maintenance. However, the sensor nodes are mostly powered with small batteries which are needed to be replaced when depleted. Thus, it is important to consider minimizing the number of transmissions and energy consumption, when designing transmission schemes. At the same time the reliability of transmissions are needed to be kept at desired levels. Attempting to minimize the transmissions without considering the impact on the reliability of data transmission could degrade the packet delivery to desired nodes. Especially, if a certain level of reliability is desired for an service, this requirement should be given a higher priority than reducing the energy consumption levels. Hence, the trade off between energy consumption and reliability should be considered, and proper balance should be maintained when designing a transmission scheme. The requirement of transmission schemes with lower energy consumption and high reliability, motivated me to work on novel data transmission schemes that could address the above mentioned issues.

1.2 Problem Statement

Transmission schemes could be broadcast, multicast or unicast depending on the application requirements. While broadcasting based protocols are used for covering all nodes in a network, multicasting based protocols are popular for covering a group of them. To provide these types of services, different transmission schemes are proposed. The traditional multicast based protocols require explicit or implicit acknowledgments (ACKs), based on the type of service under consideration. These ACK required transmission schemes tend be more reliable considering the delivery of a data message [3]. However, due to increasing number of data and ACK transmissions, these types of sensor networks could consume higher energy and face also broadcast storm problems [4]. Correspondingly, backbone based concepts like connecting dominated set (CDS) [5] are employed to minimize energy consumption by limiting the number of nodes that transmit messages through a backbone which represents all nodes in the network. However, CDS techniques could be still prone to energy hole problems [6]. When a single node performs a higher number of transmissions, it will eventually drain its energy, making a hole in the network of nodes. Hence, the requirement for improving existing transmission schemes and propose new schemes that has better performance has been a major research interest for some time now.

Traditionally, the wireless links in WSNs are considered to be independent of each other for data transmission. However, recent studies [7], [8] have experimentally revealed the existence of link correlation among wireless links. That is, when a transmitter has multiple receivers, the links between the transmitter and the receivers are indeed correlated. Also, [7] has proposed a link correlation based flooding scheme called collective flooding (CF)that minimizes the number of transmissions and energy consumption using collective ACKs and dynamic forwarder selection methods. However, from our analysis on CF, we observed that CF's performance degrades with weak correlation. Also, CF mainly rely on coverage estimation methods to predict the delivery of data packets. These estimations sometimes could be based on obsolete link information which could cause lower delivery percentages specially at leaf nodes of a topology. Furthermore, the conditional packet reception probability, (CPRP) which is used by CF for collective acknowledgments will not give an accurate picture of the link correlation levels in certain bit combinations as further discussed in Ch. 2. These observations influenced us to propose novel data transmission schemes utilizing the concept of link correlation.

1.3 Research Approach

Initially, a literature review was performed on areas relevant to the thesis topic. The general concepts about WSNs, existing data transmission schemes and the research work related to concept of link correlation was studied.

After that, the design process of transmission scheme begun. Major components related to transmission schemes were analyzed and designed separately. In this design process, some of the existing concepts studies from the literature was extended. In addition, some novel concepts were also proposed to address different design challenges.

Next phase was the implementation stage. The proposed schemes were implemented using MATLAB software. The performance evaluation parameters were identified and the evaluation scenarios were defined. At the same time, two more transmission schemes were implemented for the purpose of performance comparison. The implemented schemes were tested with different inputs and their behavior was analyzed. The troubleshooting of implemented schemes were performed in parallel with the testing phase.

After the implementation and testing, the performance evaluation and comparison was performed. The implemented schemes were simulated for defined evaluation scenarios and the results were obtained and plotted. A detailed analysis and comparison of the results was performed. A detailed discussion was done based on the obtained results and observations. And conclusions were obtained. In parallel to all these processes the thesis writing was also carried out.

1.4 Report Outline

The rest of the thesis is organized as follows.

- In Ch. 2, we present the enabling techniques and the background work for this thesis. Some relevant concepts of WSNs and data transmission schemes are presented here. In addition, the concept of link correlation and some of its applications in WSNs are discussed. Our previous work done related to the concept of link correlation is also presented.
- Ch. 3 presents the proposed transmission schemes.A detailed description about the concepts and algorithms in the proposed schemes are presented in this chapter.
- Ch. 4 consists of the implementation details and performance parameters. The flow charts of all implemented transmission schemes are presented and the operations are explained. In addition, the performance parameters considered and their deriving methods are explained.
- In Ch. 5, first, the simulation scenarios are described. Then, the obtained results are presented with help of graphs for each scenario and for overall results. An explanation of the behavior of results is also provided here.
- Ch. 6 consist of discussions about the results obtained, and other observations related to the thesis work. It also presents further discussions on methods to improve the proposed schemes based on our experience.
- Finally Ch. 7 concludes the report with a conclusion and a list the contributions. Future work recommendations that were identified are also presented.
- The report also includes a list of references and appendices. In Appendices some of the MATLAB codes used for the simulations are presented.

Chapter 2

Enabling Techniques

In order to propose novel transmission schemes, it is important to study the existing literature, relevant to the area of interest. This chapter presents a brief background on the enabling technologies for this thesis work. Initially, the relevant concepts in WSNs are presented. Then, the concepts related to wireless link correlation are explained in detail. A summary of our previous work related to wireless link correlation based transmission schemes, is also presented.

2.1 Relevant Concepts in WSNs

WSNs has a wide range of research topics starting from component level, system level to the application level [1]. Since our interest is in designing transmission schemes, we are more concerned about the topics relevant to the system level, which covers the concepts about data transmissions. In this section, we will focus our attention to some of the relevant concepts of WSNs in system level.

2.1.1 Cooperative Communication in WSNs

In a cooperative communication system, each use performs the duties of transmitting data as well as acting as a relay or a cooperative agent for another users transmission. The Fig. 2.1, represents an example of cooperative communication between two nodes trying to communicate with a bast station.

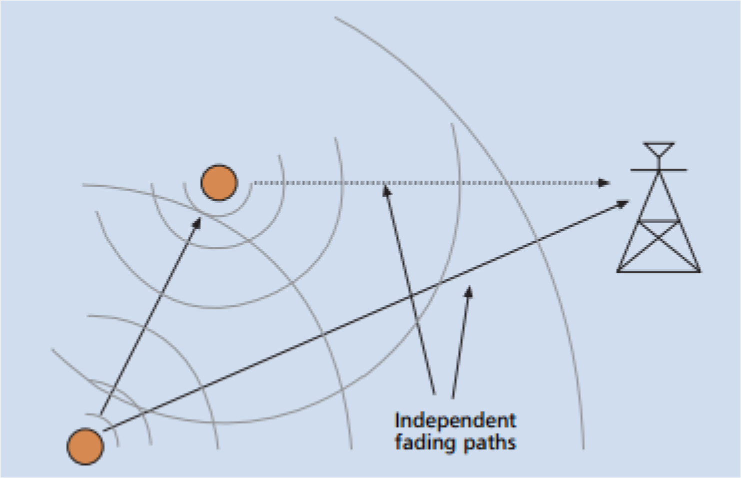


Figure 2.1: Example of a cooperative communication scenario [9].

There could be several reasons for cooperation. One obvious reason is the limitations in reaching the destination node. In a sensor network, if the sender node is not capable of reaching the destination node, it has to rely on the neighbor nodes cooperation to move the data packet forward. Depending on the network topology many nodes may need to cooperate for this data transmission phase. In this account, the sensor nodes are cooperating and acting as an intermediate relay for the source nodes transmissions. Cooperative communication could also occur when the original sender can reach the destination node. In such situations cooperation is used in the aim of reducing the overall transmission energy involved in the process. With the help of cooperation, the sender can transmit with reduced level of power which will help to lengthen its life time. Cooperative communication is also utilized to combat the effects of channel fading by exploiting the diversity of the involved links [10]. In most situations, WSN are designed to rely on cooperative communications in order to transmit or forward data towards the destination nodes. Higher density of node distribution , lower transmit powers and various application scenarios also encourage WSNs to operate using cooperative communications.

2.1.2 Topology Control by Virtual Backbones

Topology control is an important parameter to be considered when a novel transmission scheme is proposed. Due to the wireless nature of a WSN, the nodes can potentially change the dynamics of network topology by adjusting their transmission power levels or by selecting a specific set of nodes to forward their messages. Hence, topology control is important to maintain network connectivity and to prolong network life time and throughput [1]. Geometric based solutions and hierarchical based solutions are used to address the challenge of topology control in a WSN. In the following, we will consider about the concept of virtual backbones which is a type of hierarchical solution for topology control.

In hierarchical based topology control methods, only a subset of nodes are involved in data forwarding to other nodes. This subset of nodes serve as backbone routers and many routing algorithms have been proposed using this concept [1], [11], [12]. This type of selected set of backbone nodes are often known as virtual backbone nodes. Although there are several types of virtual backbone selection concepts, we will consider about one main virtual backbone method which is the CDS [11]. For the definition of CDS, a dominating set is defined as follows. A subset S of V is a dominating set if each node u in V is either in S or is adjacent to some node V in S. Correspondingly, a subset C of V is a CDS if C is a dominating set and C induces a connected sub graph [1]. Finding the minimal CDS is considered as a NP hard problem. Hence, distributed methods are used to find the minimal CDS.

There are a large number of algorithms proposed in literature for the construction of CDS [13]. These includes centralized methods as in [14], [15] and distributed methods presented in [16], [17]. The Fig. 2.2 presents an example procedure for CDS construction.

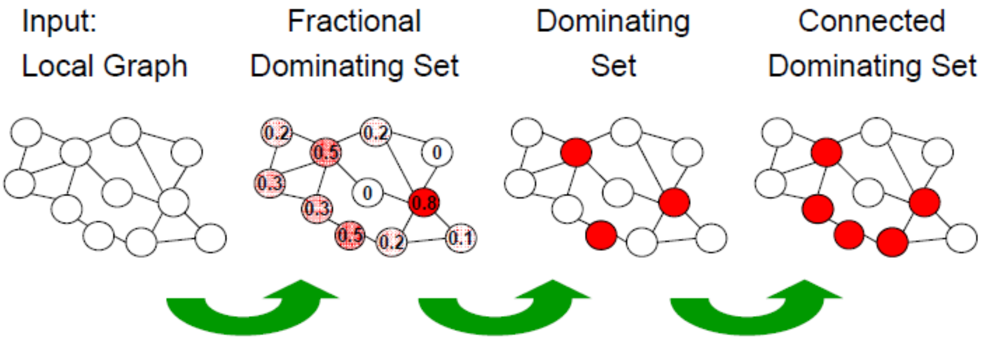


Figure 2.2: CDS construction example [18].

The fractional dominating set is obtained from the input local graph by deploying a distributed linear algorithm. The values mentioned in the nodes are based on some pre-defined metric, like connectivity. Then, the dominated set is derived from the fractional dominated set by using a probabilistic algorithm. Finally, the CDS is derived by connecting the dominated set by a tree of bridges [18].

The CDS nodes once established are the only nodes in the topology who engaged in data retransmission process. Thus, the number of forwarding nodes in a network will be greatly reduced, and this will result in reducing the energy consumption levels of overall WSN.

2.1.3 Medium Access Control Protocols in WSNs

In a WSN, a medium access control (MAC) protocol defines the mechanism of controlling access to the shared communication medium. MAC protocols can be designed depending on the requirements of the sensor network. The main requirements of a MAC protocol are presented below [1].

- Energy efficiency: As discussed before, due to the nature of WSNs each component needs to be critical about the energy consumption. For MAC protocols also this remains a major concern. MAC protocols should be designed keeping the requirements of energy efficiency in mind.
- Scalability: WSNs could consist of large number of nodes with high density. Hence, a MAC protocol needs to be able to perform properly irrespective of the network size or node topology.
- Fairness: When multiple nodes need to access the channel, the channel allocation should be done in a fair way so that each node gets the opportunity it deserves. However, depending on the application requirements and transmission scheme, fairness may always not be implemented. Certain nodes may get a higher priority to access the channel. Since all nodes are acting on behalf of a common goal, this kind of behavior may not be a major concern in certain protocols.
- Latency: Latency should be considered depending on the application requirements. However, the trade-off between latency and energy conservation needs to be considered in MAC protocol design.
- Bandwidth utilization: A MAC protocol needs to efficiently use the bandwidth in order to maximize the performance of a WSN.

MAC protocols can be divided in to two main categories [1]. One category consists of schedule based protocols. In these types of protocols, the time division multiple access (TDMA) scheme is often used. In TDMA based protocols, one node in the topology will distribute a TDMA schedule during the network initialization period. This helps nodes to be aware of their transmission times and hence these protocols are considered to be collision free. LEACH [19] is an example for such a protocol that uses the schedule based MAC mechanism [1]. The second category consists of contention based protocols. Most of these contention based protocols are carrier sense multiple access (CSMA) protocols. In CSMA protocols, the sensor nodes sense the carrier before transmitting data to avoid collisions. The distributed coordination function (DCF) [20] is an example of a MAC protocol that uses contention based MAC mechanisms. This protocol is mainly used in mobile ad hoc networks (MANETs) [1].

2.1.4 Data Aggregation in WSNs

Data Aggregation also referred as fusion, is a mechanism used for reduce the energy consumption and traffic load of a WSN. In data aggregation, data is processed at node level and instead of sending all data to the sink, only processed data is sent to the sink node [1]. For example when the task is to measure and report the average temperature, data aggregation will allow nodes to combine their measurements and send one processed data packet. This will reduce the number of packet transmissions and traffic load in the network. Opportunistic aggregation and clustering are two approaches used in aggregation [1]. In opportunistic aggregation,when multiple packets reach the same node, they are opportunistically aggregated. Directed diffusion proposed in [21] is an example for this type of data aggregation. Clustering based approaches divides nodes in a network in to different clusters. The packet aggregation happens at the cluster head and then sent to the sink node. The LEACH [19] is an example for distributed clustering protocol.

There are 3 types of aggregation architectures [22].

- Centralized: This is the simplest form of aggregation. A central node aggregates the results collected locally and sent by nodes around it. However, in this architecture, the work load is unbalanced and concentrated at the central node.
- Decentralized: In decentralized systems, aggregation happens locally at each sensor node. The information gained by observations and neighbor information exchange is used for this aggregation process.
- Hierarchical: In this architecture, the nodes are partitioned in to different levels. The sensing nodes are at level 0 and the base stations are at the top level. The reports are sent from the lower levels to the upper ones. Hierarchical architecture enables to achieve a balanced load distribution.

2.1.5 Data Transmission Schemes in WSNs

There are a large number of transmission schemes proposed for WSNs. In different research articles these schemes are identified by different names such as data transmission schemes, flooding protocols, routing protocols and data dissemination schemes. Depending on the application requirements and operating layer, a specific name could be denoted for the above mentioned schemes. In general, all these schemes define a mechanism of data transmission inside a WSN for different purposes. Hence, in this thesis, we will refer these schemes as data transmission schemes for the sake of convenience.

These schemes could be divided in to probabilistic approaches and deterministic approaches [7]. In probabilistic approaches, the receiving node forwards a packet with a certain probability value P. The method used to determine this value for P could vary with different transmission schemes considered. Generally, this decision on P is an informed decision. The required information is either gathered by local observations at the corresponding node, or through information exchange with neighbor nodes.

The probabilistic schemes such as [23] uses a predefined probability value for all nodes in the topology to re-transmit a receiving packet. In [6], the P value for each sensor is determined according to the link quality values of its local observation and transmission channels. However, these schemes could cause lot of redundant transmissions due to the simpler P determination methods. To avoid this, a distance and location based transmission scheme is proposed in [24]. In this scheme, the position or the area of sensor nodes are used to minimize the redundant transmissions [7]. If ACKS are used in a probabilistic scheme, it will provide a better reliability for the cost of higher energy consumption. Similarly, the schemes without ACKs are capable of providing better energy consumption and transmission results for the cost of lesser reliability. Moreover, gossiping based transmission schemes such as [25] is capable of reaching high reliability levels with multiple message exchanges.

In deterministic approach based schemes, a fixed node from the CDS set is determined as the data forwarding node. The CDS determining process is explained in Subsec. 2.1.2. These types of fixed forwarder based schemes has been proposed in Double-covered broadcast [26], where high reliability is achieved with broadcast redundancy.

For all these transmission schemes, managing the trade-off between energy consumption and reliability seems to be a major challenge. Hence, when proposing novel schemes, it is important to consider methods for managing this trade-off as per the requirements specified by the applications.

2.2 Wireless Link Correlation based Transmissions

Until recently, for most of the research work, wireless links were considered to be as independent from each other. However, [8] and [7] authors have experimentally proven the existence of wireless link correlation from a single sender to several receivers. These experiments proved that transmissions from a single transmitter to multiple receivers are correlated. This observation can have serious implications on wireless communication systems as the concept of correlation can be used to improve the existing systems in different ways.

2.2.1 Existing Correlation based Solutions for WSNs

The link correlation based research work for WSNs can be divided in to two main categories. They are the link correlation based multicast/broadcast and link correlation based unicast. For link correlation based multicast/broadcast schemes, [7] introduces the concept of collective flooding (CF) which uses the CPRP metric in order to calculate the conditional reception on two receivers based on the link correlation levels. CF uses CPRP to collective acknowledgement and also has dynamic forwarder selection methods which reduces the energy consumption in a WSN. The concept of CF will be further explained in the following section. In [27] the concept of correlated flooding is introduced in order to solve the ACK implosion problem and to minimize energy consumption of traditional flooding. In correlated flooding the packet receptions at multiple correlated nodes are acknowledged by a single ACK. The design and implementation of 'CorLayer' which is a general supporting layer for energy-efficient reliable broadcast is presented in [28]. This corlayer is capable of blacklisting poorly correlated wireless links in order to improve the transmission efficiency of the network. In [29] a multi-packet flooding protocol called SYREN is presented that exploits the synergy among link correlation and network coding. SYREN claims to reduce the number of redundant transmissions and capable of reaching higher reliability.

When considering link correlation based unicast methods [30] present a link correlation aware opportunistic routing (OR) scheme which exploits the diverse low correlated forwarding links, for the candidate selection of OR. Here, considering link correlation for forwarder set selection results in more diverse link sets in the forwarder set and hence reduces the number of transmissions required for the unicast. In [31] the concept of sender based correlation is introduced. In this paper, both spatial and temporal correlation is utilized for the sender based correlation calculations. The correlation in this case works from multiple senders to a single receiver. This concept is utilized by them to develop an energy efficient unicast scheme.

Most of the research work on link correlation have been influenced by the two initial research done in [7] and [8]. In order to get a better understanding of these two research works which will be utilized in Ch. 3, some main concepts of their work is presented here.

2.2.2 Collective Flooding

CF uses periodic hello message exchange between neighbor nodes to establish sequence of received results at each node, which is called as the bitmap. As shown in Fig. 2.3 the nodes S , N1 and N2 exchanges hello messages periodically. The results of last 4 hello messages transmitted by each node is recorded by the receiving node as an array of '1's representing a successful reception and a '0' for a failed reception. Fig. 2.3 shows the bitmap of S's transmissions as recorded by N1 and N2. After each 4 hello messages, the resulting bitmap of the previous four hello messages is exchanged between each neighbor node. This establish bitmap is used to calculate link quality levels and the CPRP values for each link.

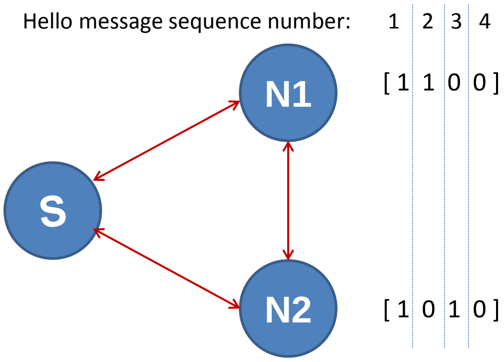


Figure 2.3: Bitmap establishment process using hello messages.

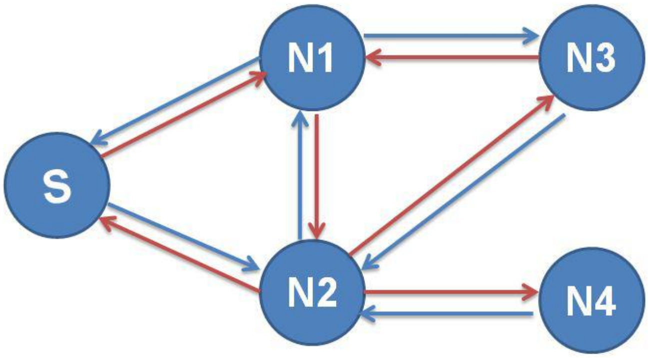


Figure 2.4: Bitmap establishment process using hello messages.

Each node uses this bitmap information to calculate link quality levels, the CPRP values for each link, as well as the coverage probability (CP) which is the probability for a node to cover its neighbor node from the current and the previous transmissions. For example, in Fig. 2.4, S transmits initially with the purpose of making all nodes in the network covered by each of its transmissions. Initially, all CPs are zero. After the transmission, S updates its sending CP on its neighbors N1 and N2 by using Eq. (2.1).

$$CPu(k) \leftarrow 1 - (1 - CPu(k)) * (1 - L(u, k)), \tag{2.1}$$

where $(1 - CPu(k))$ denotes the possibility of node k not being covered by a previous transmission and $L(u, k)$ denotes the link quality value in the link $u \rightarrow k$. After each transmission, the sender has to calculate this sender CP values and compare the CP values to a pre-defined threshold value α. If the CP of a neighbor is lower than the α value, then that neighbor node is considered as an uncovered node of sender S. Else, if the CP value exceeds α, that neighbor node is a covered node of S. S is in the sender mode and will compete with its receivers for transmission until all its neighbor nodes are covered. The above mentioned procedure is the same for any sender of the network. Now, let us assume that both N1 and N2 receive the transmission of S at the first attempt. N1 and N2 update their receiving CP values on their neighbors. Since the message is received from S, S is considered to be a covered node for both nodes. For other cases N1 and N2 will use Eq. (2.2) to calculate the CP values of their neighbors which are also neighbors of S t.

$$CPu(k) \leftarrow 1 - (1 - CPu(k)) * (1 - Pv(k|u)). \tag{2.2}$$

The term $Pv(k|u)$ represents the CPRP value of the two nodes, calculated using Eq. (2.3) where v is the sender, and u and k are receivers. The other neighbors of N1 and N2 who are two hops away from S such as N3 and N4 are considered to be uncovered nodes with a CP of 0. If any of the receivers has an uncovered neighbor, it will enter the competition for re-transmission based on its transmission effectiveness (TE). TE is based on the number of uncovered neighbors, the CP values and the quality of the respective links [7]. When nodes with uncovered neighbors are competing with each other, they calculate their TE values by using Eq. (2.4). This is the process that any receiving node in the network will follow once a data message is received. Once the TE values are calculated the back-off time is calculated by using Eq. (2.5), where C is a predefined constant in seconds.

$$Pv(k|u) = \frac{\sum_{i=1}^{i=w} Bvk(i) \& Bvu(i)}{\sum_{i=1}^{i=w} Bvu(i)}. \tag{2.3}$$

After calculating their back-off times, the nodes immediately set their timers on and time start elapsing. The node with the highest TE value will get the least back-off time and hence its timer will elapse first. When this happens that node will transmit and re-calculate the sender mode CP values as mentioned above. The other competing nodes when they hear the winners transmission will stop their timers and go back to the receiver mode and start updating the receiving CP values. This process continues until all nodes in the network are covered. This procedure enables nodes to use collective ACKs and dynamic forwarder selection, resulting in reduced number of transmissions [7].

$$TE(u) = \sum_{k \in U(u)} L(u, k) \times (1 - CPu(k)). \tag{2.4}$$

$$T_{\mathit{Backoff}} = \frac{C}{TE(u)}. \tag{2.5}$$

However, there are certain limitations associated with the proposed CPRP metric in [7]. As mentioned in [27], there are possibilities of false indication of correlation due to small bitmap length. For example, consider two bitmaps $A = [11101]$ and $B = [00001]$. If the CPRP calculation is performed at node B, according to the Eq. (2.3), B will get a conditional probability value of 100%. However, since the bitmap B has a single '1' digit, the CPRP calculation only considers the corresponding position of bitmap A. Due to this reason, the sample size of this calculation becomes one and it is difficult to predict that A will receive the same packet as B in the next transmission round. For this reason, in [27], a hamming distance based distance metric is used for correlation measurement.

2.2.3 κ Factor Metric

Another metric proposed to measure link correlation is the statistical cross correlation index based κ factor proposed in [8]. In this paper, the authors initially analyze the existing cross-conditional probability χ as an inter-link correlation metric. According to their analysis, although it is good for identifying independent links, χ fails to effectively identify the links with positive link correlation. Hence, as an alternative they consider the cross-correlation index ρ for measuring correlation. The definition of ρ is presented in Eq. 2.6

$$\rho_{t,x,y} = \begin{cases} \frac{E[x.y]-E[x].E[y]}{\sigma_x.\sigma_y}, & \sigma_x.\sigma_y \neq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\sigma_x = \sqrt{E[(x-E[x])^2]}$ is the standard deviation of x, $E[x,y]$ is the empirical mean of the x, y product and $E[x]$, $E[y]$ are the means of x and y. The term $E[x.y]$ can be termed as the probability that both x and y receive the same packet which can be denoted as $P^{(t)}_{x,y}(1,1)$. The $E[x]$ and $E[y]$ terms represent packet reception ratios of the links $t \rightarrow x$ and $t \rightarrow y$ respectively. This further gives us, $\sigma_x = \sqrt{P_x.(1-P_x)}$ and $\sigma_y = \sqrt{P_y.(1-P_y)}$.

Using the newly denoted symbols ρ can be also presented as Eq. ((2.7)). The ρ value compares the probability of both links receiving a specific packet to the probability of both receiving the same packet of the receptions are independent. If the difference between this comparison is zero that implies the two links are independent. A positive difference implies that the links are positively correlated while a negative difference means a negative correlation between links.

$$\rho_{t,x,y} = \begin{cases} \frac{P^{(t)}_{x,y}(1,1) - Px.Py}{\sqrt{P_x(1-P_x).P_y,(1-P_y)}}, & \sigma_x.\sigma_y \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.7}$$

The authors of [8] show that ρ is capable of properly identifying perfectly positive and perfectly negative correlation between links for the simplest cases. This is an advancement from earlier metric χ. However, their further analysis proves that ρ is packet reception ratio (PRR) biased and it is difficult to identify more generic cases of correlations. As a result, ρ is further normalized so that ρ^{max} is mapped to 1 and ρ^{min} is mapped to -1. The equations for maximum and minimum values of ρ is presented in Eq. (2.8) and Eq. (2.9) respectively. This normalized ρ is defined as κ and presented in Eq. (2.10).

$$\rho^{max} = \frac{min(P_x, P_y) - P_x.P_y}{\sigma_x.\sigma_y}. \tag{2.8}$$

$$\rho^{min} = \begin{cases} \frac{-(P_x.P_y)}{\sigma_x.\sigma_y}, & P_x + P_y \leq 1 \\ \frac{P_x + P_y - 1 - P_x.P_y}{\sigma_x.\sigma_y}, & otherwise. \end{cases} \tag{2.9}$$

$$\kappa_{t,x,y} = \begin{cases} \frac{\rho_{t,x,y}}{\rho^{max}_{t,x,y}}, & \text{if } \rho_{t,x,y} > 0 \\ \frac{-\rho_{t,x,y}}{\rho^{min}_{t,x,y}}, & \text{if } \rho_{t,x,y} < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

The following observations has been reported in [8] on the values of κ. if $\kappa_{t,x,y} = 0$, it implies that x and y has independent receptions for packets from t. κ can have maximum value of 1 which implies the perfect positive reception correlation. When $\kappa = 1$ and $PRR_y > PRR_x$, then if x receives a packet we can consider that y also received the same packet. Also if y loses a packet then x will also loose the same packet. κ has the minimum value of -1 corresponding to perfect negative correlation. When $\kappa = -1$ and $PRR_x + PRR_y < 1$ then x and y can be considered as never to receive the same packet. Also, if $PRR_x + PRR_y > 1$ it can be considered that x and y never losses the same packet.

2.3 Own Previous Work

During the previous modules in IKT for the Advanced project we were able to implement CF and analyze and compare its performance with other transmissions schemes. We implemented CDS based and multicast based transmission schemes for the comparison purposes. Overall results showed that

CF has a better performance in-terms of energy consumption and number of transmissions in most bitmap combinations. However, when the correlation among several links are weak CF's performance degraded considerably. From this work a paper [33] was submitted to the PIMRC'15 conference.

During the Specialization project a sliding window based enhancement was proposed for the bitmap establishment process of CF. In original CF the bitmaps results were exchanged after fixed number of hello messages. In the proposed scheme each hello message includes the result of its previous receptions from neighbor nodes and the bitmap could be updated dynamically. This allows the nodes to take dynamic node selection and correlation decisions based on more updated information. The comparison between the newly enhanced scheme and the original CF was done using simulations in MATLAB. The results suggested that the newly proposed scheme is capable of achieving higher reliability due to more accurate information for its decision making.

The previous work on analyzing and enhancing CF aided us to improve the background knowledge on sensor network transmission schemes. For this thesis, we utilize this knowledge and propose novel data transmission schemes that could provide a better performance in terms of energy consumption and reliability.

Chapter 3

The Proposed Data Transmission Schemes

The design of a data transmission schemes involves consideration of all parameters that could effect efficient and reliable data delivery. This chapter presents a detailed description about the proposed data transmission schemes. The concepts, algorithms and building blocks of each scheme is presented in detail.

3.1 Overview of The Proposed Schemes

An overview of the proposed data transmission schemes are shown in Fig. 3.1 . The basic scheme (BS) consists of link correlation based transmission (LCT) and selective re-transmission (SR) phases. The Advanced scheme has explicit re-transmission (ER) phase on top of above two phases. The BS is propose to achieve higher levels of reliability while maintaining a low energy consumption level. The AS on the other hand focus on achieving 100 % reliability of data delivery. For both schemes, the purpose is to cover all nodes in a given topology from the data originated by the source node. The proposed schemes are explained using an example topology of 6 nodes as presented in Fig. 3.2 for illustration purposes. However, these schemes are designed in a way that they can be applied to any large scale network. The components of each scheme would be presented individually in the following sections.

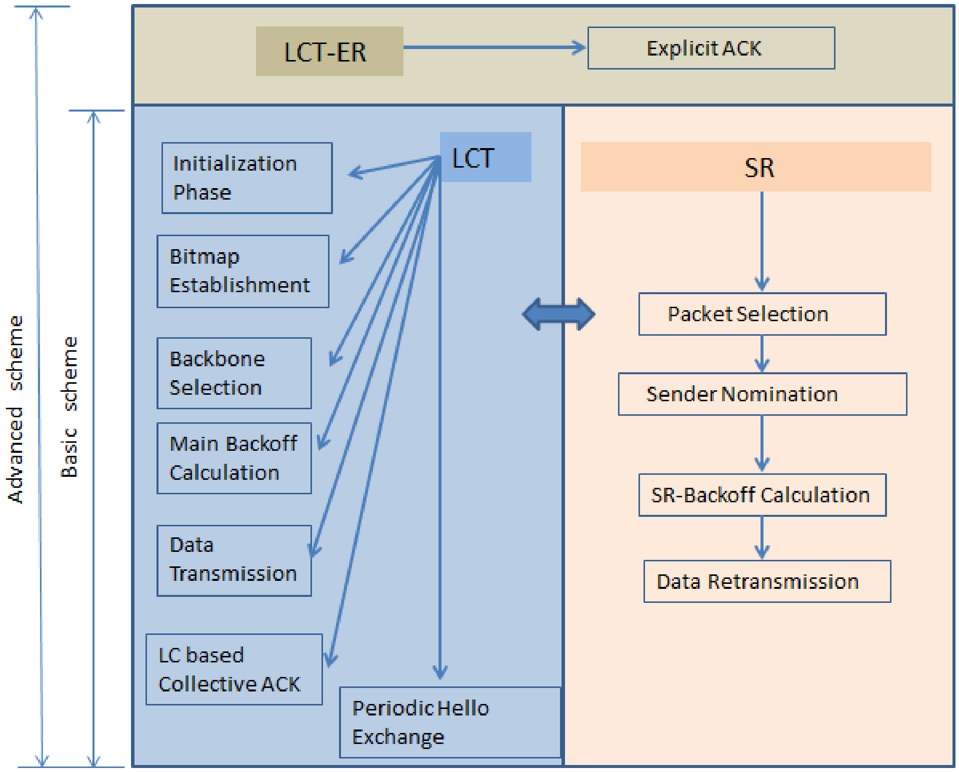


Figure 3.1: Overview of proposed transmission schemes.

3.2 Basic Scheme

The BS consist of many individual components that acts as the building blocks for the overall scheme. First, we will consider the example topology that will help us to describe these components which will follow in later subsections.

3.2.1 Example Topology

For the purpose of analyzing the proposed mechanism, we use the topology presented in Fig. 3.2 which consists of 6 nodes. Node A is considered as the main source node and the requirement is to cover all nodes in the topology for each transmission of A. Nodes C and F are the leaf nodes and will not participate in forwarding the data messages for the proposed schemes. The intermediate nodes B, E and D acts as backbone nodes and participates in forwarding the data originated by A.

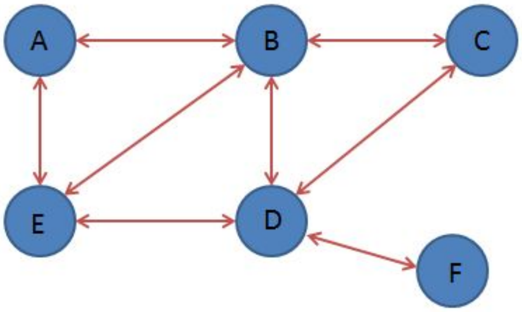


Figure 3.2: Node topology under consideration.

3.2.2 Hello Message Exchange and Bitmap Establishment

For any transmission scheme the nodes are capable of taking informed decisions only if they posses the required neighbor and link information. This information is gathered using hello messages. Hello message exchange process plays a vital role in establishing bitmaps that determines the link quality and link correlation levels of each link in the topology. A sliding window based method for utilizing hello message information is used in the proposed mechanism.

First, we assume that all nodes have pre-determined synchronized time for hello message exchange. Initially all nodes have an initialization period where they exchange hello messages and establish bitmap combinations for each neighbor link. For our analysis let's assume that the bitmap window size w is equal to 4. After this process is completed data messages are sent using the information gained from link quality and correlation values. However, with time the wireless link status will change due to various reasons. Hence, each node needs to exchange periodic hello messages to update the bitmap values that represent the changing channel conditions with time.

In the proposed sliding window based hello exchange process, each hello message that is transmitted by a node needs to carry the previous results of the hello message transmissions received from its neighbors. To make this concept more clear lets consider Fig. 3.3 which illustrates the hello message exchange process for two nodes D and F.

D and F are two neighbors that exchange Hello messages at a fixed time interval. From the H2 Hello message we can see that each node has included a (1) or (0) value representing the previous result of the H1 hello message. For example in the first Hello message exchange, H1 transmitted by D and F has been received by both nodes successfully. As a result both nodes include the respective result as (1) in their next hello message H2. However the hello message exchange of H2 has a different result. D successfully receives the H2 Hello message while F fails to receive it. When D receives H2

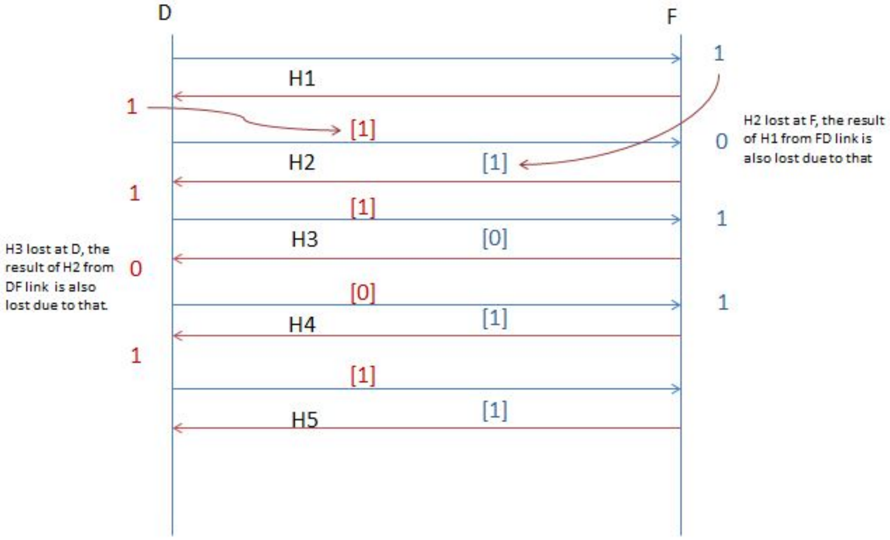
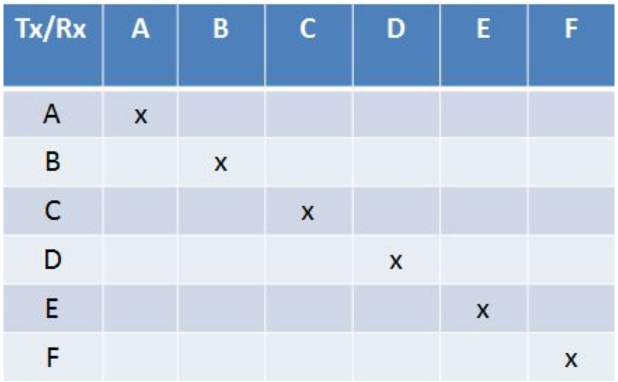


Figure 3.3: Hello exchange process between nodes D and F.

from F, D checks the contents of H2 to find the result of its previous transmission. However, since F does not receive H2 from D, F will not know the result of its previous transmission. Now the new result of H2 will be included in the next H3 Hello message. D will include a (1) while F will include a (0). Likewise the process continues and both nodes get updated about the results of its recent transmission.

In order to make this process work, each hello message should contain an additional label which contains the result of the previous transmissions of its neighbors. In addition , each Hello message sent by a node will contain the some additional information about the data transmission phase which will be discussed in future.

Table 3.1: Table to keep track of neighbor bitmaps.



Tx/Rx	A	B	C	D	E	F
A	x					
B		x				
C			x			
D				x		
E					x	
F						x

When a hello message is received each node requires to update it self using the information available in the hello message about the single hop and two hop neighbors bitmaps. So each node will keep track of its neighbor links bitmap information in a table as presented in Tbl. 3.1.

The mechanism of data transmissions using sliding window based hello messages is presented in Fig. 3.4

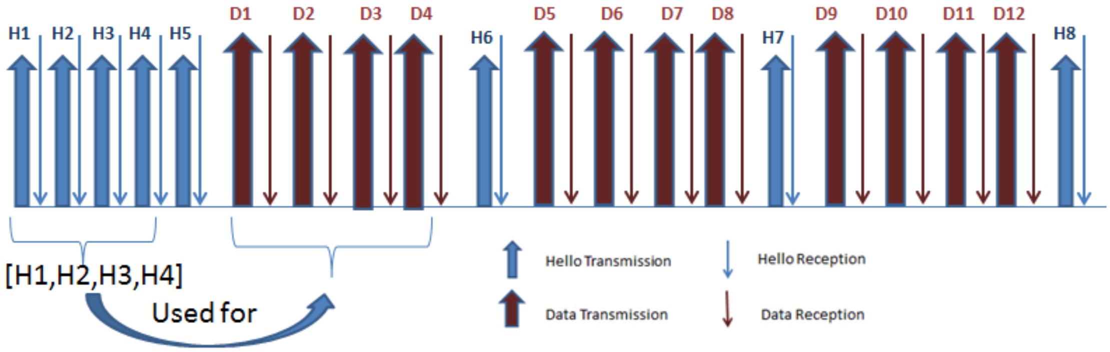


Figure 3.4: Hello messages for D1-D4 data messages.

The hello messages H1 to H4 are the messages exchanged at the initialization period. We assume that each node has received these messages and the link quality and correlation values for the first four data packets D1 to D4 are calculated using these hello messages. Next, after the four data packets all nodes engaged in periodic hello message exchange. In this H6 hello message, the sending node includes the result of the previous H5 hello messages it received from all its neighbor nodes. If this H6 message is correctly received, nodes will use the result of H5 to update bitmaps and to calculate the new link quality and correlation values for next phase of data transmissions. For example as shown in Fig. 3.5, data messages D5 to D8 will use H2, H3, H4 and H5 messages to update the above mentioned values for transmission. If the H6 message is lost then the node is not aware of the result of its previous H5 transmission and in such case D5 to D8 would be using the already existing bitmap created by hello message information using H1 to H4. This process continues with each hello exchange and the bitmap updates with each successful hello message exchange as presented in Fig. 3.6.

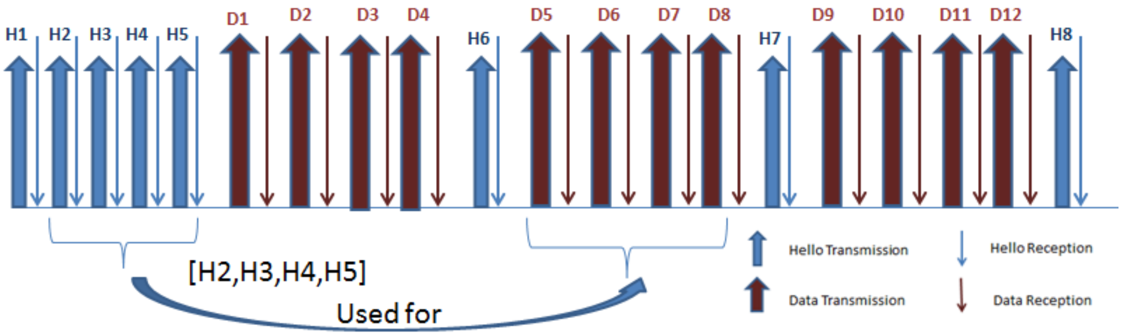


Figure 3.5: Hello messages selection for D5-D8 data messages.

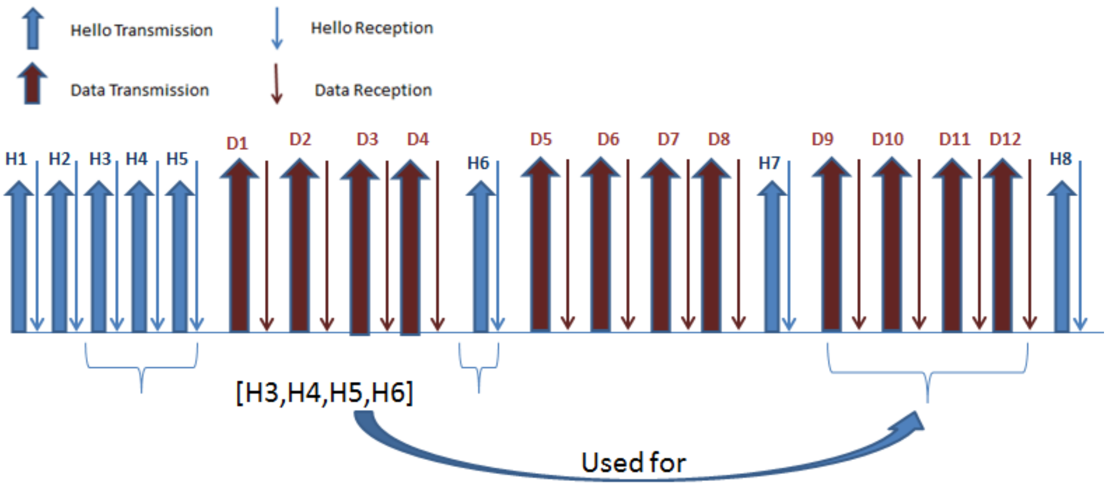


Figure 3.6: Hello messages selection for D9-D12 data messages.

3.2.3 Initialization and Backbone Selection

For the purpose of data transmissions, we propose a virtual backbone based system to avoid data transmission by all nodes. As discussed in Subsec. 2.1.2, virtual backbones help to reduce the number of transmissions and reduce energy consumption. However, the traditional CDS based schemes could cause energy holes in certain topologies due to the unbalanced distribution of load among nodes. In this scheme we extend the idea of CDS nodes by defining a set of core nodes which consist of main core nodes and supporting core nodes. Main core nodes are the nodes which must be selected in order to reach to a specific node in a topology. Supporting core nodes are a set of nodes such that atleast one of them must be selected for a certain transmission round in order to propagate the data packet forward.

Initially the sending node will send a broadcast message advertising that it is the sender and the related details regarding transmission. These details includes the inter packet interval ,transmission rate,hello message exchange interval. Then the nodes will exchange hello messages as explained in above section. After the initial Hello message exchange each node can assess the neighbor information and determine CDS set of nodes according to [11] as explained in Subsec. 2.1.2.

If we refer to our topology under consideration, we have two different possibilities of selecting a CDS set. This is understandable since CDS is not a unique set . These two possibilities are shown in Fig. 3.7 with (B, D) and (E, D) as the possible CDS sets. This scheme utilizes these different CDS sets when selecting supporting nodes.

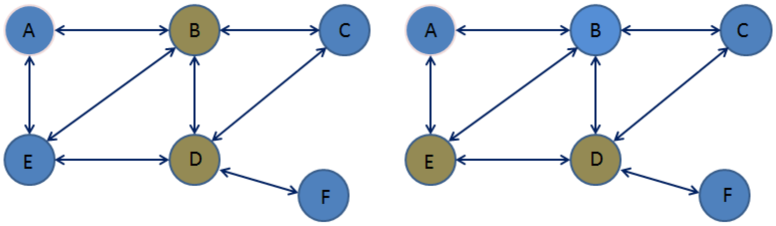


Figure 3.7: Possible CDS sets: (B, D) and (E, D) .

Since node D is common in both CDS sets we select node D as the main core node as it is necessary to cover node F. In such a situation node D will announce it self as a main core node for node F. Next, we have node B and E from the two sets remaining. In traditional CDS selection methods we select one of these two nodes as the remaining CDS node. However in our case we select both these options node B and E as the supporting core nodes. If more than two sets of CDS nodes are available the best two options are selected considering the coverage possibility and neighbor count of candidate nodes. In a data transmission phase at least one of nodes B and E will support node D to propagate the data packet forward. The resulting node types is shown in Fig. 3.8. The decision of which node to choose will depend on several factors. If both nodes receive a transmission from A both nodes will compete for transmission using the back-off algorithm proposed in Subsec. 3.2.5, and the winner will get the chance to forward the packet. If only one node receives the packet it will win the competition and will forward. Hence, the data forwarding will change between the two nodes according the these factors. And the backbone for that specific data transmission will adapt accordingly. For a larger network, after identifying the main core nodes it is possible to select 2 most suitable supporting nodes for each main core node, if there are a list of candidates. This can be done by considering the number of neighbors and nodes residual energy levels. Main core nodes also can act as supporting nodes for other core nodes depending on node topology. Supporting nodes are expected to share the load and avoid energy hole situations in the network.

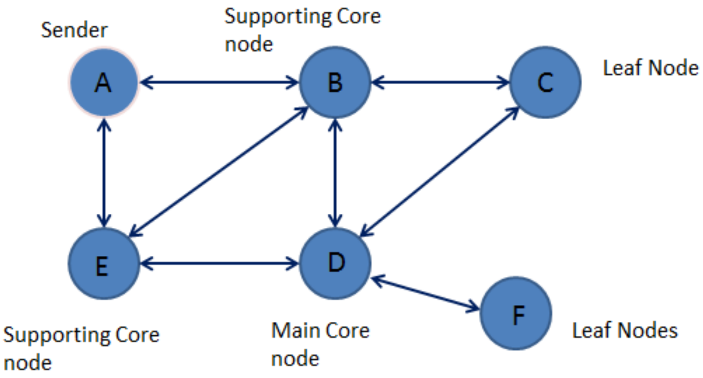


Figure 3.8: Type of nodes in the considered topology.

3.2.4 Data Transmission with Link Correlation

For the data transmission phase, it is important to determine how many times a node should transmit in order to consider its neighbor nodes as covered. For this purpose we use the concept of coverage probability [7]. CP as described in Subsec. 2.2.2, is the probability of node being covered from the current transmission or from a previous transmission. We consider each transmitting node is in the sender mode when transmitting a data packet. Similarly all receiving nodes are in the receiving mode. For sender nodes, CP is an estimation on its neighbors coverage based on the link quality levels and the existing CP values calculated from previous transmissions. Let us consider a small topology of sender s and receivers r_1 and r_2 shown in Fig. 3.9. The sender uses the Eq. (3.1) for calculating the CP values for both r_1 and r_2 and the process of sender mode CP calculation of node s is presented in Alg. 1. If a neighbors CP exceeds the predefined threshold value α then that node is added to the covered node set. The Sender will keep on competing for transmission if it has a non empty uncovered neighbor set. The value for α is considered as 0.75 for the simulations described in Ch. 5. The impact of different values of α is further analyzed in Sec. 5.5.

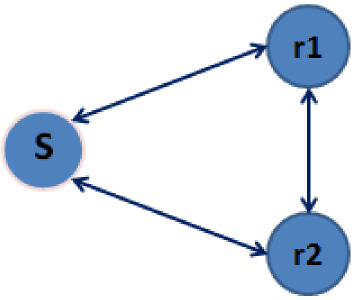


Figure 3.9: Sender s and receivers r1 and r2 .

At the receiving mode the concept of link correlation is used for determining collective ACKs. Although [7] and other previous work has used CPRP for link correlation calculations CPRP has some problems associated with is as explained in Ch. 2. Hence, for this scheme in order to measure the correlation between a sender and multiple receivers, the κ factor metric proposed in [8] is used. For a node topology of Fig. 3.9 , Eq. (3.2) can be used to calculate the κ factor. A detailed description of κ factor calculations and the notation of equations used are presented in Ch. 2. Utilizing this κ factor metric we propose the Alg. 2 for the CP calculations in receiver mode.

$$CPs(r_k) \leftarrow 1 - (1 - CPs(r_k)) \times (1 - L(u, r_k)). \tag{3.1}$$

$$\kappa_{s,r_1,r_2} = \begin{cases} \frac{\rho_{s,r_1,r_2}}{\rho^{max}_{s,r_1,r_2}}, & \text{if } \rho_{s,r_1,r_2} > 0 \\ \frac{-\rho_{s,r_1,r_2}}{\rho^{min}_{s,r_1,r_2}}, & \text{if } \rho_{s,r_1,r_2} < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

After each transmission, the sender node s will evaluate the CP of neighbor nodes as in line 3 of Alg. 1. If the condition in line 4 is satisfied that neighbor node is added to the covered node set of s. If s has atleast one node in the uncovered set it will try to compete for transmission. Otherwise, s will exit from competition and consider all neighbors as covered nodes.

When analyzing Alg. 2 for the receiver nodes, when ever a node receive a packet from a neighbor, that neighbor node is considered as a covered node. Afterwards the receiver calculates the κ factor among the neighbor links and takes decisions on their CP values. Due to the nature of how κ is defined, when $\kappa = 1$, the node with lower link quality (having less PRR value) is capable of assuming the result of the higher quality link node as shown in lines 6, 7 in Alg. 2. In this case node r1 can predict that r2 has received the same packet when it receives the packet from S and the κ is equal to 1.

Algorithm 1 Sender mode CP calculations for s

```
1:  if $CP_{sr_k} < \alpha$ then
2:      $S$ transmit
3:      $CP_{sr_k} \leftarrow 1 - (1 - CP_{sr_k})(1 - L(s, r_k))$
4:      if $(CP_{sr_k} \geq \alpha)$ then
5:          add $r_k$ to Covered set of S
6:      else
7:          add $r_k$ to Uncovered set of $s$
8:      end if
9:      if (Uncovered set $\neq \emptyset$) then
10:         Compete for transmission
11:     else
12:         Exit competition.
13:     end if
14: else
15:     Exit competition.
16: end if
```

However, if r1 is the node with higher PRR link it cannot predict the reception result of the lower PRR link node r2. Hence, it updates the CP_{r1r2} value by adding the link quality value between s and r1 $(Q_{sr2}$ to the existing CP as demonstrated in line 9. For the case of $\kappa = -1$, their are two possibilities depending on link quality levels as shown in lines 11 to 16. If the addition of two link quality values is less or equal to one, then when one node receives a packet their is high possibility of the other node not receiving the same packet. In such case, the existing CP values are maintained and no updating takes place. Otherwise, the link quality is added to the existing CP value as mentioned before. When $\kappa = 0$, the links are considered to be independent with each other. In such cases the receiving node cannot predict the reception out come of its neighbor node. And it uses the sender links quality vale to update the existing CP values as shown in line 18. Similar to the Alg. 1, the receiver node will also try to compete until all its neighbor nodes are considered as covered by satisfying the condition in line 25. In such case it will exit the competition.

3.2.5 Back-off Time Calculation Procedure

It is important to define a procedure for channel access when several nodes in a topology requires concurrent shared channel access. When a node has atleast one uncovered neighbor that node tries to access the medium for data transmission. However, when more than one node is in this transmission mode each node needs to have different priorities in accessing the channel in order to avoid collisions at the receivers. The proposed scheme defines a contention based procedure for competing nodes to

Algorithm 2 Receiver mode CP Calculations for r_1

```
if Sender=$s$ then
    $CP_{r_1S} = 1$
    add $S$ to Covered set of $r_1$
    Calculate K factor between $sr_1$ and $sr_2$ links
    if $k = 1$ then
        if $Q_{sr_1} \leq Q_{sr_2}$ then
            $CP_{r_1r_2} = 1$
        else
            $CP_{r_1r_2} = CP_{r_1r_2} + Q_{sr_2}$
        end if
    else if $k = -1$ then
        if $(Q_{sr_1} + Q_{sr_2}) \leq 1$ then
            $CP_{r_1r_2}$
        else
            $CP_{r_1r_2} = CP_{r_1r_2} + Q_{sr_2}$
        end if
    else if $k = 0$ then
        $CP_{r_1r_2} = CP_{r_1r_2} + Q_{sr_2}$
    end if
else if Sender=$r_2$ then
    $CP_{r_1r_2} = 1$
    $CP_{r_1s} = 1$
    add $s$ and $r_2$ to Covered set of $r_1$
end if
if $CP_{r_1r_2} \geq \alpha$ then
    add $r_2$ to Covered set of $r_1$
else
    add $r_2$ to Uncovered set of $r_1$
end if
if Uncovered set of $r_1 \neq \emptyset$ then
    Compete for transmission
else
    Exit competition.
end if
```

access the shared wireless channel.

In the proposed scheme each node calculates its own back-off time using the Eq. (3.3).

$$T_{\textit{Back-off}} = \frac{C}{L * N(u)} \tag{3.3}$$

Here $T_{\textit{Back-off}}$ is the back-off time calculated for node n , C is a predefined time constant,L represents the hop count from the source node and $N(u)$ is the number of uncovered nodes of the competing node.

Each node waits the amount of time calculated as the back-off time before transmitting or forwarding data packets. The node with the minimum back-off time will transmit first since its timer elapses first. This node whose timer elapses first is known as the winner of the competition among nodes in need to access the wireless channel. When the winner transmits other neighboring nodes can overhear this transmission. When this happens, each node who overheard this transmission will stop its timer and recalculate the coverage probabilities of its neighbors to update their $N(u)$ set. If all the neighbor nodes are covered, such a node will exit from the competition while a node with uncovered nodes will recalculate its back-off time value according to Eq. (3.3), and reenter the competition. Although this process of back-off calculation has a similar approach to other contention based schemes as in [7], this scheme introduces the novel concept of utilizing the hop count from the original sender to minimize the number of intermediate re-transmissions.

Let us analyze the purpose of using the hop count distance metric L to calculate the back-off time. If we reconsider the given topology in Fig. 3.2, the purpose of the transmission is to disseminate all the packets transmitted by sender A to all the nodes in the topology. So when the transmitting node is further away from original sender, it has a better chance of covering new set of nodes and at the same time these transmissions act as an implicit ACK to the competing nodes with a lesser hop count. For example lets consider a competition between nodes B, E and D for transmission of a data packet with same sequence number. For nodes B and E to be in the competition, some of its neighbor nodes must be still considered as uncovered. For node D to be in competition it should receive the same packet of a previous transmission from either B or E. At the same time nodes B and E are 1 hop away from the original sender A and node D is two hops away from it.

If we disregard the hop count of each competing node and consider only the other parameters like number of uncovered nodes, then there is a high probability of node B or E wining the competition and re-transmitting. However, if we consider the hop count metric as in Fig. 3.10, we can assign nodes B and E with $L = 1$ and node D with $L = 2$ giving node D a higher probability to win the competition. This is according to Eq. (3.3), if all other parameters remain constant. In other words, node D will get a higher priority with regard to the hop count distance from the source node A. This will provide node D with a higher probability of winning the competition and to transmit before B or E re-transmit the same packet. Now this transmission from D will help the packet to propagate forward in the topology quickly and reach new nodes as in node F. At the same time this transmission if received will act as an implicit ACK to nodes B and E. This implicit ACK will help nodes B and E to update its CP values and consider D as a covered node. Furthermore, according to Subsec. 3.2.4, this ACK will indirectly help to update the other neighbors CP values as well. For example D's transmission will help B to update the CP values on node C. The higher the number of transmissions by D, the better chance of B and E exiting the competition with out re-transmitting. Hence, with the help of L node B and

E's chances of unnecessarily re-transmitting the packet when D has already received it will decrease. Hence, according to the proposed back-off scheme, if a core node is further away from the source node, it has better chance of winning the competition and moving the data packet forward while reducing intermediate re-transmissions from nodes closer to the source node. Since L is measured from the source node it uses only the $N(u)$ metric for back-off calculations.

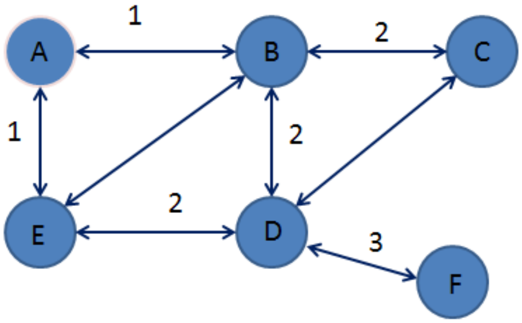


Figure 3.10: Minimum hop count from A to each node .

In case when nodes with the same hop count distance competes for transmission, the uncovered neighbor count metric $N(u)$ will help to prioritize between them. For an example, nodes B and E are two nodes with same hop count distance from the source node A. When they are competing the number of uncovered nodes $N(u)$ will play a major role in deciding the node with minimum back-off time value. The idea of using this metric is intuitive. A node with higher number of uncovered nodes has a better chance of covering more neighbor nodes from its transmission. Hence, in a general scenario node with a higher hop count from source and with a higher number of uncovered nodes will get the priority over the other nodes.

The value C is a time constant in milliseconds (ms). This parameter should be defined considering the transmission duration and propagation delay so that the nodes will have the time to receive a transmitted packet before their timer expires. Also the size of the network and the number of hops to the furthest node in the network need to be considered when deciding on a C value to avoid too large or too small back-off values.

3.3 Selective Re-transmission Procedure

After a data transmission phase of size w nodes go to selective re-transmission (SR) phase before returning to the next data transmission phase. Here w is the number of data packets transmitted by the source before each node enters the SR phase. For example lets consider $w = 4$. After, the first 4 data packets are transmitted and forwarded through the network , all nodes exchange the periodic hello messages with each other. The timing of this exchange is announced by the source node at initialization phase where source node is announced. Since the source node is aware of each packet transmitted, it could either consider this period in terms of the number of packets transmitted or a pre-defined fixed interval. However some nodes may not receive all the w packets at the first attempt. Hence if the number of packets are used as an indicator to hello message exchange, these nodes will have trouble in deciding the timing of exchange. To avoid this confusion we assume that all nodes exchange hello messages after a fixed time period Tw. During this period the source node has transmitted w number of packets. Fig. 3.11 presents an example of SR that occurs periodically when $w = 4$.

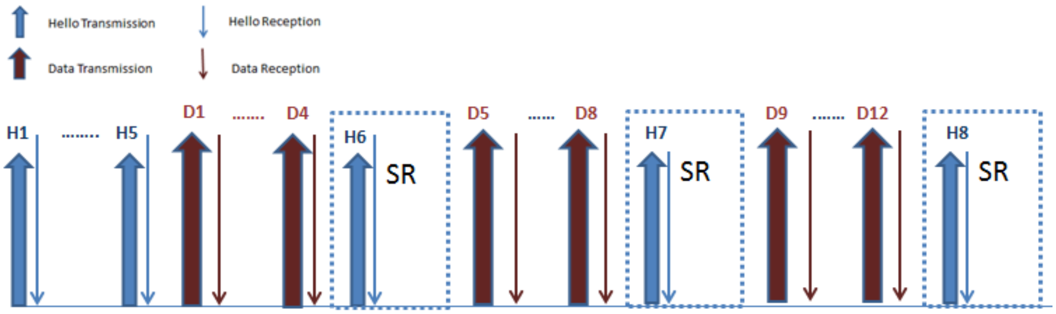


Figure 3.11: Periodic SR procedure .

For the SR phase to be effective each core node needs to keep a copy of the received packets for a temporary time period. This copy will be used to re-transmit the packet if a neighbor is requesting a re-transmission. The advantage of this process is that instead of re-transmitting the packet from the source node, the core nodes closer to the destination node can transmit the missing packets from their recorded copy of the message. This will reduce higher number of re-transmissions from other nodes and make the re-transmission procedure more efficient. Once core nodes make sure that the neighbor nodes have received a specific sequence number they can delete the copy of that packet from memory.

3.3.1 Packet Selection Criteria

Each node has the array of sequence numbers of the packets it received during the data transmission phase. Each node compares the received set of sequence numbers with the expected set of sequence numbers and find the missing set of sequence numbers.

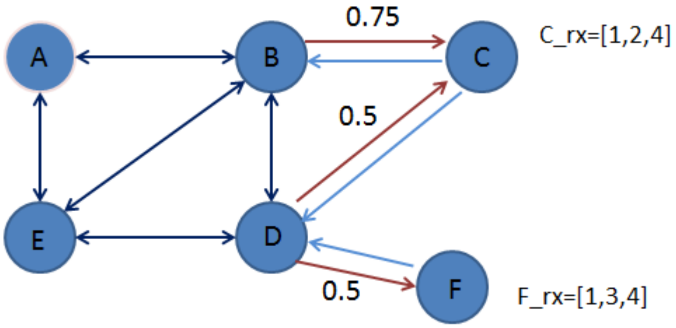


Figure 3.12: Missing packet selection.

Let us consider the Fig. 3.12 where the topology is shown with leaf nodes missing certain packets. Lets assume that the source node A has transmitted four data packets and node F has received packets with sequence numbers 1, 3, 4 . Likewise, node C is assumed to be received sequence numbers 1, 2, 4. Assume that all other nodes in the topology have successfully received all packets. When the SR phase begins, all nodes assess the missing packet numbers as described above. Here node C will compare its C_{rx} array with expected array of [1, 2, 3, 4] and will obtain sequence number 2 as a missing packet. Similarly node F will find that packet with sequence number 3 is missing. When this missing packets are calculated each node includes this missing packet array in the next hello message they send. And the neighbor nodes will be aware of the missing packets of respective neighbors.

3.3.2 Nomination of Senders

When re-transmitting the missing packets, several nodes could receive the missing packets request from a certain node. In this case there could be competition among these nodes to send the same packet. For example in Fig. 3.13, when node C announces that it is missing the sequence number 3, both nodes D and B could receive this hello message and try to transmit this packet to C. However, if we consider the link quality levels of BC and DC links, BC link has 0.75 link quality level which is greater than the 0.5 link quality level of DC. Hence, the best candidate to forward the missing packet to C in such a situation is node B.

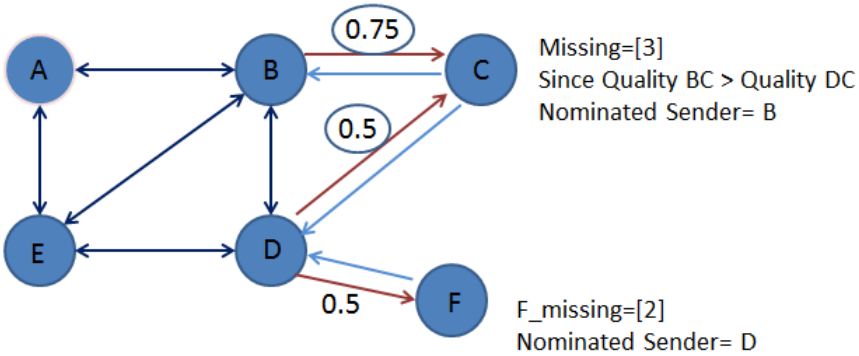


Figure 3.13: Sender nominating example.

In order to make sure that the best quality links are utilized for re-transmission of missing packets, each node nominates one of its neighbor nodes as the nominated sender in SR phase. This nomination is based on the link quality levels from the neighbor nodes to that specific node. In the case of C, since BC has the better link quality level node C nominates node B as the sender for its missing packets. In the case of node F since it has only D as the neighbor its default sender choice is node D.

If a node has multiple choices as in the case of C and if both has equal link quality levels,the node with lower hop count distance from source node is selected as the nominated sender. Since the node with lower hop count is closer to the source node it has a higher chance of already receiving the missing packets than a further away node. This scenario is shown in Fig. 3.14. In addition to the missing packets array, each node includes information regarding the nominated sender in the next hello message.

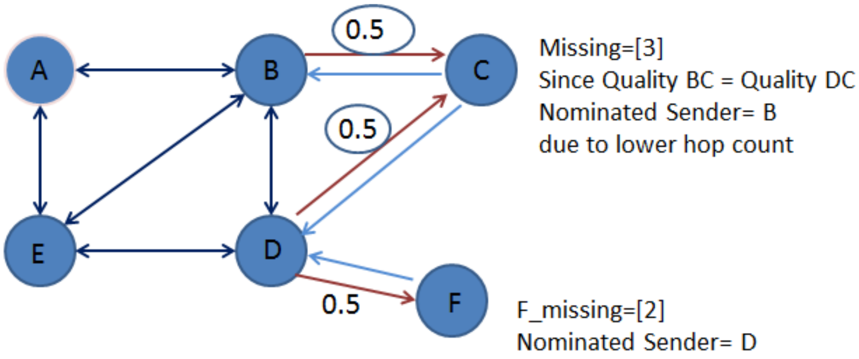


Figure 3.14: Sender nomination for equal quality neighbor links .

3.3.3 Back-off Calculation Method for SR

When several nodes want to re-transmit missing packets to their neighbors we need to define a method of accessing the channel for each node to avoid collisions. However the re-transmission scenario is quite different from the original case we discussed earlier in Subsec. 3.2.5. At the SR phase nodes can have several packets to transmit towards a specific neighbor who nominated that node. In this case, each node has to transmit the missing packets if the packet is already in their possession. So, there is no special need of giving any node a priority since the transmission of that node does not have any impact on the other nodes transmissions. Hence, in this scenario we only want to avoid collisions between the senders. To achieve this each node selects a random time between [5,15] ms interval and count down the timer. When the timer is elapsed that node transmit first and the other nodes reset their timers and select new random values for next round of competition and start the countdown again. Here the node who selected the least random number will transmit first and that node will transmit all the packets that is required from it one after the other. The number of times a node transmits depends on the CP value as before and when the CP for one sequence number is satisfied node will move to transmit the next packet. When one node finishes with its transmission the other node with next least back-off time value will win the competition and start transmitting. The process continues until all nodes have finished their expected re-transmissions.

To make the process of SR simpler, the nodes do not calculate the κ value and collective acknowledgments in this phase. Since this is a recovery mode, the nodes try to cover each of its neighbors with missing packets without relaying on collective ACKs. The Fig. 3.15 shows an example to further illustrate the described procedures.

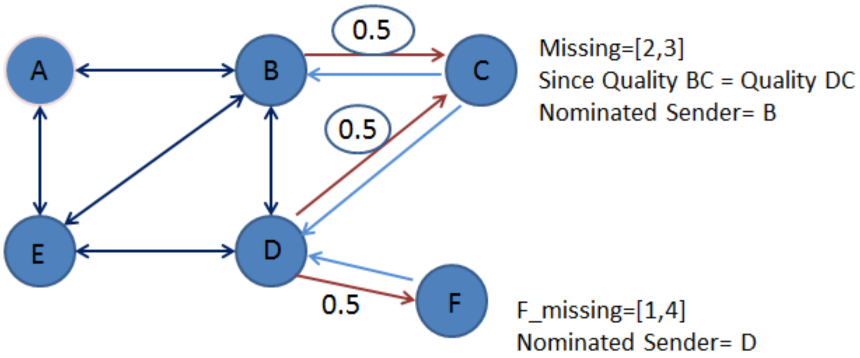


Figure 3.15: Back-off exmaple for SR .

According to the figure node C is missing packets with sequence numbers 2 and 3. Due to the equal quality levels between BC and DC links the nominated sender is B which has the lower hop count from A. For node F, packets with sequence number 1 and 4 is missing and its default sender is D. If we assume both B and D to posses all the required packets, then both nodes will choose a random number between [5,15] and start the count down. Assume that B has the least random value and B wins the competition. Then B will first transmit packets with seq number 2. Since the link quality is 0.5 B needs to transmit twice to get the required CP values to reach the threshold of α. After the two transmissions of packet number 2, B will again transmit packet number 3 twice and exit from transmission. At the same time if D hears this transmission it will reset the timer and select a new random number to wait. With each reception of B's transmissions D will renew its back-off time and when B finishes its transmissions D will get its chance and will eventually transmit packet number 1 twice followed by packet number 4 twice to reach the desired CP values. After this both nodes consider their neighbors to be covered from the missing packets and they exit the SR phase.

3.4 Advanced scheme

The use of SR phase involved in the BS helps to improve the reliability of overall packet delivery. However a 100% delivery rate cannot be guaranteed using the above mechanisms only. Many reasons could contribute to this. The hello messages carrying the missing packets may not received at the nominated sender. Also even after several re-transmissions on SR phase, the packet may still not receive the intended receiver. Sometimes certain applications may require 100% reliability of data transmission. To achieve this reliability requirement the AS scheme us proposed. The AS has all components described above under the BS scheme. AS runs on top of BS and its additional components are the ER phase and data re-transmission process until all nodes are covered with all packets.

3.4.1 Explicit Re-transmission Phase

In the ER phase each node will send an explicit acknowledgement to its neighbor nodes asking for the missing packets. This is different from SR phase in the following way. SR was used in periodic fashion and the hello messages were used to send information regarding missing packets. However in ER, instead of hello messages, explicit ACKS are used for communication. This ACK does not require all the fields that hello message contains and it will have a smaller packet size compared to hello message size. For the purpose of demonstration lets assume that we implement ER at the end of the total data transmission phase as shown in Fig. 3.16. So when the original data transmission and SR phase in between is finished, there could still be missing packets that needed to be transmitted. When ER scheme is activated nodes who are missing packets will send an explicit ACK which contains the missing packet sequence numbers to its neighbor nodes. After sending the ACK nodes will wait for a certain amount of pre-defined time period to see if they receive the packets from its neighbors. If the ACK is received by the intended node, and if the missing packets are available at that node, then the data re-transmission phase begins. Once the re-transmission is finished, each node will re-asses its missing packet status and send and updated request using another explicit ACK. This process continues until all the packets are received at each node.

The missing packet calculation, sender nomination , back-off calculations and data re-transmission phase on this process is similar to the process described under SR phase.

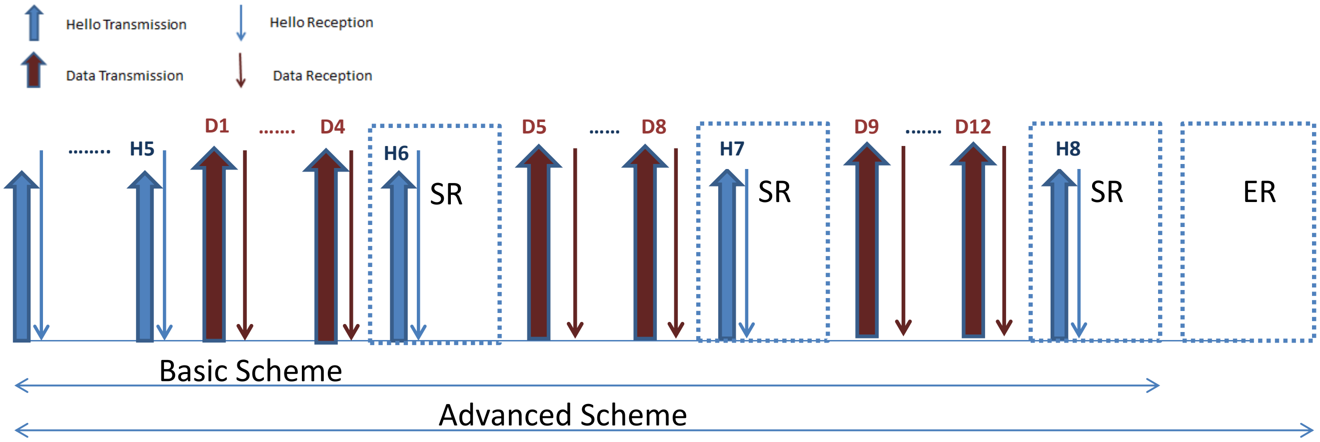


Figure 3.16: Implementing ER Scheme at end of data transmission .

For large number of packets activating ER at the end could may cause a larger delay in receiving missing packets. Due to this the intermediate nodes will have to keep a copy of missing packets for a long period of time. For such situations ER could be employed in paralleled with SR. So depending on the reliability requirement ER can be utilized adaptability with the main scheme.

Chapter 4

Implementation and Performance Parameters

The transmission schemes proposed in Ch. 3 are implemented in MATLAB for simulation and performance evaluation purposes. This chapter presents a detailed description about the 4 different transmission schemes that was implemented in MATLAB. In addition, the performance parameters under consideration are also explained.

For the performance evaluation the proposed BS and AS were implemented using MATLAB software. For the purpose of comparison, CF and Non-link correlation based transmission schemes (NLCT) were also implemented. The simulations were run assuming a data transmission of 16 packets with 9 hello messages as explained in Subsec. 3.2.2. The periodic hello messages were defined to exchange after 4 data packet transmissions from the sender node A. For the simulation purposes, we assume that the interval between two data packet transmissions as 100 ms. In the following sections we consider the implementation details of each scheme explained with use of unified modeling language (UML) based flow chart diagrams.

4.1 Basic Scheme

As described in Ch. 3, the BS consists of LCT and SR phases. Fig. 4.1 presents the general high level implementation flow chart for the BS. The flow charts are presented having a general topology and implementation in mind without limiting to the proposed topology. Initially, sender announcement and initial hello exchange processes occur. The information obtained from initial hello message exchange is used for bitmap establishment and the backbone establishment phase. After the bitmap and backbone is established, data transmission phase begins. In the flow chart, data transmission is shown in a separate process which will be further illustrated in Fig. 4.2. The SR phase occurs after the data transmission phase and it is also represented in a separate process in Fig. 4.3. The new bitmap is established using the hello messages used in SR phase and this procedure continues until all data packets are transmitted.

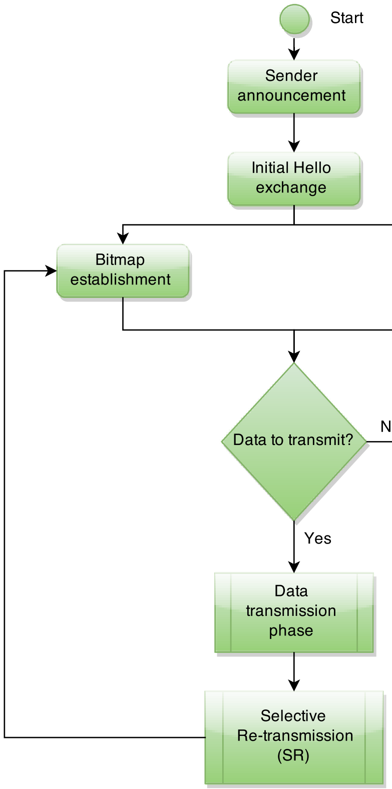


Figure 4.1: High level flow chart of BS implementation .

In order to implement the hello message exchange, a sequence of bits were generated representing various overall link quality levels. According to Fig. 3.2, there are 16 links in the system that we need to analyze. If we consider a bitmap size of 4 bits, there are 2^4 possible bit combinations as the bitmap values for a single link. When we consider all 16 links, it means that there are 2^{64} different bitmap combinations. However, for this simulation we have omitted bitmaps with all zero bits. If such a bitmap occurs then those two nodes may no longer be considered as neighbors and hence, this situation is avoided. Even without all zero combinations, there are very large number of possible bitmap combinations and it is not feasible to analyze them all in this study. Hence, the selected combinations of bitmaps for analysis were generated considering the overall link quality levels of the topology which will be further explained in Ch. 5. The bitmap establishment process was done according to the sliding window based mechanism described in Ch. 3.

Backbone establishment has to be done considering the topology in to account. This follows the process explained in Ch. 3, under backbone establishment. For the considered topology when the above process is followed, nodes B, E and D can be selected as the backbone nodes that participate in forwarding the transmissions from A.

4.1.1 Data Transmission Phase

The data transmission phase of BS is shown in Fig. 4.1. After the packet transmission sender node calculates the sender CP values while the receivers calculate the receiver CP. This receiver CP calculation includes the κ factor calculation as explained in the Ch. 3. After the CP calculations both sender and receivers check if they got any uncovered neighbors. If any uncovered neighbors exist, that node will calculate the back-off time using the hop count and uncovered node number based back-off algorithm. The node with minimum back-off value is then designated as the winner and that node is considered as the next packet forwarder. For other nodes in competition they will go back to receiver mode and update their CP values according to the new transmission of the winner. This process continues until all nodes have no more uncovered neighbors.

For each node that is listening for a packet reception there are 3 reception status possibilities for an incoming packet. The packet could be properly received, received with errors or could be completely lost in the wireless channel. Receiving nodes will only calculate the CP values if the packet is properly received. In order to simulate the packet reception status, a parameter called channel link quality is introduced. The channel link quality is calculated using the bitmap sequence initially generated. Since certain hello messages could be not received properly,channel link quality could sometimes be different with link quality values calculated and perceived by individual nodes. After each transmission the channel link quality of the link between transmitter and receiver is compared with a randomly generated number between $[0, 1]$. If

$$\text{Channel link quality} \geq \text{Random number [0,1]},$$

then that transmission is considered to be successfully received at the receiver. This consideration ensures that the links with higher link quality has a better chance of successfully transmitting a data packet than a link with lower link quality which is similar behavior to a real-life scenarios.

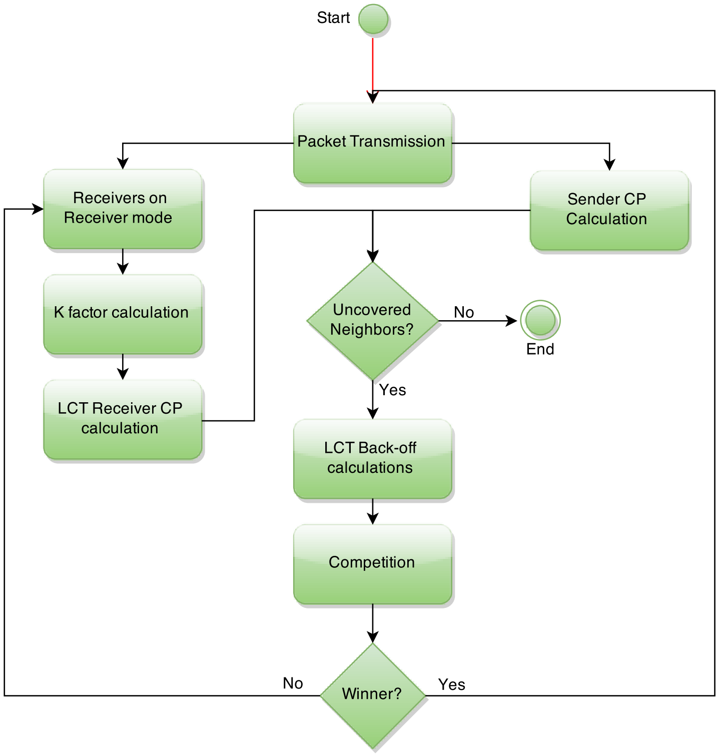


Figure 4.2: High level flow chart of BS data transmission phase implementation .

4.1.2 SR Phase

The flowchart of SR phase of the BS scheme is presented in Fig. 4.3. This SR phase occurs after a certain fixed number of data transmissions as explained in Ch. 3. In the SR implementation, each node will assess their received packets. If any packets are missing they will nominate a neighbor node as the sender depending on the link quality levels of the incoming links. The periodic hello message exchange is presented inside the SR phase, and the hello message carries the information regarding the missing packets and the nominated sender. When a node receive a hello message that nominates him as a sender, it checks whether the missing packets are available for transmission. If available, node will got to competition and the winner will transmit all the missing packets until the CP values are reached. Once the current sender node is completed with its transmission, other nodes can win the competition and transmit.

For this implementation SR phase was used after every four data transmissions. The back-off times were generated using randomly using MATLAB functions and the node with minimum value was considered as the winner.

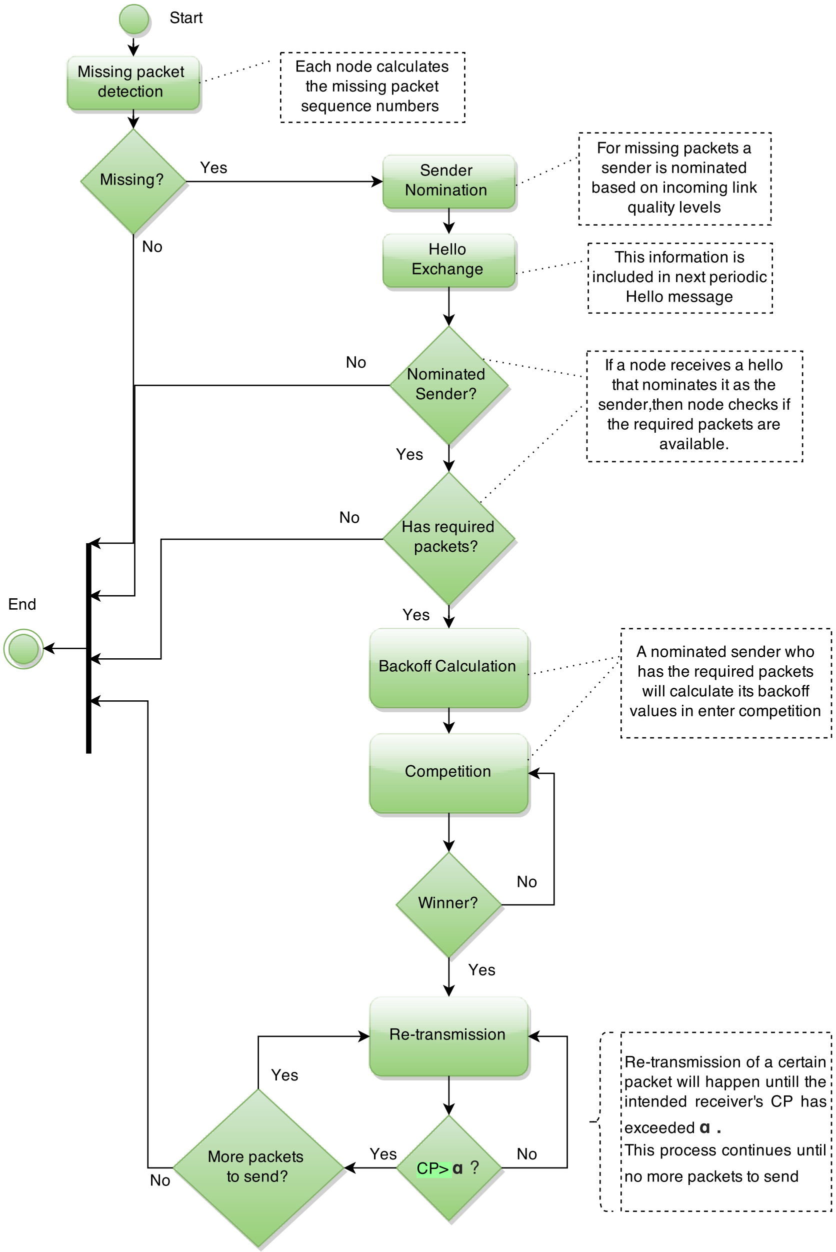


Figure 4.3: General implementation flowchart of SR phase in BS scheme.

4.2 Advanced Scheme

The AS extends the BS for achieving 100 % reliability as shown in Fig. 4.4. ER phase could be enabled depending on the reliability requirement. For large number of packets with higher reliability it could be enabled in parallel with SR phase. For smaller data transmissions ER could be enabled after all data has been transmitted. For the simulation purposes we have used the ER phase at the end of all data transmissions.

The flow chart of ER phase of AS is presented in Fig. 4.5. It slightly differs from SR phase in following ways. Instead of periodic hello messages used in SR phase the ER phase use explicit ACKs. Hence, after each data re-transmission, nodes will reassess its missing packets and send an explicit ACK to neighboring nodes. Unlike SR phase, this process continues until all nodes have received all packets.

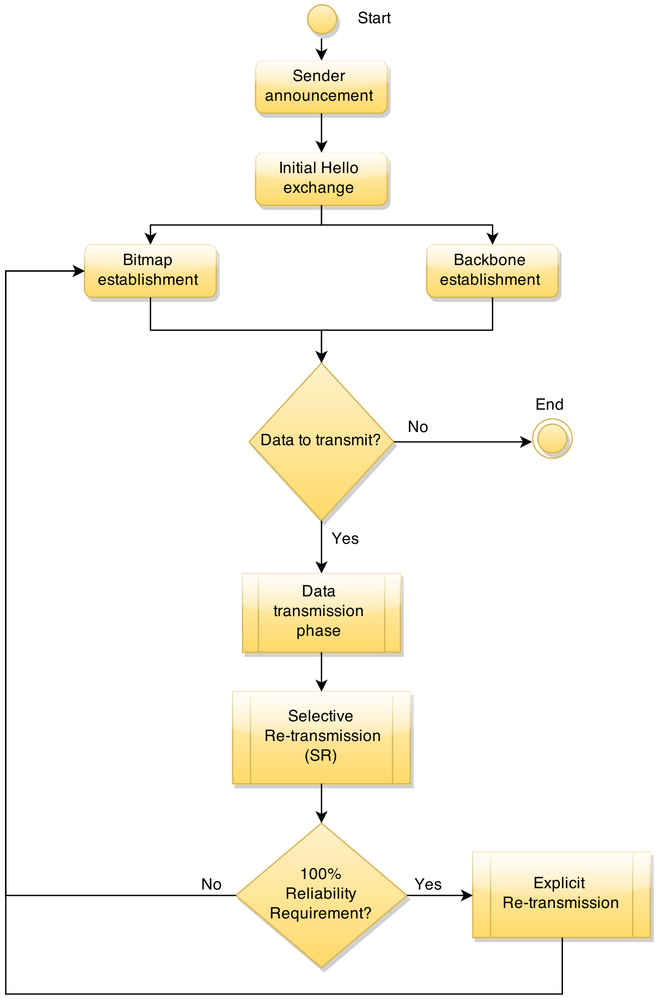


Figure 4.4: High level flow chart of AS implementation.

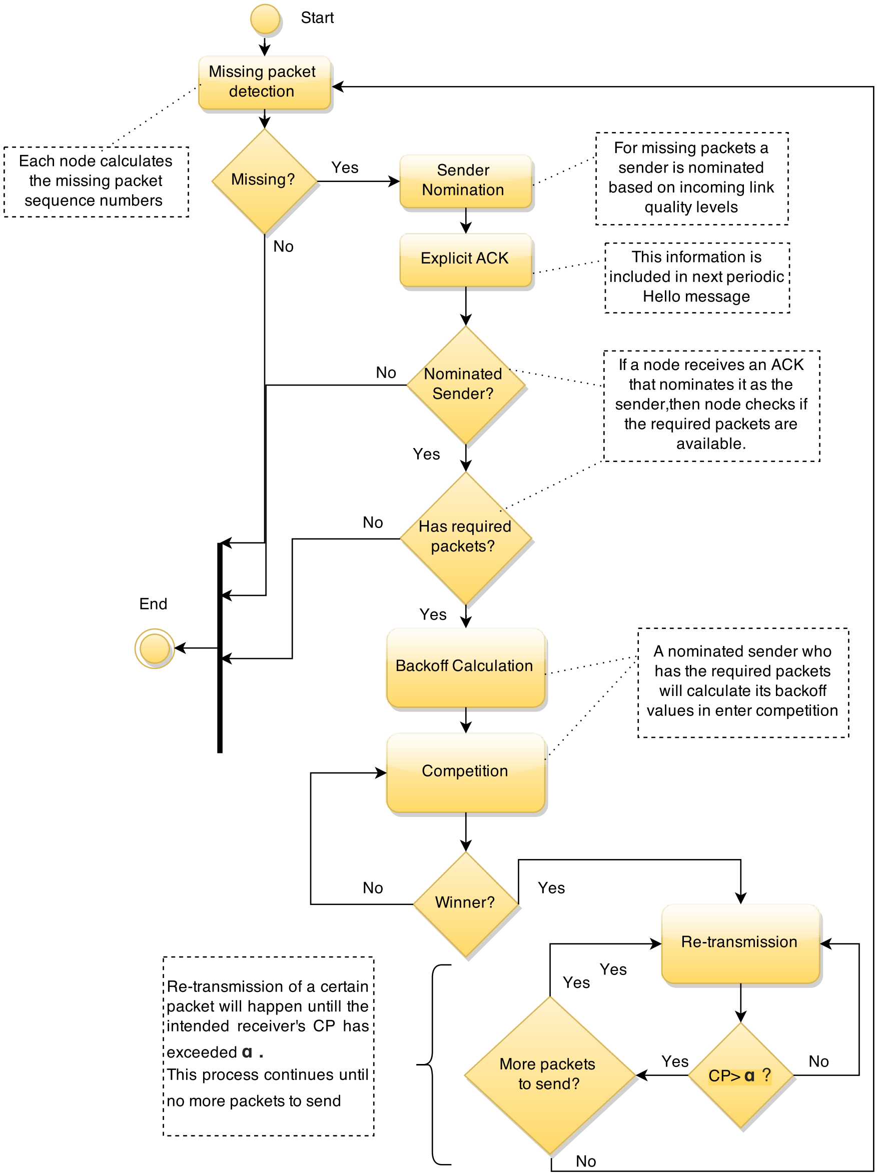


Figure 4.5: High level flow chart of ER implementation.

4.3 Collective Flooding

The CF protocol is implemented according to [7] as explained in Subsec. 2.2.2. In order to have a fair comparison the modified version of CF with sliding window enhancement was used for this implementation. In general, CF has a similar implementation flow as explained in BS . However, it differs mainly from BS via differences in data transmission phase and unavailability of a SR phase.

In addition, CF does not have a backbone establishment phase. CF uses the concept of dynamic forwarders. Hence, the data forwarding nodes are not limited to core nodes as in AS.

The data transmission phase in CF is presented in Fig. 4.7. For CF implementation, CPRP based collective ACKs are used to utilize link correlation and to calculate the receiving CP values. In addition, for channel access contention, a TE based back-off calculation method as described in Subsec. 2.2.2 is implemented.

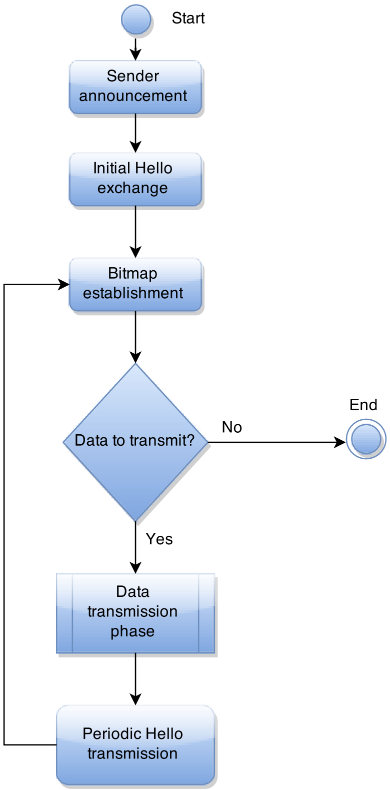


Figure 4.6: High level flow chart of CF implementation.

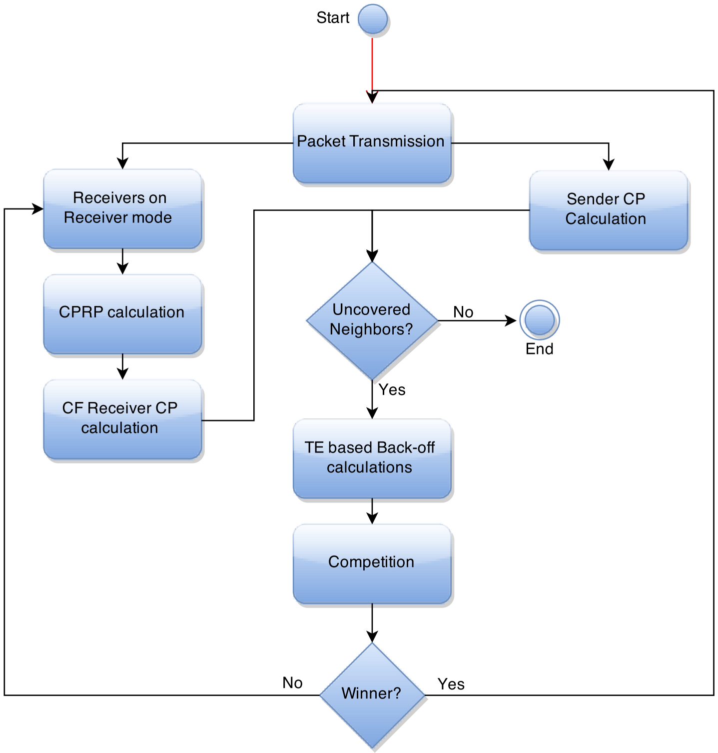


Figure 4.7: High level flow chart of CF data transmission implementation.

4.4 Non-Correlation based Transmission

For NLCT scheme, the overall implementation flow diagram is the same as BS presented in Fig. 4.1. However, the data transmission phase differs from that of AS as shown in Fig. 4.8. Since NLCT deals with the independence assumption between a sender and multiple receivers, the receiver mode CP calculations are completely different from BS. The NLCT receiving nodes can not use the collective ACKs due zero no correlation between links. Hence, the the nodes could only infer that the sender of the transmission is covered from a given transmission. To cover the neighboring nodes, NLCT needs to depend on its sender CP calculations based on link quality.

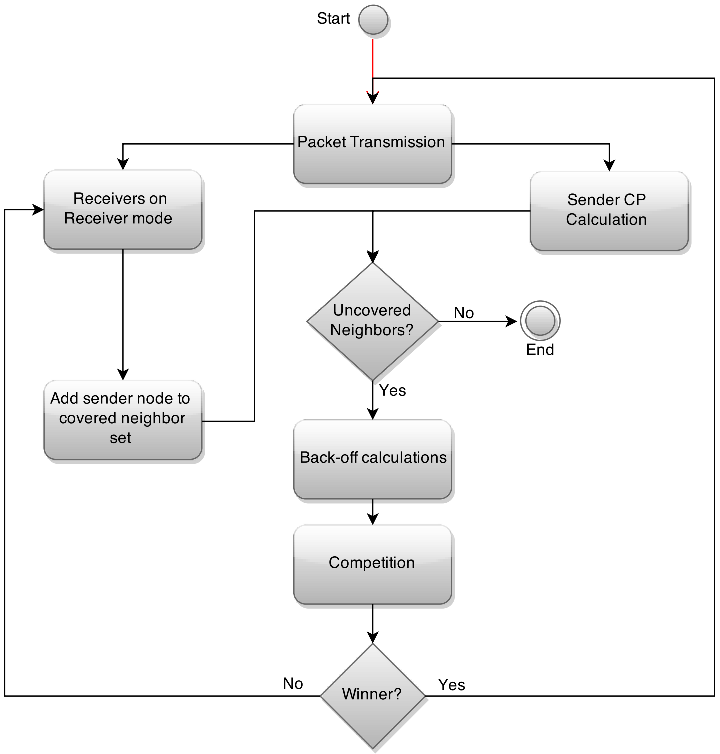


Figure 4.8: High level flow chart of NLTC data transmission implementation .

4.5 Performance Metrics

In order to evaluate and performance of the proposed transmission schemes a set of performance metrics are used.

4.5.1 Total Number of Transmissions

Total number of transmissions is the sum of all transmissions done by each node in the topology for the considered time period. For each packet transmission, the participating nodes and the number of times each node transmits could vary depending on link quality levels, correlation and reception status of nodes. Hence, for each transmission the transmission counter of each node is incremented and at the end the total number of individual transmissions of each node and the system as a whole can be calculated.

4.5.2 Total Energy Consumption

Total Energy consumption is the summation of energy consumed by each node in the topology for the considered time period. A given node can consume energy in following methods.

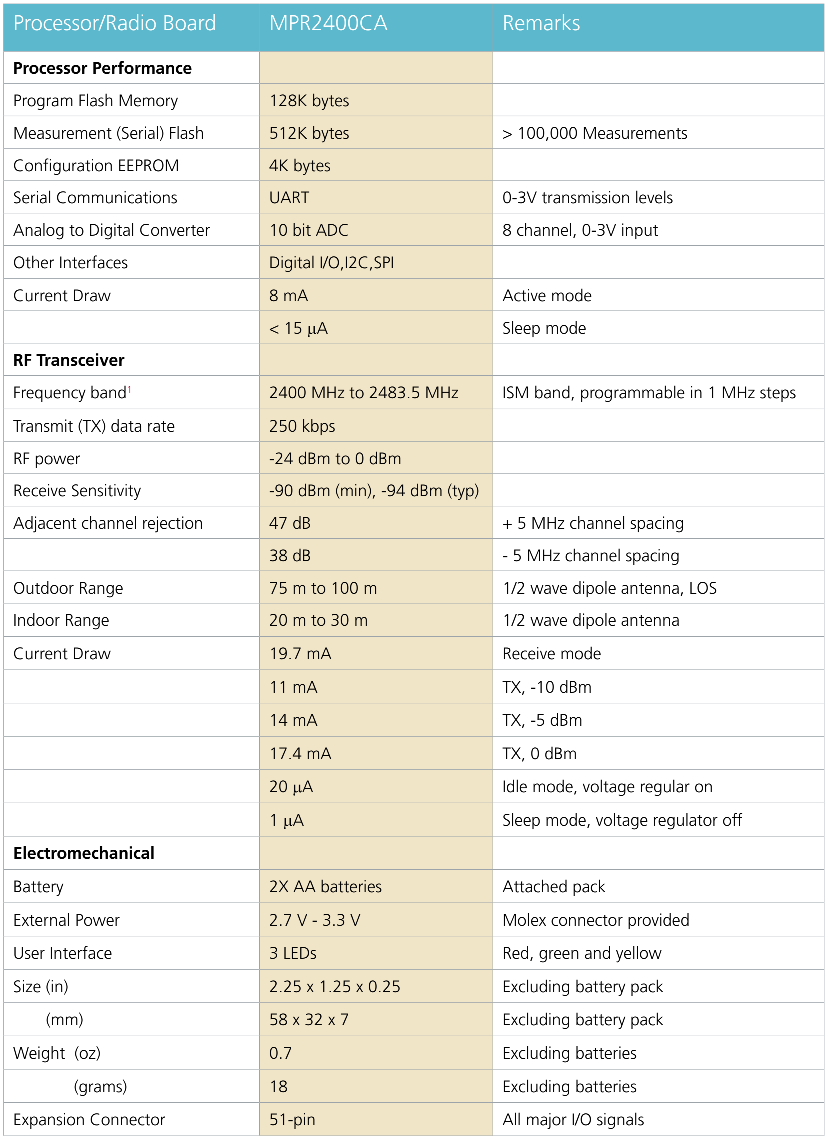
- Transmission energy
- Receiving Energy
- Idle listening Energy
- Sleep mode energy

As illustrated below, the total energy E_{tot}, is the sum of the energy consumed for transmission and reception of data, hello and ACK messages as well as during the sleep periods.

$$E_{tot} = \sum_{j=1}^{N_t} P_{tx}T_{tx}(j) + \sum_{j=1}^{N_r} P_{rx}T_{rx}(j) + \sum_{j=1}^{N_c} P_sT_s(j) + \sum_{j=1}^{N_c} P_iT_i(j), \tag{4.1}$$

where N_t, N_r, and N_c are the number of transmitted, received data packets, and transmission cycles, T_{tx}, T_{rx}, T_s,T_i P_{tx}, P_{rx}, P_s and P_i are transmission, reception, sleep time and idle time power for each packet respectively. Note that typically $N_t \neq N_r$ due to possible re-transmissions. The corresponding power consumption values for transmission, reception, sleeping mode and idle mode are based on the data sheet of MICAz sensor nodes. [32] as presented in Fig. 4.9.

The energy consumption for sleeping mode and idle mode duration can be easily added based on idle and sleep time calculations. However, since all the schemes under comparison have similar idle and sleeping time periods, the impact of considering these parameters for the comparison will be minimal. Hence, the more dominant energy consumption methods of transmission and receiving of data ,Hello and ACK messages are considered here.



Processor/Radio Board	MPR2400CA	Remarks
Processor Performance		
Program Flash Memory	128K bytes	
Measurement (Serial) Flash	512K bytes	> 100,000 Measurements
Configuration EEPROM	4K bytes	
Serial Communications	UART	0-3V transmission levels
Analog to Digital Converter	10 bit ADC	8 channel, 0-3V input
Other Interfaces	Digital I/O,I2C,SPI	
Current Draw	8 mA	Active mode
	< 15 µA	Sleep mode
RF Transceiver		
Frequency band[1]	2400 MHz to 2483.5 MHz	ISM band, programmable in 1 MHz steps
Transmit (TX) data rate	250 kbps	
RF power	-24 dBm to 0 dBm	
Receive Sensitivity	-90 dBm (min), -94 dBm (typ)	
Adjacent channel rejection	47 dB	+ 5 MHz channel spacing
	38 dB	- 5 MHz channel spacing
Outdoor Range	75 m to 100 m	1/2 wave dipole antenna, LOS
Indoor Range	20 m to 30 m	1/2 wave dipole antenna
Current Draw	19.7 mA	Receive mode
	11 mA	TX, -10 dBm
	14 mA	TX, -5 dBm
	17.4 mA	TX, 0 dBm
	20 µA	Idle mode, voltage regular on
	1 µA	Sleep mode, voltage regulator off
Electromechanical		
Battery	2X AA batteries	Attached pack
External Power	2.7 V - 3.3 V	Molex connector provided
User Interface	3 LEDs	Red, green and yellow
Size (in)	2.25 x 1.25 x 0.25	Excluding battery pack
(mm)	58 x 32 x 7	Excluding battery pack
Weight (oz)	0.7	Excluding batteries
(grams)	18	Excluding batteries
Expansion Connector	51-pin	All major I/O signals

Figure 4.9: MICAz data sheet. [32]

Using the above data the P_{tx} and P_{rx} values are calculated using following equations

$$P_{tx} = V \times I_{tx} = 17.4 \times 3 \times 10^{-3} = \quad 52.2 \quad mW. \tag{4.2}$$

$$P_{rx} = V \times I_{rx} = 19.7 \times 3 \times 10^{-3} = \quad 59.1 \quad mW. \tag{4.3}$$

The data transmission rate is given as $250kpbs$. The duration of data transmission T_{Data} and hello transmission T_{Hello} is calculated using Eqs. (4.4) and (4.5). The data packet size for this simulation was taken as 100 bytes . For the hello messages the size of hello message was taken as 50 bytes both including the protocol overhead. Using these values and the above transmission rate in the given equations resulted in transmission times of $3.2ms$ $1.6ms$ for data and hello messages respectively. The transmission and receiving power for a single data packet was calculated using $P_{tx} \times T_{Data}$ and $P_{rx} \times T_{Data}$ formulas similarly for hello messages the T_{Data} is replaced by T_{Hello}.

$$T_{Data} = \frac{\text{Data size} \times 8}{\text{Data rate } \times 1000} \tag{4.4}$$

$$T_{Hello} = \frac{\text{Hello size} \times 8}{\text{Hello rate} \times 1000} \tag{4.5}$$

4.5.3 Delivery Percentage for Leaf Nodes

The data delivery percentage for leaf nodes is measure using the Eq. (4.6). The delivery percentage was measured at nodes C and F for all simulations. Since these nodes are the furthest from the source node, delivery at the leaf nodes gives a better representation of the reliability of the schemes under investigation.

$$\text{Data delivery percentage} = \frac{\text{Total packets received at leaf node}}{\text{Total packets transmitted by source node}} \times 100\% \tag{4.6}$$

4.5.4 Dissemination Delay

The DD for a transmission cycle is defined as the duration from the time that either the source initiates the packet to the time the last node receives the packet or no more nodes re-send the packet for a single flood. DD is expressed as follows.

$$DD = \sum_{i=1}^{n} ((T_{bo}(i) + T_{tx}(i)) \tag{4.7}$$

where $T_{bo}(i)$ is the back-off time of the i^{th} transmission, T_{tx} has the same meaning as in Eq. (4.1), and n is the number transmissions experienced by a packet along the path from source to destination.

Chapter 5

Simulations and Numerical Results

The performance evaluation and comparison of proposed transmission schemes were done through extensive simulations. In this chapter, initially the different simulation scenarios considered are explained. Afterwards, the obtained numerical results for each scenario and transmission scheme is presented via graphs. A description of the behavior of observed results is also presented.

5.1 Link Quality Indicator and Test Scenarios

For the performance evaluation of proposed transmission schemes, several bit stream combinations were used. However, as discussed in Ch. 4, the evaluation of all possible bit stream combinations is not feasible. At the same time the schemes were implemented in a way that enable us to evaluate any arbitrary bit stream combination that matches the requirements in Ch. 4. Hence, from all these possibilities 5 different bitmap scenarios representing different link quality levels and link correlation values were considered for the evaluation and comparison process.

The scenarios were classified according to their overall link quality level which is termed as the link quality indicator. The classification process can be understood by looking at Tbl. 5.1.

Table 5.1: Link quality ranges with respective levels.

Quality range	Level
>80	High
60-80	Medium
<60	Poor

If the overall quality is larger than 80 %, such scenarios were considered as high quality scenarios. For overall quality between 60 % to 80 %, the scenarios were considered as medium quality. The scenarios with overall quality less than 60 % was considered as low quality. The Tbl. 5.2 presents the bit steam combinations of each link for different scenarios. The nine bits under each scenario represents the hello message results that helps to establish bitmaps in data transmission. The first four bits were used in initial data transmission phase and then on-wards the bitmaps are adjusted based on sliding window based procedure as described in Subsec. 3.2.2. The five scenarios with their quality level indicators are presented in Tbl. 5.3. The selected set of scenarios consist of one high quality , two medium quality, and two low quality scenarios. The scenario five has the same quality level as scenario 4. However, scenario 5 consist of more negative correlation links and it is selected to evaluate the behavior of schemes under weak or negative correlation situations.

Table 5.2: Bit streams of each link for different scenarios.

Link	Scenario: 1	Scenario: 2	Scenario: 3	Scenario: 4	Scenario: 5
AB	[1,1,1,1,1,1,1,1,1]	[1,1,1,0,0,1,1,1,1]	[0,1,0,1,0,1,0,1,1]	[0,1,0,1,0,1,0,0,0]	[0,1,1,1,0,0,1,1,0]
AE	[1,1,1,0,1,1,0,1,1]	[0,0,1,0,1,1,1,1,0]	[1,0,1,1,0,1,1,0,1]	[1,0,1,1,0,1,0,0,1]	[0,1,1,1,0,1,0,0,1]
BA	[1,1,1,1,1,1,1,1,1]	[1,1,1,1,1,1,1,1,1]	[0,1,1,1,1,1,0,1,1]	[0,0,1,0,0,1,0,1,1]	[0,0,1,0,0,1,0,1,1]
BE	[1,1,1,1,1,1,1,1,1]	[1,1,1,0,1,1,0,1,1]	[1,1,0,1,0,1,1,1,1]	[0,1,0,1,0,0,1,0,1]	[0,1,0,1,0,0,1,0,1]
BC	[1,1,1,0,1,1,1,0,1]	[1,0,1,1,0,1,0,1,1]	[1,1,1,0,1,0,1,0,1]	[1,0,1,0,1,0,1,0,0]	[0,0,1,0,1,0,1,0,0]
BD	[1,1,1,1,1,1,1,1,1]	[1,1,1,1,0,1,0,1,0]	[1,0,1,1,0,1,0,1,1]	[1,0,1,0,0,1,0,1,0]	[1,0,1,0,1,0,0,1,0]
EA	[1,1,1,1,1,1,1,1,1]	[1,1,1,1,0,1,0,1,1]	[0,0,1,1,0,1,1,0,1]	[0,0,1,1,0,1,1,0,1]	[0,0,1,1,0,0,1,0,1]
EB	[1,1,1,1,1,1,0,1,1]	[1,1,1,0,1,1,0,1,0]	[1,1,0,1,1,1,0,1,1]	[0,1,0,1,0,1,0,1,1]	[0,1,0,1,1,0,0,1,1]
ED	[1,1,1,1,1,1,1,1,1]	[1,0,1,1,0,1,0,1,1]	[1,0,1,1,0,1,1,0,1]	[1,0,1,1,0,1,1,0,1]	[1,0,1,1,0,1,1,0,1]
DB	[1,1,1,0,1,1,1,1,1]	[0,1,1,0,1,0,1,1,1]	[1,1,0,1,0,1,0,1,1]	[1,1,0,1,0,1,0,1,1]	[0,1,0,1,0,1,0,1,1]
DE	[1,1,1,1,1,1,1,1,1]	[1,1,1,0,1,1,1,0,1]	[0,1,0,1,1,0,1,1,1]	[0,1,0,1,1,0,1,1,0]	[0,1,0,1,1,0,1,1,0]
DC	[1,1,1,1,1,1,1,1,1]	[1,1,0,1,0,1,0,1,0]	[1,0,0,1,1,0,1,0,1]	[1,0,0,1,1,0,1,0,1]	[1,0,1,0,1,0,1,0,1]
DF	[1,1,1,1,1,1,1,1,1]	[0,1,1,0,1,1,1,0,1]	[1,0,1,1,1,0,1,0,1]	[1,0,1,0,1,0,1,0,0]	[1,1,1,0,1,0,1,0,0]
CB	[1,1,0,1,1,1,0,1,1]	[1,1,1,0,1,0,1,0,0]	[1,0,1,1,0,1,1,0,1]	[1,0,0,1,0,1,1,0,1]	[1,0,0,1,0,1,1,0,1]
CD	[1,1,1,1,1,1,1,1,1]	[0,1,1,1,1,0,1,1,0]	[0,1,1,0,1,0,1,0,1]	[0,0,1,0,1,0,1,0,1]	[0,0,1,0,1,0,1,0,1]
FD	[1,1,1,1,1,1,1,1,1]	[0,1,1,1,1,1,1,1,1]	[1,1,0,1,0,1,0,1,1]	[0,1,0,1,0,1,0,1,1]	[0,1,0,1,0,1,0,1,1]

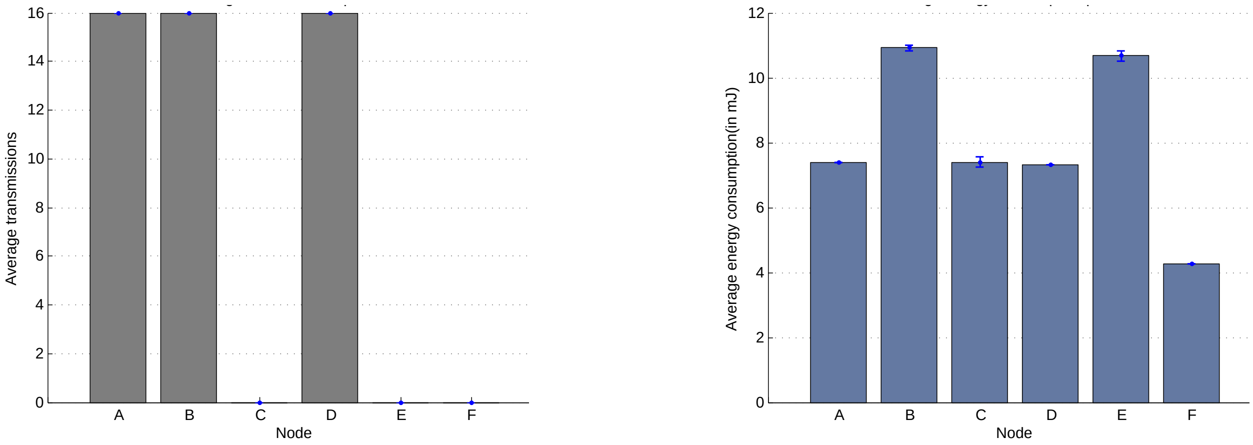
Table 5.3: Overall average quality and respective quality levels for considered scenarios.

Scenario	Average overall quality	Level
1	94.44	High
2	72.92	Medium
3	65.97	Medium
4	51	Low
5	51	Low

5.2 Number of Transmissions and Energy Consumption

The total number of transmissions and energy consumption for each node is calculated during the simulations. The simulations were run for 10 rounds and the average values of these parameters were calculated. In the following subsections the obtained results are presented using graphs for all the scenarios and all transmission schemes. The standard deviation is represented using vertical error bars.

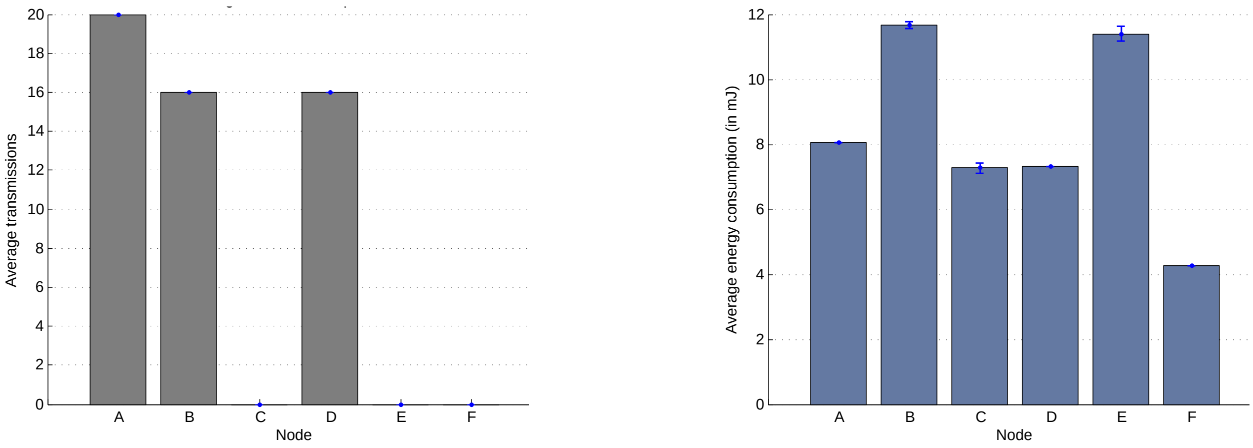
5.2.1 Scenario 1



(a) Average transmissions

(b) Average energy consumption

Figure 5.1: Per node results for BS, AS and CF in scenario 1 with error bars representing SD.



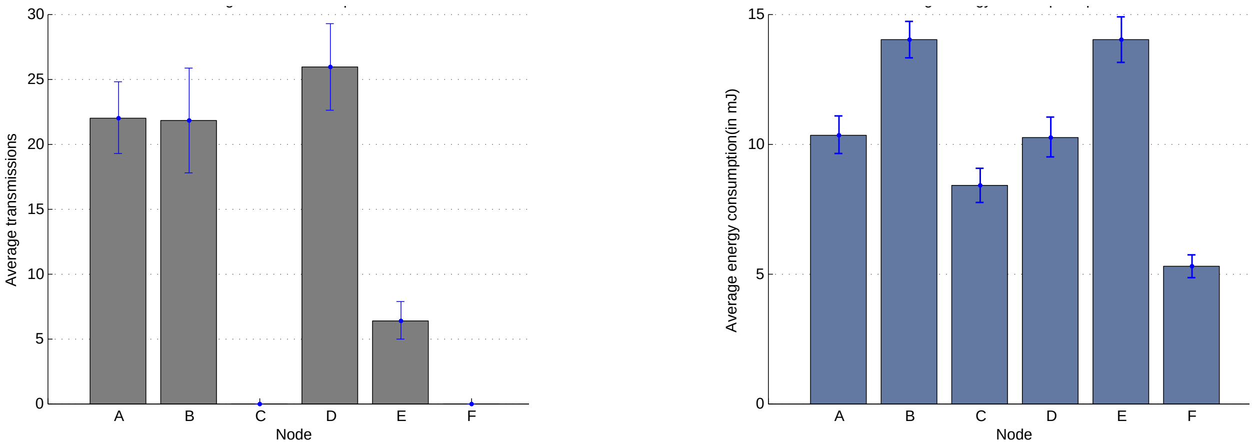
(a) Average transmissions

(b) Average energy consumption

Figure 5.2: Per node results for NLCT in scenario 1 with error bars representing SD.

Scenario 1 consists of higher quality links and it is reflected by having similar results for all schemes. BS, AS and CF all have similar number of transmissions and energy consumption. The transmissions are equally distributed between nodes A, B and D while nodes B, and E has higher energy consumption due to receiving packets from both nodes A and D. NLCT has slightly larger values since node A has taken more transmissions than other schemes.

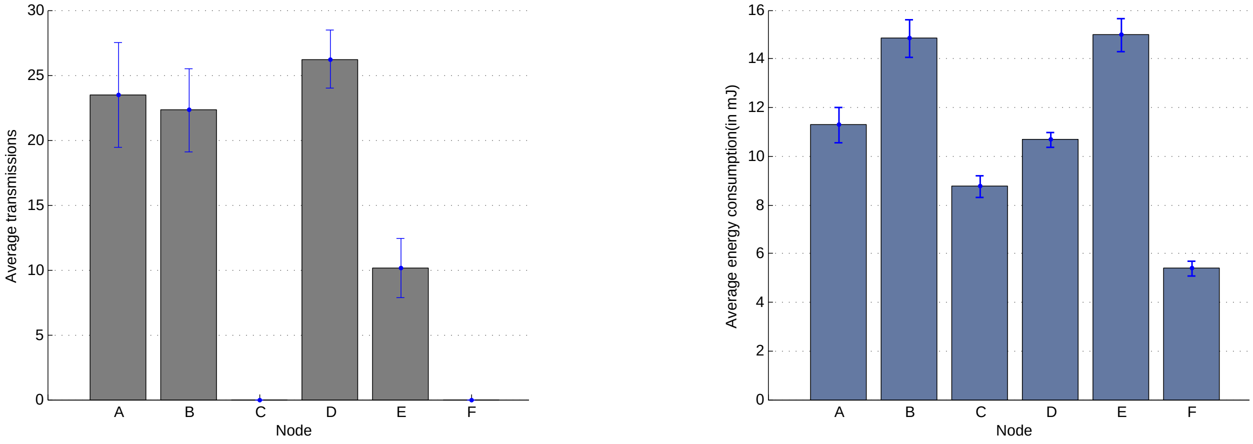
5.2.2 Scenario 2



(a) Average transmissions

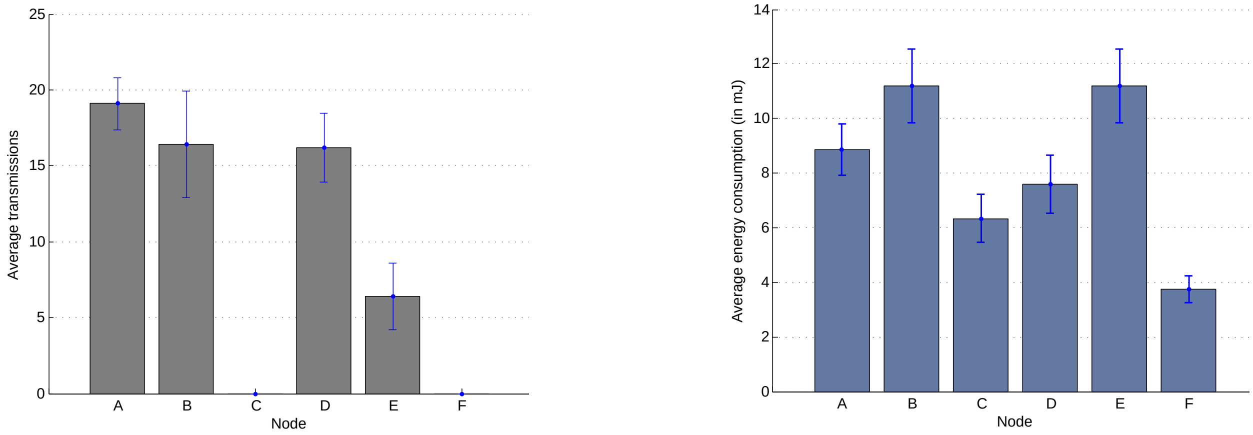
(b) Average energy consumption

Figure 5.3: Per node results for BS in scenario 2 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

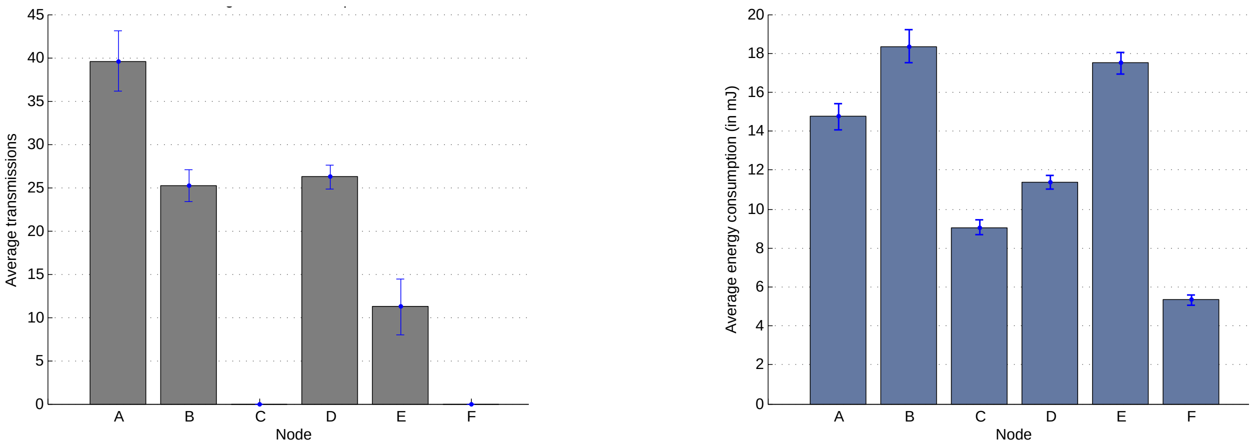
Figure 5.4: Per node results for AS in scenario 2 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

Figure 5.5: Per node results for CF in scenario 2 with error bars representing SD.

The scenario 2 consists of medium quality links. Above results shows that CF has least amount of energy consumption and transmissions among all schemes compared while NLTC has the highest values. Node D has higher number of transmissions among core nodes in BS and AS while node A has the highest number of transmissions in CF and NLCT. Nodes B and E has the highest energy consumption as seen in previous scenario.

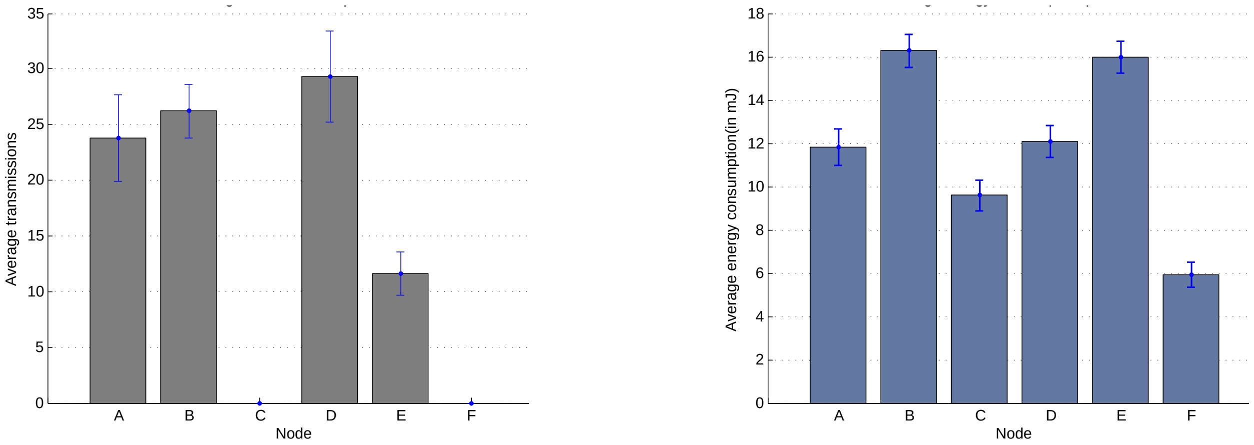


(a) Average transmissions

(b) Average energy consumption

Figure 5.6: Per node results for NLCT in scenario 2 with error bars representing SD.

5.2.3 Scenario 3

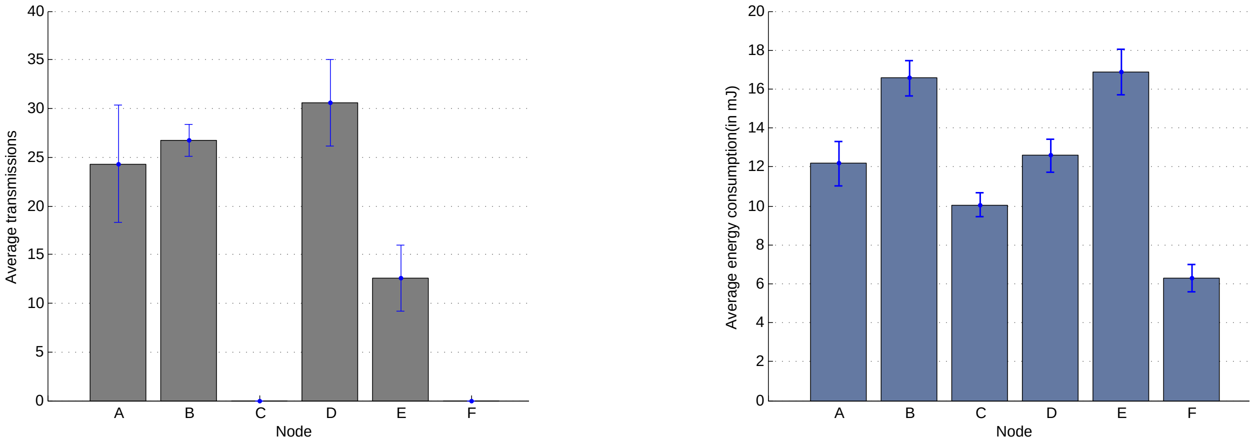


(a) Average transmissions

(b) Average energy consumption

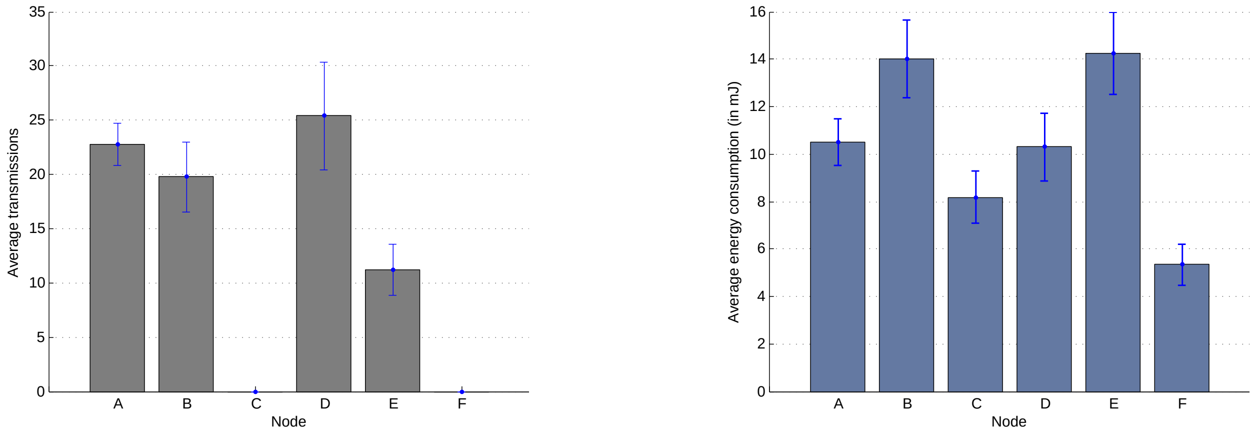
Figure 5.7: Per node results for BS in scenario 3 with error bars representing SD.

Scenario 3 also consists of medium quality links. However, it has lesser overall average quality than scenario 2. The results demonstrate the same pattern observed in previous scenario as CF resulting least transmission and consumption values followed by BS, AS and NLCT. The node D has the highest number of transmissions for all schemes while nodes B and E has the highest energy consumption.



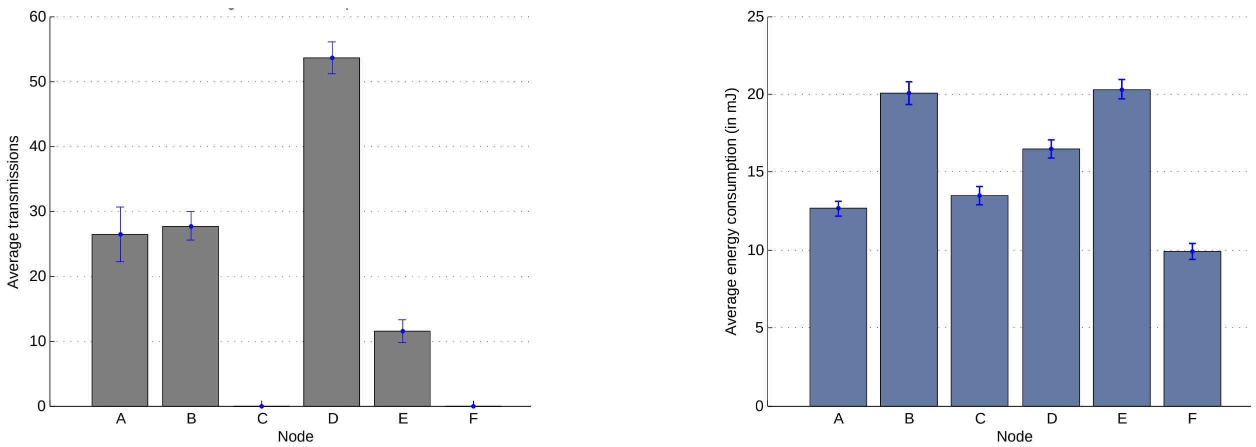
(a) Average transmissions (b) Average energy consumption

Figure 5.8: Per node results for AS in scenario 3 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

Figure 5.9: Per node results for CF in scenario 3 with error bars representing SD.

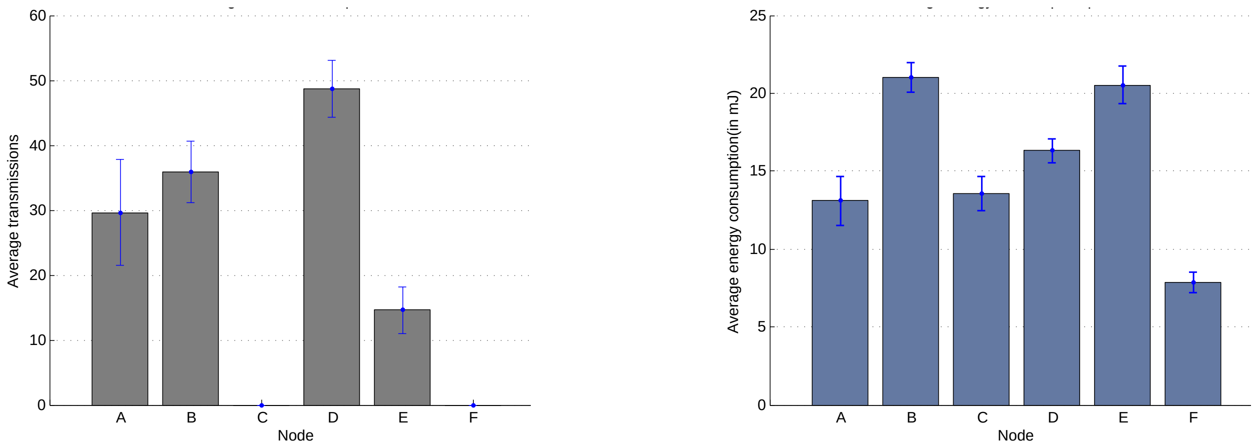


(a) Average transmissions

(b) Average energy consumption

Figure 5.10: Per node results for NLCT in scenario 3 with error bars representing SD.

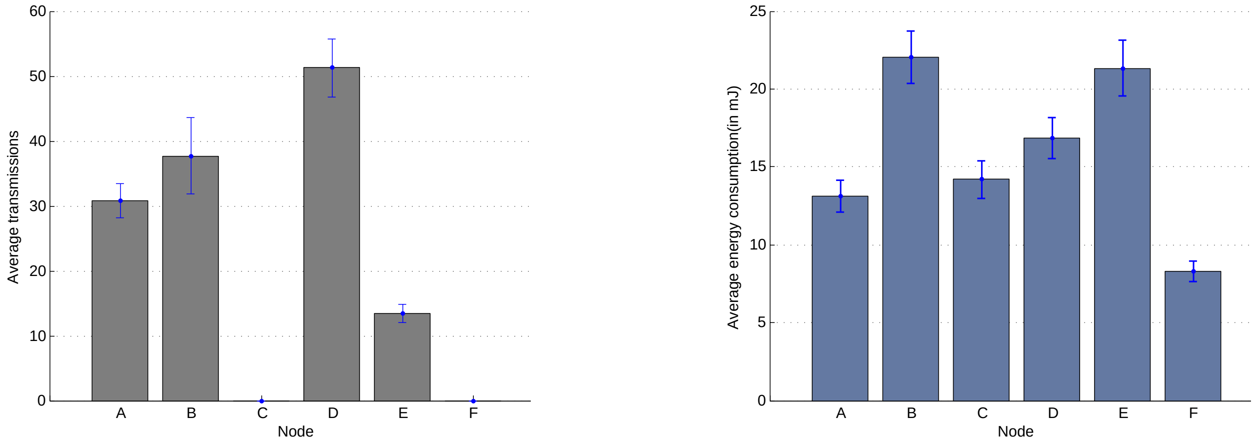
5.2.4 Scenario 4



(a) Average transmissions

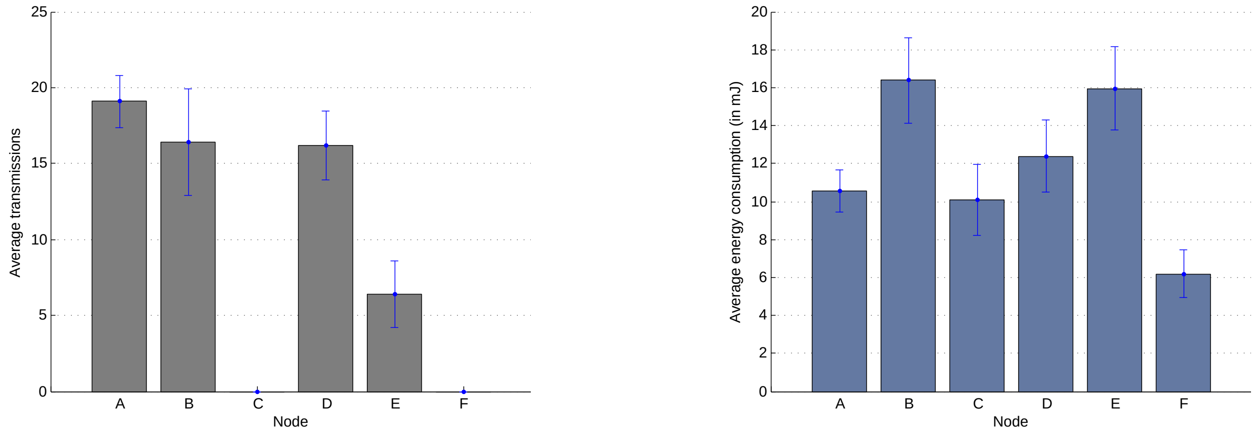
(b) Average energy consumption

Figure 5.11: Per node results for BS in scenario 4 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

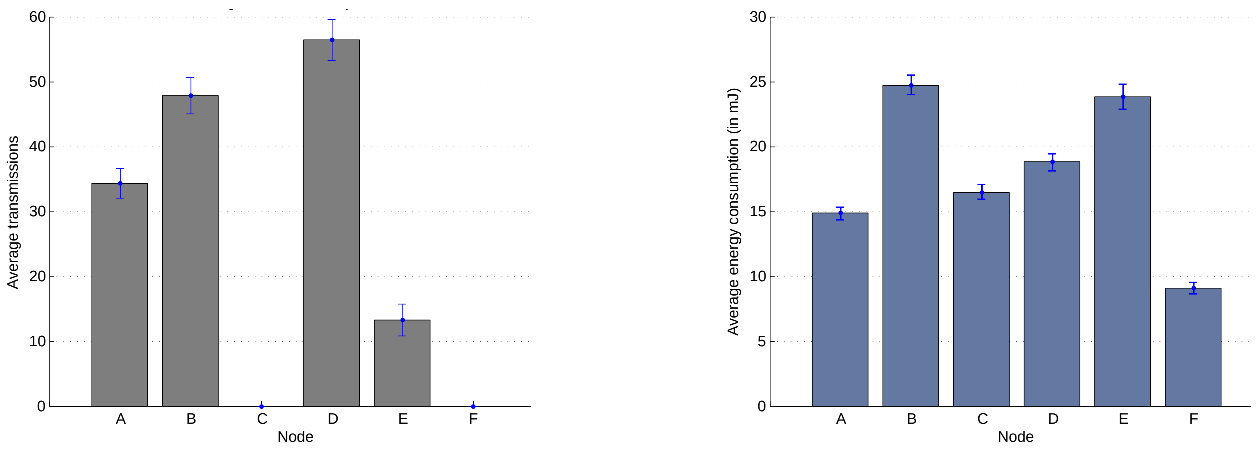
Figure 5.12: Per node results for AS in scenario 4 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

Figure 5.13: Per node results for CF in scenario 4 with error bars representing SD.

Scenario 4 consists of low quality links, which has resulted in the highest numbers of transmissions and energy consumption compared with other scenarios observed so far. The schemes follow the same pattern observed in previous 2 scenarios. The node D has the highest number of transmissions for all schemes apart from CF, where node A is the highest transmitting node. Nodes B and E remains as the nodes with highest energy consumption.

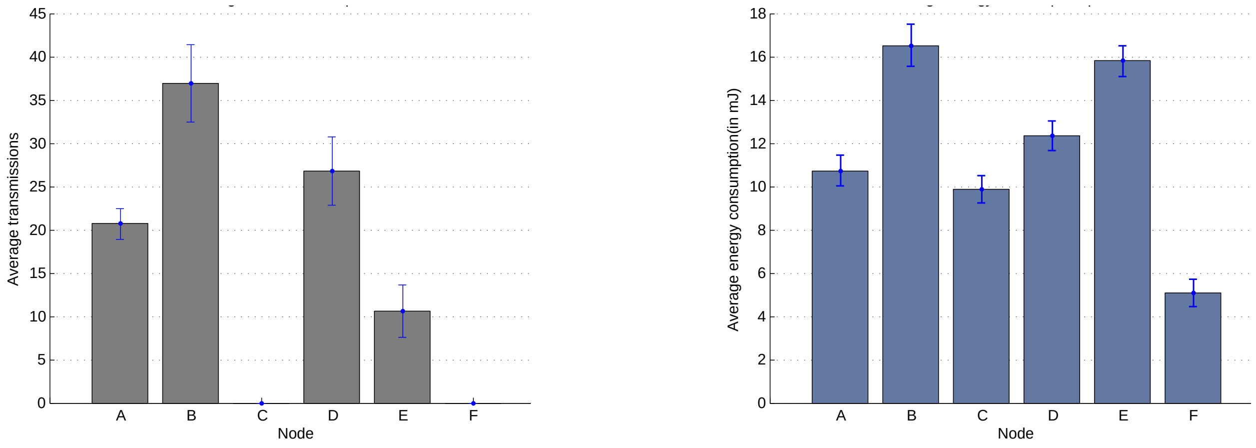


(a) Average transmissions

(b) Average energy consumption

Figure 5.14: Per node results for NLCT in scenario 4 with error bars representing SD.

5.2.5 Scenario 5

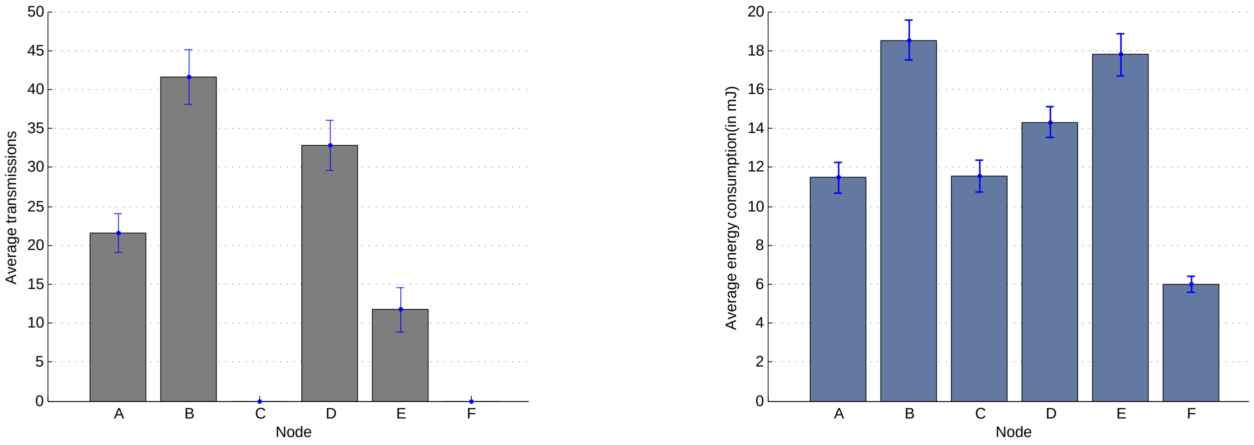


(a) Average transmissions

(b) Average energy consumption

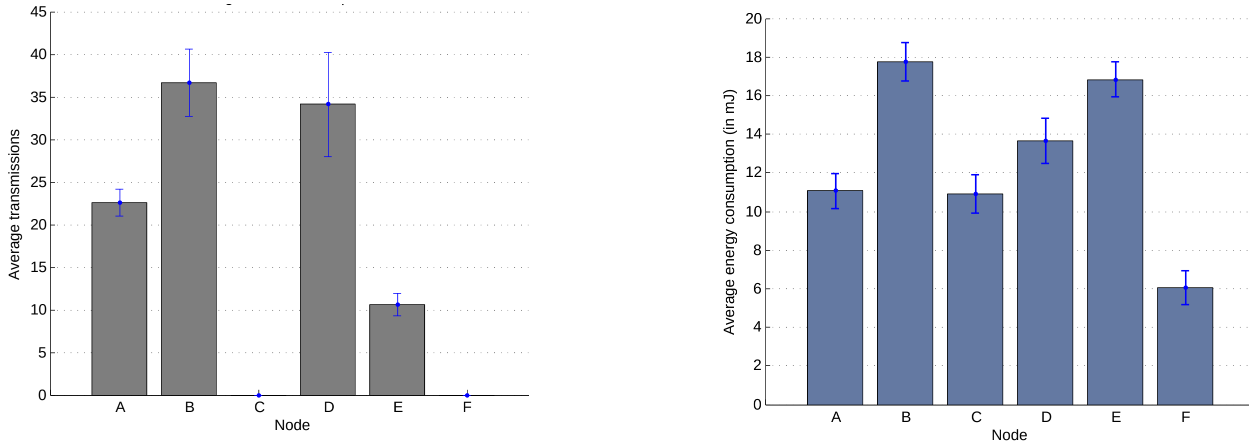
Figure 5.15: Per node results for BS in scenario 5 with error bars representing SD.

Scenario 5 also consists of low quality links with similar overall link quality level as scenario 4. However, the link quality distribution is different from scenario 4. Furthermore, some critical links in the topology have bitmaps with negative link correlation. This can be observed from looking at the initial bits in Tbl. 5.2. In this scenario, BS has the best performance in both number of transmissions and energy consumption. BS is followed by CF, AS and NLCT in that order. Here, node B does the highest number of transmissions for BS,AS and CF while NLCT has node D as the highest contributor, The energy consumption remains the same pattern observed in above scenarios with B and E leading the way.



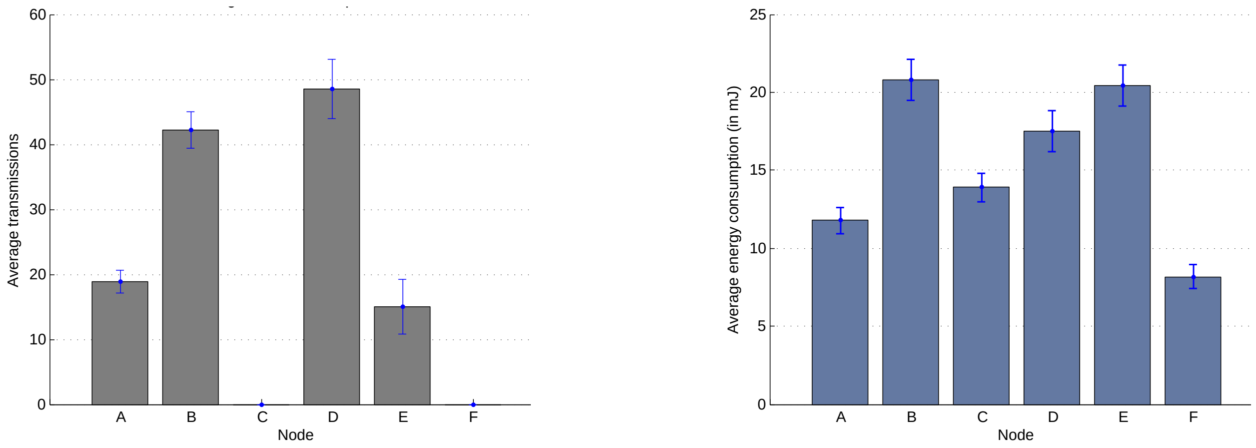
(a) Average transmissions (b) Average energy consumption

Figure 5.16: Per node results for AS in scenario 5 with error bars representing SD.



(a) Average transmissions (b) Average energy consumption

Figure 5.17: Per node results for CF in scenario 5 with error bars representing SD.



(a) Average transmissions

(b) Average energy consumption

Figure 5.18: Per node results for NLCT in scenario 5 with error bars representing SD.

5.2.6 Overall results

The average of total number of transmissions for all schemes and all scenarios is compared in Fig. 5.19. NLCT has the highest number of transmissions for all scenarios while CF has lower number of transmissions for scenarios 2, 3, and 4.

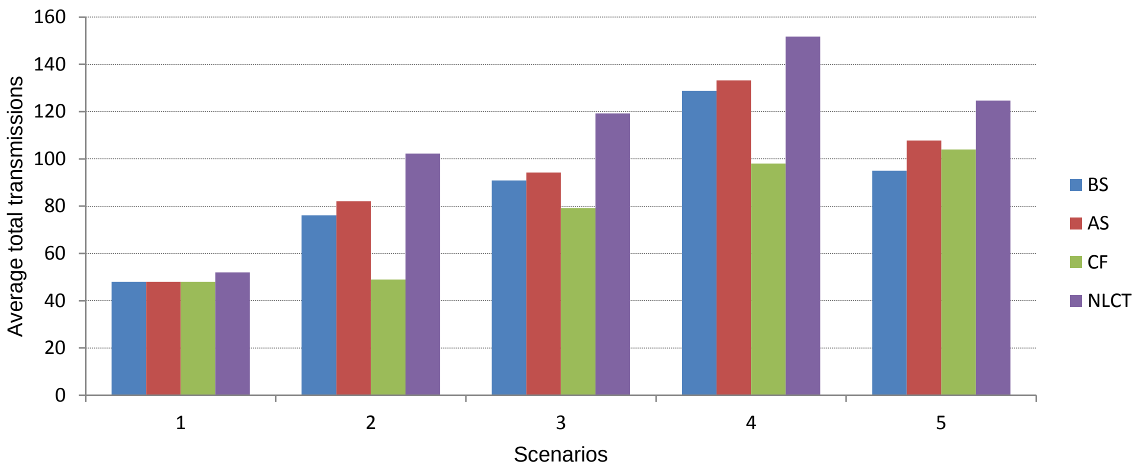


Figure 5.19: Average total number of transmissions.

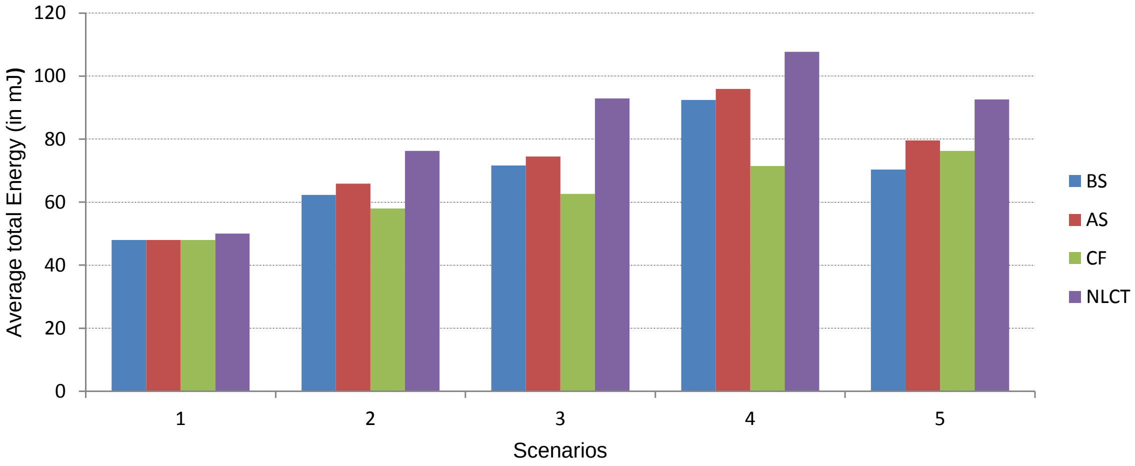


Figure 5.20: Average total energy consumption.

In scenario 5, when there are high negative correlated links, CF's transmissions seems to increase compared with other schemes. The average total energy consumption results presented in Fig. 5.20, also follow the same trend. BS over performs CF in scenario 5 while it has equal or higher number of transmissions and energy consumption than CF in other scenarios. AS in all cases apart from scenario 1 consumes higher energy than CF and BS. This is understandable since higher reliability comes with a higher energy and transmission cost.

5.3 Data Delivery Percentage

The purpose of data dissemination is to cover all nodes in a given topology. The energy consumption and number of transmissions metrics alone will not give an accurate picture about how well a transmission scheme performs. Hence, the data delivery percentage at the leaf nodes was calculated for all transmission schemes. Figs. 5.21 and 5.22 illustrate the results obtained for all scenarios mentioned. The AS as expected always delivers packets with 100% reliability. At the same time, BS and NLCT has similar packet delivery percentages. However, the packet delivery percentage of CF is much lesser than all other schemes considered. Although CF has lesser energy consumption and transmission results, it seems to have come at a cost of lesser data delivery percentage.

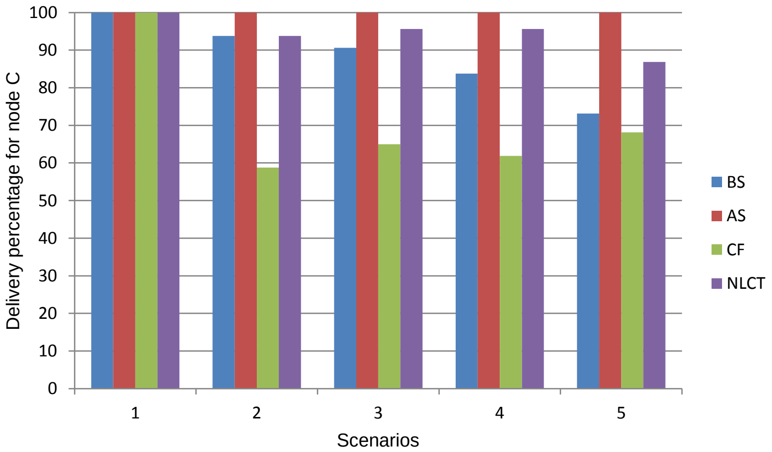


Figure 5.21: Data delivery percentage for at C.

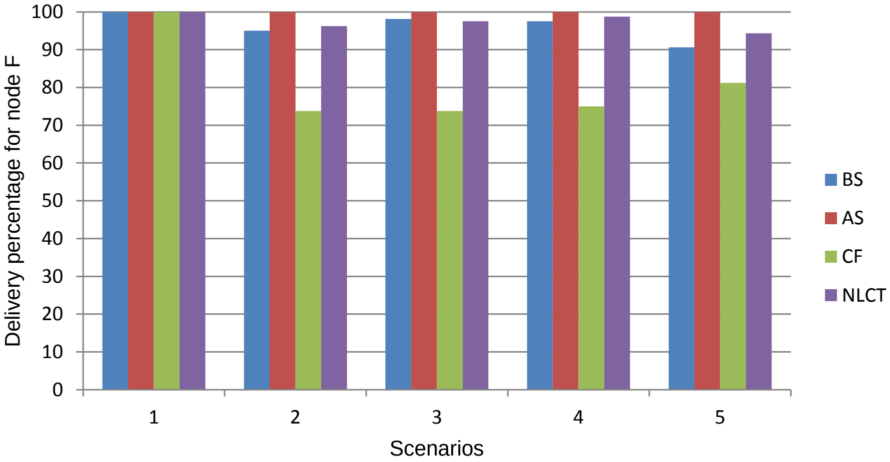


Figure 5.22: Data delivery percentage for at F.

Similarly, NLCT's higher delivery percentage comes at the cost of high energy consumption. Meanwhile, BS seems to reach higher delivery percentages with a smaller additional cost of energy as observed earlier. When weighing the suitability of a transmission scheme it becomes a trade off between many factors that need to be evaluated. When energy consumption and data delivery is considered, minimizing energy consumption will not achieve the required objectives if the data have not reach all nodes in a network. Hence, when considering the energy consumption, number of transmissions and data delivery , BS seem to be a better choice as a transmission scheme. The ability to shift to AS when required also makes BS a better candidate for applications with dynamic network requirements.

5.4 Dissemination Delay

The DD values are calculated for AS using Eq. (2.5). Bitmap 3 with medium quality values is selected for this DD calculation. The Fig. 5.23, illustrates the DD values for packet sequence numbers which have been successfully transmitted at the first attempt. This means that, these packets have reach all nodes in the topology during the original transmission cycle and no re-transmissions using SR or ER phase have been done. Hence, the values contains the total back-off values and the data transmission times for the considered packet. In Fig. 5.24, the DD for the remaining packets are shown. These packets have not reached all nodes on their original transmission. At least one node has not received the packets and hence it has been re-transmitted using SR and ER phases. Since our SR and ER phase comes after few data cycles, the delay has been multiplied by several inter packet interval values. For example, packet with sequence number 1 has been successfully reached all nodes at the SR phase after 16 packets have been transmitted. As a result this packet has the largest delay value while a packet like number 15, has a lesser delay due to immediate re-transmissions after 16 packets.

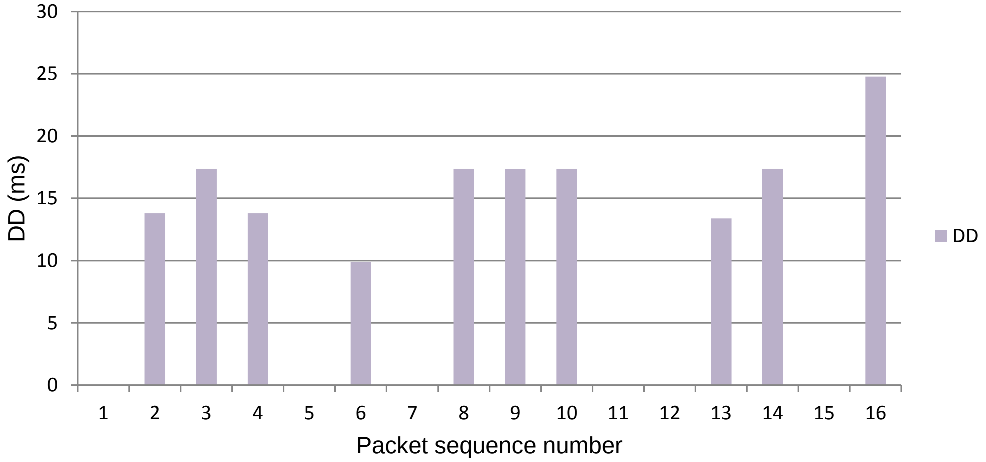


Figure 5.23: DD for successfully transmitted packets at the first attempt in scenario 3.

The results also suggests some interesting points regarding the latency of AS scheme. Although AS is capable of achieving 100 $\%$ reliability, the DD of the re-transmitted packets could be a large value depending on the implementation point of ER scheme. For delay sensitive applications ER phase should be used in more frequent intervals to avoid this problem. If the application does not care about the DD, and the sole purpose is to have all packets reach all nodes in the topology, then the ER can be implemented as done in this simulations.

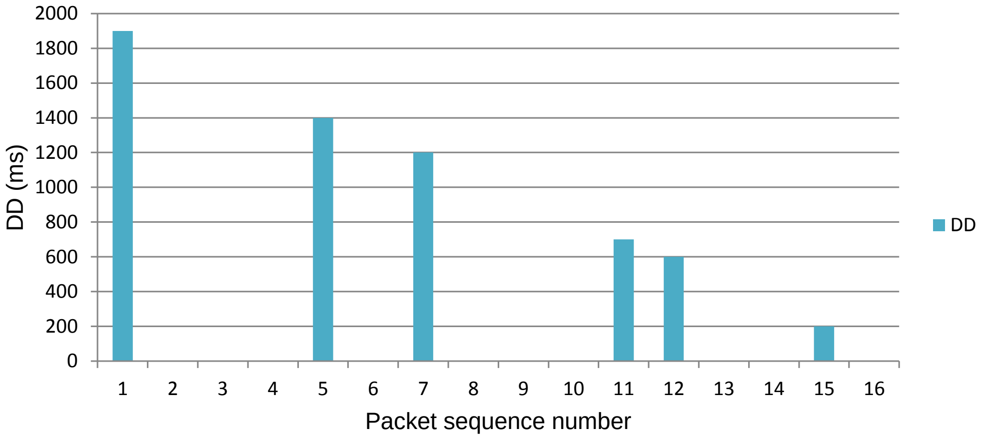
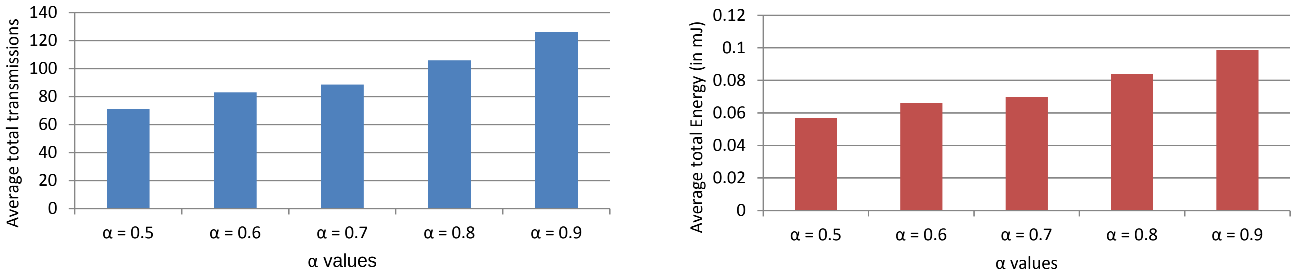


Figure 5.24: DD for re-transmitted packets using SR and ER in scenario 3

In addition, impact of estimation based transmission decision making can be observed from the given figures. Fig. 5.23 represents all packets that were successfully transmitted from the estimation of CP. However, this estimation process has not able to deliver the packets in Fig. 5.24. These packets have been delivered due to the application of explicit ACKS through SR and ER phases. Hence, the combination of estimation methods and use of ACKs has resulted in giving a higher reliability for the AS scheme. This is one major advantage of AS over schemes like CF where only estimation is used for packet delivery.

5.5 Performance of BS with Different α values

The CP threshold value α plays a major part when deciding the number of transmissions needed to cover a neighbor node. The impact of α on the performance of BS was investigated by changing the α values and observing the results for number of transmissions, energy consumption and data delivery percentage. The results are presented in Fig. 5.25 and 5.26 respectively. This results were obtained considering the scenario 3 with medium overall link quality values.



(a) Average transmissions

(b) Average energy consumption

Figure 5.25: Total results for BS in scenario 3 with different α.

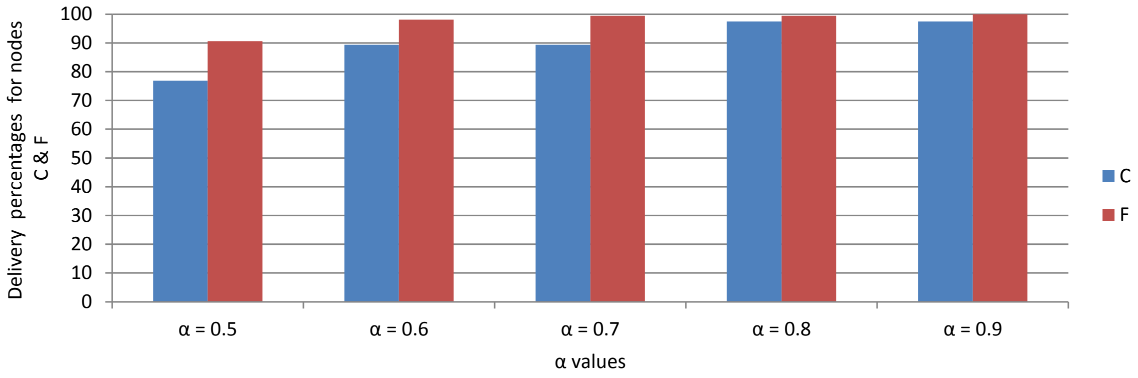


Figure 5.26: Data delivery percentage for at C and F nodes with different α in scenario 3.

According to the observed results, it is evident that the number of transmissions and energy consumption is increasing with increasing α. At the same time the delivery percentage also improves with increasing α values. This results seems intuitive as the higher values of α causes nodes to transmit more times causing high transmission and energy cost, while reaching better delivery percentages. At the same time in Fig. 5.25, when α is 0.5, BS has achieved lower energy consumption, transmissions and higher delivery percentage than CF when compared with CFs results on scenario 3 on Figs. 5.9, 5.21 and 5.22. Hence, according to this analysis BS is capable of providing a higher delivery percentage than CF even at low α values. This result further emphasize the observation that was done during last section. Having bulk ACKs through hello messages in addition to coverage estimation provides BS a higher chance of delivering to leaf nodes. Since CF only uses CP estimation, the delivery percentage is much lesser than BS.

Chapter 6

Discussions

The results observed in the previous chapter gives a great insight on the behavior of proposed schemes. This chapter presents an in-depth discussion about the obtained results and the overall observations during the thesis work.

6.1 Advantages of BS and AS

According to the results of the previous chapter, it can be seen that BS and AS can be useful to achieve higher reliability with lesser number of transmissions and energy consumption cost. BS, even at lower α values has a higher delivery percentage to leaf nodes when compared to CF with higher α values. Hence, BS is capable of reaching higher reliability levels with comparatively low transmission and energy consumption values. At the same time if ER is enabled BS can be easily converted in to AS when 100% reliability is required. AS gives this reliability for a lower number of transmissions and energy cost compared with schemes like NLCT which has very high number of transmissions.

6.1.1 Utilizing Link Correlation

BS and AS uses the information through hello messages to obtain bitmaps and to calculate the link correlation levels between the links using the κ factor [8] . This information is used to estimate the CP of neighbor nodes using the Alg. 2 proposed in Subsec. 3.2.4. Also, the use of correlation enables to use the concept of collective acknowledgments [7] which minimizes the transmissions required for neighbor coverage. When compared with non-correlation based schemes like NLTC, the correlation consideration helps to reduce the number of transmissions considerably and minimizes the overall energy consumption. The comparison with CF also helps to draw some conclusions about the performance of proposed schemes. In [7], the CF scheme is compared with few other existing non-correlation based transmission schemes like Reliable Broadcast Propagation [3] and Double-Covered Broadcast (DCB). Their results indicated that CF has better performance than both these schemes in terms of energy consumption and reliability. Hence, the performance of BS and AS schemes in comparison to CF gives us a general idea of its performance when compared with these other schemes indirectly. The ability to match or better CF's performance bodes well for the proposed schemes performances.

6.1.2 Adaptive Backbone Advantages

With the adaptive backbone procedure, supporting core nodes B and E helps the main core node D to forward data towards the leaf nodes. If a traditional CDS core based backbone was used, nodes B and D will do all the transmissions and node E will not contribute towards the data forwarding process. In such a situation, node B will have to transmit in each data transmission cycle resulting in an increment of the energy consumption of node B. When we analyze the energy consumption results in Ch. 5 it is evident that node B and E has the highest consumption values for each scenario considered. Hence, if node E was not used for data transmission, node B's energy consumption will further increase causing an further unbalanced load distribution among nodes and possibly causing an energy hole at node B. This situation has been avoided in BS and AS by assigning both B and E nodes as supporting core nodes. Depending on the reception of A's transmissions either node B or E will adapt to forward the packet to D and the forwarding process will change according to link quality and correlation values in neighboring links.

6.1.3 Advantages of Hop Count based Back-off Calculations

The proposed schemes utilizes the hop count distance from the source node when calculating the back-off waiting times for contention based channel access. This proposed algorithm gives priority to competing nodes further away from source node to transmit first. By doing so, it gives an opportunity for the intermediate nodes to overhear this transmission as an implicit ACK and update their CP values. Hence, nodes like B and E who are in competition due to their estimation of CP on D could hear such implicit ACK, and may exit from competition with out further re-transmissions. The overall process will help to reduce the unnecessary intermediate transmissions done by nodes like B and E and also contribute to move the data packet forward (further away from source node) in less time.

6.2 Impact of Negative Correlation

Correlation based transmissions schemes could experience performance degradation when the critical links in the topology have negative correlation. This impact for CF was identified by the our previous work in [33]. The results of scenario 5 also confirm this findings as CF suffers a comparatively higher performance degradation when compared with other scenarios. However, the aforementioned impact is much lesser on BS and AS schemes. The reason for this behavior is the correlation based data transmission algorithm proposed in Alg. 2 which uses the CP of transmitting nodes to update neighbor CP values in addition to collective ACKs. Therefor, the proposed schemes are not over reliant on correlation in the decision making process which helps them to achieve higher reliability with lesser number of transmissions in negative correlation scenarios.

6.3 Revisit the Energy Hole Problem

In traditional research, it is noticed that the nodes closer to sink nodes deplete their energy far more quicker than the other nodes. This scenario is described as an energy hole situation. In most many to one type transmissions the nodes one hop away from the sink nodes mostly suffer from this high energy depletion [34], [35]. Meanwhile some research work [34] has shown that, by the time one hop neighbors to sink nodes deplete their energy, the nodes further away from sink nodes has upto 93 $\%$ energy remaining. This uneven depletion of energy causes the network life time to reduce drastically and the energy hole becomes a bottle neck in the topology. When we analyze the topology under consideration, since node F has node D as the only neighbor , node D always transmits atleast once to cover node F. When we analyze the results for total number of transmissions of individual nodes it can be seen that node D has the highest number of transmissions in most of the scenarios. So going by the traditional knowledge it is fair to assume node D as the energy hole in this topology since it has the most number of transmissions and it's one hop away from the two leaf nodes. However, when we analyze the energy consumption results, almost all the time nodes B and E has higher energy consumption than node D. Hence, if the transmissions takes place for a certain time period, there is a high probability of node B or E being depleted from energy before node D. This observation can be explained by the higher number of receptions happening at both B and E nodes. Since nodes B and E are in the middle of the topology they are subjected to overhear the transmissions from nodes A, D and of each other. This adds up with their transmission energy increasing the overall energy consumption. When both nodes B and E are depleted from energy there occurs an energy hole in the topology.

The impact of receiving energy is further magnified by the higher receiving power consumption in modern sensor nodes. When we analyzed the most commonly used sensor node data sheets [32], [36] was observed that the receiving power consumption is higher than the transmit power consumption. The reason for the higher receiving energy is explained in [37] as follows. For short range communication, the difference between transmit and receive energy levels are very small. Due to the sophisticated modulation techniques the decoding energy dominates the encoding energy. As a result the receiving power is higher than the transmission power in modern sensor nodes.

This observation forces us to reconsider our perceptions on energy holes. Energy hole may not necessarily be the node that transmits the highest. It could be a node that is a neighbor of this highest transmitting node which has larger number of receptions who is also participating in transmissions.

For the proposed schemes the energy depletion of nodes could be reduced by introducing sleeping patterns in between data transmissions. For example, if a node has received a certain packet and if all its neighbors are covered it can goes to sleeping mode. For the next data transmission phase that node should be waking up at the negotiated time. This would eliminate the reception of unnecessary duplicates of the same packet and save the reception energy. Same can be used at SR and ER phases. However, implementation of this concept needs strict time synchronization between nodes for for data transmissions with smaller inter packet times. If the synchronization challenges are addressed the sleeping nodes will save more energy and increase the life time of the network.

6.4 Effect of Bitmap Size

The bitmap size for the simulations has been taken as 4. The κ factor calculations and link quality calculations depend on this bitmap size. When considering the κ factor higher number of hello message results should provide a better statistical value when calculating the link correlation factor . However a larger bitmap size means that it consists of older hello message information . For example if we take bitmap size as 8 and exchange the hello messages in the same time intervals as before, then when calculating the correlation, the old information could have a high impact on the actual correlation results. The recent correlation trends could be overshadowed by this inclusion of older hello messages which sometimes could be obsolete. Hence, there is always a trade off between a having higher number of bits to make the κ factor more accurate and the validity of old hello information in a higher bitmap size. Bitmap size could also be increased with more frequent hello messages so that the messages reflect current channel status. However, this will also increase the energy consumption of individual nodes due to frequent hello exchange. Hence,the selection of the bitmap size should be done considering all these factors into account.

6.5 Scalability of the Proposed Schemes

As mentioned in Ch. 3, the proposed schemes are scalable for much larger networks with higher node number and density. The fundamental concepts can be extended for any node number and distribution. Some parameters needed to be calibrated according to the network size and packet distribution requirements. For larger networks when the hop count is large, the back-off constant C can be selected according to the node distribution. If the network is dense with large number of nodes, the nodes could be clustered and assigned different C values for each cluster to decrease the back-off time difference between nodes closer to source node and nodes further away. The C value should be carefully selected to avoid very small back-off times at places far away from source node. It should be noted that although the back-off times decrease with hop count, the competition only exists between neighboring node set which can overhear each other. So having larger back-off value near source nodes and smaller ones far away will not limit the nodes nearer to the source from transmitting when necessary. It will only delay these transmissions while trying to minimize the unnecessary ones.

The enabling of ER in AS scheme also should be performed considering the number of packets needs to be transmitted. For smaller number of packets ER phase could be enabled at the end of data transmissions as did in the simulations. For larger data or for latency critical applications, ER could be enabled more frequently in parallel with SR phase so that nodes can delete their copies of received packets afterwards. This would save some memory of nodes and also decrease the overall dissemination delay.

6.6 Further Discussions

6.6.1 Using Data Packets for Bitmaps

The proposed schemes use the periodically exchanged hello messages to update the bitmap information on neighbor links. An alternative method for updating the bitmap is to utilize the results of data messages transmitted. Then, when ever a data message is received its sequence number can be added to the set of received data messages. Thus, this results of data messages can be conveyed to neighbor nodes though a periodic hello message. Since the data is transmitted more frequently than hello messages, the bitmap information would be more accurate and more recent updates can be gained. However, since only core nodes does data transmissions, updating the reverse links from non transmitting nodes becomes a problem in this approach. Hence, it would be better if a hybrid system that consist of hello and data messages can be used for updating the link bitmaps.

6.6.2 Consideration for Adapting Temporal Correlation

The proposed schemes utilizes spatial link correlation in order to gain collective ACKs in data transmission phase. However, there is also temporal correlation that can be taken in to account. In [38] the authors claim that the most intermediate links in 802.15.4 test-beds are bursty. Hence, they propose a β metric to measure the link burstines that could effect the protocol performances. This β metric could be utilized for having a measurement of temporal correlation. However, for an accurate β measurement the inter packet arrival times has to be very small. If the data packets can be used to establish bitmaps as discussed in previous subsection, there could be a possibility of having smaller inter arrival times and measure the β factor using it.

Chapter 7

Conclusions and Future work

From extensive simulations and analysis, we were able to reach several conclusions about the proposed transmission schemes. In this chapter, the thesis is concluded by summarizing these main conclusions and contributions. In addition, some future research directions that were identified are also presented.

7.1 Conclusions

The main objective of this thesis was focused on proposing novel data transmission schemes for WSNs, that could improve the overall performance of the network. For achieving this goal, initially the existing transmission schemes and concepts related to WSNs were studied in detail. In a WSN, all transmission schemes attempts to reduce the number of transmissions and energy consumption, while reaching required levels of reliability. In order to achieve this goals, most of the transmission schemes rely on the independence assumption among wireless links. Recent research has shown the existence of wireless link correlations and its implications on wireless systems. Since this concept hasn't widely used for data transmission schemes we decided to utilize this and many other novel and existing concepts to propose two new data transmission schemes.

The proposed schemes contains several novel concepts and also extends some existing concepts. BS and AS utilize the concept of wireless link correlation for determining the number of transmissions that is required to cover neighboring nodes. Furthermore, their data transmission is performed by selected set of core nodes which adapts with receptions and not limited to traditional CDS themes. In addition,the hop count based novel back-off algorithm was proposed for channel contention while the SR and ER phases was introduced to re-transmit missing packets. The proposed schemes were implemented in MATLAB for performance evaluation purposes. Two other transmission schemes in

CF and NLTC, were also implemented for comparison purposes. The simulations were done for 5 different scenarios representing different link quality and correlation levels.

Overall, the correlation based schemes showed much better performance in-terms of energy consumption and number of transmissions when compared to non-correlation based NLCT scheme. The observed results showed that BS and AS can achieve a high level of reliability when compared to CF which lags behind in reliability. However, proposed schemes lag behind CF in terms of energy consumption and number of transmissions in majority of scenarios. Further analysis using different coverage threshold α values reveled that, under low α values, BS is capable of reaching higher reliability levels while having less transmission and energy consumption compared with CF. This observation emphasizes the ability of BS to reach higher reliability levels even under low threshold levels. The use of neighbor coverage estimation and SR phase for re-transmissions enables BS to reach higher reliability which CF cannot afford due to the sole dependence on coverage estimation. On the other hand, AS scheme demonstrated its ability to reach 100 $\%$ reliability in all test scenarios. However, the dissemination delay of the re-transmitted packets could vary depending on the point of time when ER phase in AS is enabled. Furthermore, in negative correlation scenarios, proposed schemes performed considerably better than CF in both energy consumption and number of transmissions, while maintaining higher reliability levels. The results emphasize the trade off between higher reliability and the energy consumption. Overall, the ability to reach high reliability levels at a comparatively low energy cost, ability to switch between BS and AS depending on the reliability requirements, and better performances in negative correlation scenarios, gives the proposed schemes an advantage over the existing schemes. In addition to above observations, from further analysis of results we demonstrated the possible energy hole scenarios in the considered topology which is different from the traditional perception of energy-holes. We observed that, due to the higher receiving powers in modern sensor nodes, the receiving power could play a dominant role than the transmission power in creating an energy hole. Hence, the neighbor nodes of the highest transmitting node are most vulnerable candidates to an energy hole situation.

Overall, from this thesis work, we were able to achieve the initial goals defined. We believe that the proposed schemes and their components would contribute towards advancement of WSN transmission schemes.

7.2 Main Contributions

The main contributions of this thesis are summarized below.

- In this thesis, two novel transmission schemes for data transmission in WSNs is proposed. The proposed schemes can reach high reliability levels with comparatively low energy and transmission costs.
- The κ factor for link correlation calculation is utilized and extended in the proposed algorithm to calculate the receiver CP values of neighbor nodes in the network.
- An adaptive backbone concept is proposed that extends the current CDS concepts. The proposed backbone consists of main core nodes and supporting core nodes which helps to achieve more balanced load distribution among core nodes.
- A hop count based back-off calculation method is proposed for channel contention process. Proposed method helps to minimize unnecessary intermediate re-transmissions and also propagates the data packet forward in quick succession.
- The SR phase is designed to help nodes re-transmit the missing packets in predefined intervals. The nomination of the sender makes sure that the next re-transmission happens in the best available link and hence, avoiding extra re-transmissions.
- The effect of higher receiving power of modern sensor nodes on energy hole determination is identified and presented.

7.3 Future Work

Several possible directions for future work was identified during the thesis work. Some of the following has being analyzed more detailed in Ch. 6.

- For this thesis work,proposed transmission schemes were evaluated using MATLAB based simulations. However, more accurate results could be obtained by implementing the schemes on a real-life sensor test-bed.
- For the proposed schemes, only spatial correlation between the links in topology were considered. It is possible to extend or modify the schemes considering temporal correlation.

- For the current schemes, the sleep wake-up duty cycles for sensor nodes have not been considered. If proper time synchronization among nodes is possible, it will be beneficial to apply sleep wake-up cycles to reduce energy consumption by avoiding unnecessary overhearing.

- Researching on the possibility of using data packets alongside hello messages to establish bitmaps would be another area that could contribute towards the advancement of proposed schemes.

Bibliography

[1] Y. Li, M. Thai, and W. Wu, *"Wireless Sensor Networks and Applications"*, 2008.

[2] REUTERS, "Wireless sensor network technology trends 2014:higher growth by wireless sensor network market." [Online]. Available: http://www.reuters.com/article/2014/02/27/research-and-markets-idUSnBw276536a+100+BSW20140227.

[3] F. Stann, J. S. Heidemann, R. Shroff, and M. Z. Murtaza, "RBP: robust broadcast propagation in wireless networks." in *SenSys*, 2006, pp. 85–98.

[4] O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, and V. K. Sadekar, "On the broadcast storm problem in ad hoc wireless networks." in *BROADNETS*, 2006.

[5] D. H. Du, W. Wu, Q. Ye, D. Li, W. Lee, and X. Xu, "CDS-based virtual backbone construction with guaranteed routing cost in wireless sensor networks." *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 652–661, 2013.

[6] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proc. 25th IEEE/INFOCOM, Barcelona, Spain, pp.1-12*, Apr. 2006.

[7] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Achieving efficient flooding by utilizing link correlation in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 121–134, Feb. 2013.

[8] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari, "The k factor: Inferring protocol performance using inter-link reception correlation," in *Proc. 16th ACM/MobiCom, Chicago, USA, pp.317–328*, Sept. 2010.

[9] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Mag.*, vol. 42, no. 10, pp. 74–80, Oct 2004.

[10] S. Wang and J. Nie, "Energy efficiency optimization of cooperative communication in wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, pp. 3:1–3:8, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/162326

[11] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets." in *ICC*, 1997, pp. 376–380.

[12] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: a core-extraction distributed ad hoc routing algorithm,” *IEEE Journal on,Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, Aug 1999.

[13] Y. Wu and Y. Li, “Connected dominating sets.” [Online]. Available: http://www.cs.gsu.edu/yli/papers/cds-chapter.pdf.

[14] M. V. Marathe, H. Breu, H. B. H. III, S. S. Ravi, and D. J. Rosenkrantz, “Simple heuristics for unit disk graphs.” *Networks*, vol. 25, no. 2, pp. 59–68, 1995.

[15] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets,” *Algorithmica*, vol. 20, pp. 374–387, 1996.

[16] J. Wu, B. Wu, and I. Stojmenovic, “Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets.” *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 425–438, 2003.

[17] P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed construction of connected dominating set in wireless ad hoc networks.” in *INFOCOM*, 2002.

[18] F. Y. Li, “WOSPF and other routing protocols,” 2014.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks.” in *HICSS*, 2000.

[20] E. Ziouva and T. Antonakopoulos, “The IEEE 802.11 distributed coordination function in small-scale ad-hoc wireless lans.” *IJWIN*, vol. 10, no. 1, pp. 1–15, 2003.

[21] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in *Proc. MobiCom*, 2000, pp. 56–67.

[22] A. Khan, “Data fusion in sensor networks.” [Online]. Available: http://www.cse.buffalo.edu/~qiao/cse620/fall04/Data_Fusion.ppt

[23] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proc. MobiCom*, New York, NY, USA, 1999, pp. 151–162.

[24] Y. B. Ko and N. H. Vaidya, “Flooding-based geocasting protocols for mobile ad hoc networks,” *Mob. Netw. Appl.*, vol. 7, no. 6, pp. 471–480, Dec. 2002.

[25] L. Alvisi, J. Doumen, R. Guerraoui, B. Koldehofe, H. Li, R. van Renesse, and G. Tredan, “How robust are gossip-based communication protocols?” *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 14–18, Oct. 2007.

[26] W. Lou and J. W. 0001, “Double-covered broadcast (DCB): A simple reliable broadcast algorithm in MANETs.” in *INFOCOM*. IEEE, 2004.

[27] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, “Correlated flooding in low-duty-cycle wireless sensor networks,” in *Proc. 19th IEEE International Conference on Network Protocols (ICNP), pp.383-392*, Oct 2011.

[28] S. Wang, S. Kim, Y. Liu, G. Tan, and T. He, “CorLayer: A transparent link correlation layer for energy-efficient broadcast,” *IEEE/ACM Trans. Networking*, vol. PP, no. 99, pp. 1–1, 2014.

[29] S. Alam, S. Sultana, Y. Hu, and S. Fahmy, “SYREN: Synergistic link correlation-aware and network coding-based dissemination in wireless sensor networks,” in *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, Aug 2013, pp. 485–494.

[30] S. Wang, A. Basalamah, S. M. Kim, S. Guo, Y. Tobe, and T. He, “Link-correlation-aware opportunistic routing in wireless networks,” *IEEE Trans. Wireless Communications*, vol. 14, no. 1, pp. 47–56, Jan 2015.

[31] J. Jun, L. Cheng, L. He, Y. Gu, and T. Zhu, “Exploiting sender-based link correlation in wireless sensor networks,” in *International Conference on Network Protocols (ICNP),*, Oct 2014, pp. 445–455.

[32] CrossbowTechnology, “MICAz datasheet, MICAz wireless measurement system.” [Online]. Available: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[33] T. N. Weerasinghe, A. Kanyeshuli, and F. Y. Li, “Evaluation and improvement of collective flooding in WSNs with various link correlations,” in *PIMRC’15, submitted on 15h May 2015*.

[34] S. Olariu and I. Stojmenovic, “Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting,” in *Proc. INFOCOM*, April 2006, pp. 1–12.

[35] Q. Zhao and Y. Nakamoto, “Routing algorithms for preventing energy holes and improving fault tolerance in wireless sensor networks,” in *Second International Symposium on, Computing and Networking (CANDAR)*, Dec 2014, pp. 278–283.

[36] “tmote sky ultra low power IEEE 802.15.4 compliant wireless sensor module.” [Online]. Available: http://www.tmcnet.com/topics/articles/212896-wireless-sensor-networks-grow-2b-2021-report.htm.

[37] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. [Online]. Available: https://books.google.fr/books?id=a3A_uAAACAAJ&hl=en

[38] K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis, “The B-factor: Measuring Wireless Link Burstiness,” in *Proc.(SenSys)*, November 2008.

Appendix A

MATLAB Programs

Four Transmission schemes of AS, BS, CF and NLCT were implemented in MATLAB for the simulation purposes. Each program consists of a main program and a collection of other functions. Due to the large code length, only AS scheme is presented in this appendix.

A.1 The main Program for AS

In the following, the MATLAB codes of AS scheme's main program is presented.

```
clear all
close all
clc ;
Total_Tranmsissions=0;Total_Energy =0;a_Total_energy = 0;b_Total_energy =  0;c_Total_energy = 0;
d_Total_energy = 0;e_Total_energy = 0;f_Total_energy = 0;c_Tx_cnt=0;f_Tx_cnt=0;TX_results=[];
delivery_results=[];Energy_results=[];a_Total_Transmissions=0;b_Total_Transmissions=0;d_Total_Transmissions=0;
e_Total_Transmissions=0;Total_Delivery_to_f =0;Total_Delivery_to_c =0;

%Total simulation rounds defined by n
n=10;
% Promot to enter a bitmap combination
prompt=(’enter a stored combination i ’ );
i=input(prompt)

% Main simulation loop.
for(m=1:1:n)
    % Defining global variables
global C;global a;global b;global c;global d;global e;global f;global alpha;global Sender;
global k;global a_Rx;global b_Rx;global c_Rx;global d_Rx; global e_Rx;global f_Rx;global Qab
global Qae;global Qba; global Qbe;global Qbd;global Qbc;global Qde; global Qdb;global Qdc;global Qdf
```

```
global Qcb;global Qcd;global Qfd;global Qea;global Qeb;global Qed; global ChQab; global ChQae
global ChQba;global ChQbe;global ChQbd;global ChQbc; global ChQde;global ChQdb;global ChQdc
global ChQdf;global ChQcb;global ChQcd;global ChQfd; global ChQea;global ChQeb;global ChQed;
global qab;global qae;global qba;global qbe;global qbd;global qbc;global qea; global qeb;
global qed;global qdb;global qdc;global qde;global qdf;global qcb;global qcd; global qfd;
global ab;global ae;global ba; global be;global bd;global bc;global de;global db;global dc;
global df;global ea;global eb;global ed;global cb;global cd;global fd;global z;global CP_ab;
global CP_ae;global CP_be;global CP_bd;global CP_bc;global CP_de;global CP_db;global CP_dc;
global CP_df;global CP_eb;global CP_ed;global a_Tx_cnt;global a_Rx_cnt;global b_Tx_cnt;global b_Rx_cnt;
global c_Rx_cnt;global d_Tx_cnt;global d_Rx_cnt;global e_Tx_cnt;global e_Rx_cnt;global f_Rx_cnt;
global b_miss;global c_miss;global d_miss;global e_miss;global f_miss;global t_Boff_a;
global t_Boff_b;global t_Boff_e;global t_Boff_d;global a_Sendermode_for_b;global a_Sendermode_for_e;
global b_Sendermode_for_e;global b_Sendermode_for_d;global b_Sendermode_for_c;global e_Sendermode_for_b;
global e_Sendermode_for_d;global d_Sendermode_for_b;global d_Sendermode_for_c;global d_Sendermode_for_f;
global d_Sendermode_for_e;global winner;global a_Sendermode;global b_Sendermode ;global d_Sendermode;
global e_Sendermode;global ab_bm;global ae_bm;global ba_bm;global be_bm;global bd_bm;global bc_bm;
global  ea_bm;global  eb_bm;global  ed_bm;global db_bm;global de_bm;global dc_bm;global df_bm;
global cb_bm;global cd_bm;global fd_bm;
global sendTo_b;global sendTo_e;global sendTo_c;global sendTo_d;global sendTo_f;


%intialization variables that run inside the main loop.
a_Rx=[]; b_Rx=[]; c_Rx=[]; d_Rx=[]; e_Rx=[]; f_Rx=[];

b_miss =[]; c_miss=[]; d_miss=[]; e_miss=[]; f_miss=[];
Total_Tx_cnt=0; deliverd_to_f = 0; deliverd_to_c = 0;

a_Total_energy_J = 0; b_Total_energy_J = 0; c_Total_energy_J = 0;
d_Total_energy_J = 0; e_Total_energy_J =0; f_Total_energy_J = 0; Total_energy_J = 0;

C=0.5; packets=[1:16]; alpha=0.75 ; a_Tx_cnt =0; a_Rx_cnt = 0; b_Tx_cnt =0;b_Rx_cnt =0;
c_Rx_cnt=0;d_Tx_cnt=0;d_Rx_cnt=0;e_Tx_cnt=0;e_Rx_cnt=0;f_Rx_cnt=0;

a_HM_Rx_Cnt=0;b_HM_Rx_Cnt=0;c_HM_Rx_Cnt=0;d_HM_Rx_Cnt=0;e_HM_Rx_Cnt=0;f_HM_Rx_Cnt=0;

z=0;

% Storing bitmap combinations. Bitmap combination for scenario 1 is shown.
% Apply other scenarios similarly
    if (i==1)
        ab_bm=[1,1,1,1,1,1,1,1,1];
        ae_bm=[1,1,1,0,1,1,0,1,1];
        ba_bm=[1,1,1,1,1,1,1,1,1];
        be_bm=[1,1,1,1,1,1,1,1,1];
        bd_bm=[1,1,1,1,1,1,1,1,1];
        bc_bm=[1,1,1,0,1,1,1,0,1];
        ea_bm=[1,1,1,1,1,1,1,1,1];
        eb_bm=[1,1,1,1,1,1,0,1,1];
        ed_bm=[1,1,1,1,1,1,1,1,1];
        db_bm=[1,1,1,0,1,1,1,1,1];
        de_bm=[1,1,1,1,1,1,1,1,1];
        dc_bm=[1,1,1,1,1,1,1,1,1];
        df_bm=[1,1,1,1,1,1,1,1,1];
        cb_bm=[1,1,0,1,1,1,0,1,1];
```

```
        cd_bm=[1,1,1,1,1,1,1,1,1];
        fd_bm=[1,1,1,1,1,1,1,1,1];
    else
    end

    % Determining the number of Hello messages  received by each node
    a_HM_Rx_Cnt= sum(ba_bm(5:9))+ sum (ea_bm(5:9));
    b_HM_Rx_Cnt= sum(ab_bm(5:9))+ sum(eb_bm(5:9))+sum(db_bm(5:9))+sum(cb_bm(5:9));
    d_HM_Rx_Cnt= sum(bd_bm(5:9))+ sum(cd_bm(5:9))+sum(ed_bm(5:9)) +sum(fd_bm(5:9));
    e_HM_Rx_Cnt= sum(ae_bm(5:9))+ sum(be_bm(5:9))+ sum(de_bm(5:9));
    c_HM_Rx_Cnt= sum(bc_bm(5:9))+sum( dc_bm(5:9));
    f_HM_Rx_Cnt= sum(df_bm(5:9));



    %% For 16 data packets k is the sequence number.
    for k = 1.0: 1: 16
        % Assign CP to zero before each transmission.
        CP_ab =0; CP_ae=0; CP_be=0; CP_bd=0; CP_bc=0; CP_de=0;
        CP_db=0;  CP_df=0; CP_dc =0;  CP_eb=0;
        CP_ed=0;

        % Set-backoff times to infinite.
 t_Boff_a =inf;  t_Boff_b= inf;  t_Boff_e=inf;  t_Boff_d= inf;
        % Initializing sender mode parameters.
a_Sendermode_for_b=0; a_Sendermode_for_e=0; b_Sendermode_for_e=0;b_Sendermode_for_c=0;b_Sendermode_for_d=0;

 e_Sendermode_for_b=0; e_Sendermode_for_d=0; d_Sendermode_for_b=0;  d_Sendermode_for_c=0;

d_Sendermode_for_e=0 ; d_Sendermode_for_f=0; a_Sendermode=0;

b_Sendermode=0; e_Sendermode =0; d_Sendermode=0;

 %% Sliding window bitmap enqueing and dequeing depending wether,
 % the sender knows its previos transmissions result or not.


  % The first four data transmissions.
        if (k<=4) % initial 4 bitmap

            k
            % Bitmap establishment
            ab =[ab_bm(1), ab_bm(2), ab_bm(3),ab_bm(4)];
            ae= [ae_bm(1), ae_bm(2), ae_bm(3),ae_bm(4)];
            ba= [ba_bm(1), ba_bm(2), ba_bm(3),ba_bm(4)];
            be= [be_bm(1), be_bm(2), be_bm(3),be_bm(4)];
            bd= [bd_bm(1), bd_bm(2), bd_bm(3),bd_bm(4)];
            bc= [bc_bm(1), bc_bm(2), bc_bm(3),bc_bm(4)];
            ea= [ea_bm(1), ea_bm(2), ea_bm(3),ea_bm(4)];
            eb= [eb_bm(1), eb_bm(2), eb_bm(3),eb_bm(4)];
            ed= [ed_bm(1), ed_bm(2), ed_bm(3),ed_bm(4)];
            db= [db_bm(1), db_bm(2), db_bm(3),db_bm(4)];
            de= [de_bm(1), de_bm(2), de_bm(3),de_bm(4)];
            dc= [dc_bm(1), dc_bm(2), dc_bm(3),dc_bm(4)];
```

```
            df= [df_bm(1), df_bm(2), df_bm(3),df_bm(4)];
            cb= [cb_bm(1), cb_bm(2), cb_bm(3),cb_bm(4)];
            cd= [cd_bm(1), cd_bm(2), cd_bm(3),cd_bm(4)];
            fd= [fd_bm(1), fd_bm(2), fd_bm(3),fd_bm(4)];

            %Assigning bitmaps for link Qaultiy calculation process
            qab = [ ab_bm(1), ab_bm(2), ab_bm(3),ab_bm(4)];
            qae= [ae_bm(1), ae_bm(2), ae_bm(3),ae_bm(4)];
            qba= [ba_bm(1), ba_bm(2), ba_bm(3),ba_bm(4)];
            qbe= [be_bm(1), be_bm(2), be_bm(3),be_bm(4)];
            qbc=[bc_bm(1), bc_bm(2), bc_bm(3),bc_bm(4)];
            qbd=[bd_bm(1), bd_bm(2), bd_bm(3),bd_bm(4)];
            qea=[ea_bm(1), ea_bm(2), ea_bm(3),ea_bm(4)];
            qeb=[eb_bm(1), eb_bm(2), eb_bm(3),eb_bm(4)];
            qed=[ed_bm(1), ed_bm(2), ed_bm(3),ed_bm(4)];
            qdb=[db_bm(1), db_bm(2), db_bm(3),db_bm(4)];
            qde=[de_bm(1), de_bm(2), de_bm(3),de_bm(4)];
            qdc=[dc_bm(1), dc_bm(2), dc_bm(3),dc_bm(4)];
            qdf=[df_bm(1), df_bm(2), df_bm(3),df_bm(4)];
            qcd=[cd_bm(1), cd_bm(2), cd_bm(3),cd_bm(4)];
            qcb=[cb_bm(1), cb_bm(2), cb_bm(3),cb_bm(4)];
            qfd=[fd_bm(1), fd_bm(2), fd_bm(3),fd_bm(4)];

            % for data transmission between sequence number 4 and 8

        elseif (4< k && k <=8)
            k
            qab =[ ab_bm(2), ab_bm(3), ab_bm(4),ab_bm(5)];
            qae=[ae_bm(2), ae_bm(3), ae_bm(4),ae_bm(5)];
            qba= [ba_bm(2), ba_bm(3), ba_bm(4),ba_bm(5)];
            qbe= [be_bm(2), be_bm(3), be_bm(4),be_bm(5)];
            qbc=[bc_bm(2), bc_bm(3), bc_bm(4),bc_bm(5)];
            qbd=[bd_bm(2), bd_bm(3), bd_bm(4),bd_bm(5)];
            qea=[ea_bm(2), ea_bm(3), ea_bm(4),ea_bm(5)];
            qeb=[eb_bm(2), eb_bm(3), eb_bm(4),eb_bm(5)];
            qed=[ed_bm(2), ed_bm(3), ed_bm(4),ed_bm(5)];
            qdb=[db_bm(2), db_bm(3), db_bm(4),db_bm(5)];
            qde=[de_bm(2), de_bm(3), de_bm(4),de_bm(5)];
            qdc=[dc_bm(2), dc_bm(3), dc_bm(4),dc_bm(5)];
            qdf=[df_bm(2), df_bm(3), df_bm(4),df_bm(5)];
            qcd=[cd_bm(2), cd_bm(3), cd_bm(4),cd_bm(5)];
            qcb=[cb_bm(2), cb_bm(3), cb_bm(4),cb_bm(5)];
            qfd=[fd_bm(2), fd_bm(3), fd_bm(4),fd_bm(5)];

            if(k==5)
                k
                if ((ba_bm(6)==1))
 % checks if A has received the result of its previos Hello messege TX.
                    ab = [ab(2), ab(3), ab(4),ab_bm(5)]  ;
 % If received, add the result of the txmitted Hello to the fourth spot of existing ab. shift by 1,Drop the ab(1)
                elseif ((ba_bm(6)==0))
 % If A hasn't received the result of previous Helo messege Tx, ab remains the existing array.
                    ab  ;
                end
```

```
if ((ea_bm(6)==1))
    ae= [ae(2), ae(3), ae(4),ae_bm(5)];
elseif ((ea_bm(6)==0))
    ae;
end

if ((ab_bm(6)==1))
    ba= [ba(2), ba(3), ba(4),ba_bm(5)];
elseif ((ab_bm(6)==0))
    ba    ;
end

if ((eb_bm(6)==1))
    be= [be(2), be(3), be(4),be_bm(5)];
elseif ((eb_bm(6)==0))
    be    ;
end

if ((db_bm(6)==1))
    bd=[bd(2), bd(3), bd(4),bd_bm(5)];
elseif ((db_bm(6)==0))
    bd   ;
end

if ((cb_bm(6)==1))
    bc=[bc(2), bc(3), bc(4),bc_bm(5)];
elseif ((cb_bm(6)==0))
    bc  ;
end


if ((ae_bm(6)==1))
    ea=[ea(2), ea(3), ea(4),ea_bm(5)];
elseif ((ae_bm(6)==0))
    ea  ;
end

if ((be_bm(6)==1))
    eb=[eb(2), eb(3), eb(4),eb_bm(5)];
elseif ((be_bm(6)==0))
    eb  ;
end

if ((de_bm(6)==1))
    ed=[ed(2), ed(3), ed(4),ed_bm(5)];
elseif ((de_bm(6)==0))
    ed  ;
end


if ((bd_bm(6)==1))
    db=[db(2), db(3), db(4),db_bm(5)];
elseif ((bd_bm(6)==0))
```

```
                db  ;
            end

            if ((ed_bm(6)==1))
                de=[de(2), de(3), de(4),de_bm(5)];
            elseif ((ed_bm(6)==0))
                de  ;
            end

            if ((cd_bm(6)==1))
                dc=[dc(2), dc(3), dc(4),dc_bm(5)] ;
            elseif ((cd_bm(6)==0))
                dc ;
            end

            if ((fd_bm(6)==1))
                df=[df(2), df(3), df(4),df_bm(5)] ;
            elseif ((fd_bm(6)==0))
                df ;
            end

            if ((dc_bm(6)==1))
                cd=[cd(2), cd(3), cd(4),cd_bm(5)]  ;
            elseif ((dc_bm(6)==0))
                cd ;
            end

            if ((bc_bm(6)==1))
                cb=[cb(2), cb(3), cb(4),cb_bm(5)];
            elseif ((bc_bm(6)==0))
                cb ;
            end

            if ((df_bm(6)==1))
                fd=[fd(2), fd(3), fd(4),fd_bm(5)]  ;
            elseif ((df_bm(6)==0))
                fd ;
            end
        elseif ( 5< k && k <=8)
            k
            ab ; ae; ba; be;bd; bc; ea;eb;ed;db; de; dc;df;cb;cd; fd;

        end

        % for data transmission between sequence number 8 and 12

    elseif (8< k && k <=12)
        k
        qab = [ ab_bm(3), ab_bm(4), ab_bm(5),ab_bm(6)];
        qae= [ae_bm(3), ae_bm(4), ae_bm(5),ae_bm(6)];
        qba= [ba_bm(3), ba_bm(4), ba_bm(5),ba_bm(6)];
        qbe= [be_bm(3), be_bm(4), be_bm(5),be_bm(6)];
        qbc=[bc_bm(3), bc_bm(4), bc_bm(5),bc_bm(6)];
        qbd=[bd_bm(3), bd_bm(4), bd_bm(5),bd_bm(6)];
```

```
qea=[ea_bm(3), ea_bm(4), ea_bm(5),ea_bm(6)];
qeb=[eb_bm(3), eb_bm(4), eb_bm(5),eb_bm(6)];
qed=[ed_bm(3), ed_bm(4), ed_bm(5),ed_bm(6)];
qdb=[db_bm(3), db_bm(4), db_bm(5),db_bm(6)];
qde=[de_bm(3), de_bm(4), de_bm(5),de_bm(6)];
qdc=[dc_bm(3), dc_bm(4), dc_bm(5),dc_bm(6)];
qdf=[df_bm(3), df_bm(4), df_bm(5),df_bm(6)];
qcd=[cd_bm(3), cd_bm(4), cd_bm(5),cd_bm(6)];
qcb=[cb_bm(3), cb_bm(4), cb_bm(5),cb_bm(6)];
qfd=[fd_bm(3), fd_bm(4), fd_bm(5),fd_bm(6)];



if(k==9)

    if ((ba_bm(7)==1))


        k;
        ab_bm(6);
        ab= [ab(2), ab(3), ab(4),ab_bm(6)] ;


    elseif ((ba_bm(7)==0))
        ab

    end

    if ((ea_bm(7)==1))
        ae= [ae(2), ae(3), ae(4),ae_bm(6)];
    elseif ((ea_bm(7)==0))
        ae;
    end

    if ((ab_bm(7)==1))
        ba= [ba(2), ba(3), ba(4),ba_bm(6)];
    elseif ((ab_bm(6)==0))
        ba    ;
    end

    if ((eb_bm(7)==1))
        be= [be(2), be(3), be(4),be_bm(6)];
    elseif ((eb_bm(7)==0))
        be    ;
    end

    if ((db_bm(7)==1))
        bd=[bd(2), bd(3), bd(4),bd_bm(6)];
    elseif ((db_bm(7)==0))
        bd   ;
    end

    if ((cb_bm(7)==1))
        bc=[bc(2), bc(3), bc(4),bc_bm(6)];
```

```
elseif ((cb_bm(7)==0))
    bc  ;
end


if ((ae_bm(7)==1))
    ea=[ea(2), ea(3), ea(4),ea_bm(6)];
elseif ((ae_bm(7)==0))
    ea  ;
end

if ((be_bm(7)==1))
    eb=[eb(2), eb(3), eb(4),eb_bm(6)];
elseif ((be_bm(7)==0))
    eb  ;
end

if ((de_bm(7)==1))
    ed=[ed(2), ed(3), ed(4),ed_bm(6)];
elseif ((de_bm(7)==0))
    ed  ;
end


if ((bd_bm(7)==1))
    db=[db(2), db(3), db(4),db_bm(6)];
elseif ((bd_bm(7)==0))
    db  ;
end

if ((ed_bm(7)==1))
    de=[de(2), de(3), de(4),de_bm(6)];
elseif ((ed_bm(7)==0))
    de  ;
end

if ((cd_bm(7)==1))
    dc=[dc(2), dc(3), dc(4),dc_bm(6)];
elseif ((cd_bm(7)==0))
    dc ;
end

if ((fd_bm(7)==1))
    df=[df(2), df(3), df(4),df_bm(6)]  ;
elseif ((fd_bm(7)==0))
    df ;
end

if ((dc_bm(7)==1))
    cd=[cd(2), cd(3), cd(4),cd_bm(6)]  ;
elseif ((dc_bm(7)==0))
    cd ;
end
```

```
            if ((cb_bm(7)==1))
                cb=[cb(2), cb(3), cb(4),cb_bm(6)] ;
            elseif ((cb_bm(7)==0))
                cb ;
            end

            if ((fd_bm(7)==1))
                fd=[fd(2), fd(3), fd(4),fd_bm(6)]  ;
            elseif ((fd_bm(7)==0))
                fd ;
            end


        elseif(9<k && k<=12)
            k;
            ab ; ae; ba; be;bd; bc; ea;eb;ed;db; de; dc;df;cb;cd; fd;

        end

        % for data transmission between sequence number 12 and 16
    elseif(12<k && k<=16)


        qab = [ab_bm(4), ab_bm(5), ab_bm(6),ab_bm(7)];
        qae= [ae_bm(4), ae_bm(5), ae_bm(6),ae_bm(7)];
        qba= [ba_bm(4), ba_bm(5), ba_bm(6),ba_bm(7)];
        qbe= [be_bm(4), be_bm(5), be_bm(6),be_bm(7)];
        qbc=[bc_bm(4), bc_bm(5), bc_bm(6),bc_bm(7)];
        qbd=[bd_bm(4), bd_bm(5), bd_bm(6),bd_bm(7)];
        qea=[ea_bm(4), ea_bm(5), ea_bm(6),ea_bm(7)]
        qeb=[eb_bm(4), eb_bm(5), eb_bm(6),eb_bm(7)];
        qed=[ed_bm(4), ed_bm(5), ed_bm(6),ed_bm(7)];
        qdb=[db_bm(4), db_bm(5), db_bm(6),db_bm(7)];
        qde=[de_bm(4), de_bm(5), de_bm(6),de_bm(7)];
        qdc=[dc_bm(4), dc_bm(5), dc_bm(6),dc_bm(7)];
        qdf=[df_bm(4), df_bm(5), df_bm(6),df_bm(7)];
        qcd=[cd_bm(4), cd_bm(5), cd_bm(6),cd_bm(7)];
        qcb=[cb_bm(4), cb_bm(5), cb_bm(6),cb_bm(7)];
        qfd=[fd_bm(4), fd_bm(5), fd_bm(6),fd_bm(7)];

        if(k==13)
            k;
            cd;
            if ((ba_bm(8)==1))
                ab = [ab(2), ab(3), ab(4),ab_bm(7)] ;
                cd;

            elseif ((ba_bm(8)==0))
                ab    ;
            end

            if ((ea_bm(8)==1))
                ae= [ae(2), ae(3), ae(4),ae_bm(7)];
            elseif ((ea_bm(8)==0))
```

```
    ae;
end

if ((ab_bm(8)==1))
    ba= [ba(2), ba(3), ba(4),ba_bm(7)];
elseif ((ab_bm(8)==0))
    ba
end

if ((eb_bm(8)==1))
    be= [be(2), be(3), be(4),be_bm(7)];
elseif ((eb_bm(8)==0))
    be    ;
end

if ((db_bm(8)==1))
    bd=[bd(2), bd(3), bd(4),bd_bm(7)];
elseif ((db_bm(8)==0))
    bd   ;
end

if ((cb_bm(8)==1))
    bc=[bc(2), bc(3), bc(4),bc_bm(7)];
elseif ((cb_bm(8)==0))
    bc  ;
end


if ((ae_bm(8)==1))
    ea=[ea(2), ea(3), ea(4),ea_bm(7)];
elseif ((ae_bm(8)==0))
    ea  ;
end

if ((be_bm(8)==1))
    eb=[eb(2), eb(3), eb(4),eb_bm(7)];
elseif ((be_bm(8)==0))
    eb  ;
end

if ((de_bm(8)==1))
    ed=[ed(2), ed(3), ed(4),ed_bm(7)];
elseif ((de_bm(8)==0))
    ed  ;
end


if ((bd_bm(8)==1))
    db=[db(2), db(3), db(4),db_bm(7)];
elseif ((bd_bm(8)==0))
    db  ;
end

if ((ed_bm(8)==1))
```

```
                de=[de(2), de(3), de(4),de_bm(7)];
            elseif ((ed_bm(8)==0))
                de  ;
            end

            if ((cd_bm(8)==1))
                dc=[dc(2), dc(3), dc(4),dc_bm(7)]  ;
            elseif ((cd_bm(8)==0))
                dc ;
            end

            if ((fd_bm(8)==1))
                df=[df(2), df(3), df(4),df_bm(7)]  ;
            elseif ((fd_bm(8)==0))
                df ;
            end

            if ((dc_bm(8)==1))
                cd=[cd(2), cd(3), cd(4),cd_bm(7)] ;
            elseif ((dc_bm(8)==0))
                cd ;
            end

            if ((bc_bm(8)==1))
                cb=[cb(2), cb(3), cb(4),cb_bm(7)]  ;
            elseif ((bc_bm(8)==0))
                cb ;
            end

            if ((df_bm(8)==1))
                fd=[fd(2), fd(3), fd(4),fd_bm(7)]  ;
            elseif ((df_bm(8)==0))
                fd ;
            end

        elseif(13<k && k<=16)
            k;
           ab ; ae; ba; be;bd; bc; ea;eb;ed;db; de; dc;df;cb;cd; fd;

        else
        end
    end
%% Calculating Quality of the links as experienced by nodes using the bitmaps they use
    qual(ab);
    Qab=qual(ab);

    qual(ae);
    Qae=qual(ae);

    qual(ba);
    Qba=qual(ba);

    qual(be);
    Qbe=qual(be);
```

```
        qual(bc);
        Qbc=qual(bc);

        qual(bd);
        Qbd=qual(bd);

        qual(ea);
        Qea=qual(ea);

        qual(eb);
        Qeb=qual(eb);

        qual(ed);
        Qed=qual(ed);

        qual(db);
        Qdb=qual(db);

        qual(de);
        Qde=qual(de);

        qual(dc);
        Qdc=qual(dc);

        qual(df);
        Qdf=qual(df);

        qual(cb);
        Qcb=qual(cb);
        qual(cd);
        Qcd=qual(cd);
        qual(fd);
        Qfd=qual(fd);




%% Real Channellink quality values using the actual bitmaps.
%This is different from the quality perceived by the nodes.

        qual(qab);  % Channel Quality
        ChQab=qual(qab);

        qual(qae);
        ChQae=qual(qae);

        qual(qba);
        ChQba=qual(qba);

        qual(qbe);
        ChQbe=qual(qbe);

        qual(qbc);
```

```
ChQbc=qual(qbc);

qual(qbd);
ChQbd=qual(qbd);

qual(qea);
ChQea=qual(qea);

qual(qeb);
ChQeb=qual(qeb);

qual(qed);
ChQed=qual(qed);

qual(qdb);
ChQdb=qual(qdb);

qual(qdc);
ChQdc=qual(qdc);

qual(qde);
ChQde=qual(qde);
qual(qdf);
ChQdf=qual(qdf);
qual(qcd);
ChQcd=qual(qcd);
qual(qcb);
ChQcb=qual(qcb);
qual(qfd);
ChQfd=qual(qfd);


% Setting SR phase periodically after 4 transmissions.

sender(k);
if(k==4 || k==8 || k==12|| k==16)
    recovery();
else
end
% Assinging   ER phase to start after all the transmissions
if(k==16)
    missing();
    f_miss;
    d_miss;
    z=10
    while(not(isempty(f_miss))|| not(isempty(c_miss)))



        if(isempty(b_miss))
            b_miss;
        else
            if(Qba> rand())
                ba_bm(z)=1;
```

```
        else
            ba_bm(z)=0;
        end

        if(Qbe> rand())
            be_bm(z)=1;
        else
            be_bm(z)=0;
        end
        if(Qbd> rand())
            bd_bm(z)=1;
        else
            bd_bm(z)=0;
        end
        if(Qbc> rand())
            bc_bm(z)=1;
        else
            bc_bm(z)=0;
        end
    end

    if(isempty(e_miss))
        e_miss;
    else
        if(Qea> rand())
            ea_bm(z)=1;
        else
            ea_bm(z)=0;
        end

        if(Qeb> rand())
            eb_bm(z)=1;
        else
            eb_bm(z)=0;
        end
        if(Qed> rand())
            ed_bm(z)=1;
        else
            ed_bm(z)=0;
        end
    end


    if(isempty(d_miss))
        d_miss;
    else
        if(Qde> rand())
            de_bm(z)=1;
        else
            de_bm(z)=0;
        end

        if(Qdb> rand())
            db_bm(z)=1;
```

```
                else
                    db_bm(z)=0;
                end
                if(Qdc> rand())
                    dc_bm(z)=1;
                else
                    dc_bm(z)=0;
                end
                if(Qdf> rand())
                    df_bm(z)=1;
                else
                    df_bm(z)=0;
                end
            end
            if(isempty(c_miss)/4)
                c_miss;
            else
                if(Qcd> rand())
                    cd_bm(z)=1;
                else
                    cd_bm(z)=0;
                end

                if(Qcb> rand())
                    cb_bm(z)=1;
                else
                    cb_bm(z)=0;
                end
            end
            if(isempty(f_miss))
                f_miss;
            else
                if(Qfd> rand())
                    fd_bm(z)=1;
                else
                    fd_bm(z)=0;
                end
            end

            % Calling data re-transmission used in SR
            recovery2();
            % Checking for missing packets
            missing();
            z=z+1
        end
    else

    end
end


TX_results(m,1)= a_Tx_cnt;
TX_results(m,2)= b_Tx_cnt;
TX_results(m,3)= c_Tx_cnt;
```

```
TX_results(m,4)= d_Tx_cnt;
TX_results(m,5)= e_Tx_cnt;
TX_results(m,6)= f_Tx_cnt;

% Calculating possible receptions at each node.
% From not received packets, 50% packets due to CRC errors after
% receving 50% completely not heard.
a_possible_receptions = b_Tx_cnt + e_Tx_cnt ;
b_possible_receptions = a_Tx_cnt + e_Tx_cnt + d_Tx_cnt;
e_possible_receptions = a_Tx_cnt + b_Tx_cnt + d_Tx_cnt;
d_possible_receptions = b_Tx_cnt + e_Tx_cnt ;
c_possible_receptions = b_Tx_cnt + d_Tx_cnt;
f_possible_receptions = d_Tx_cnt;

a_Rx_cnt=a_Rx_cnt+ceil((a_possible_receptions-a_Rx_cnt)/2);
b_Rx_cnt=b_Rx_cnt+ceil((b_possible_receptions-b_Rx_cnt)/2);
c_Rx_cnt=c_Rx_cnt+ceil((c_possible_receptions-c_Rx_cnt)/2);
d_Rx_cnt=d_Rx_cnt+ceil((d_possible_receptions-d_Rx_cnt)/2);
e_Rx_cnt=e_Rx_cnt+ceil((e_possible_receptions-e_Rx_cnt)/2);
f_Rx_cnt=f_Rx_cnt+ceil((f_possible_receptions-f_Rx_cnt)/2);

%Energy calculation using MICAz

P_Tx= 17.4*(1/1000) * 3 ;
P_Rx=19.7 * (1/1000)*3 ;

data_Sz = 100 ;
data_rate = 250 ;

T_data = (data_Sz*8)/(data_rate*1000);
Hello_Sz=50 ;
T_Hello= (Hello_Sz)*8/(data_rate *1000) ;
P_Sleep = 3/(1000000) ;

% Data energy calculations

a_data_energy = P_Tx *T_data* a_Tx_cnt + P_Rx *T_data* a_Rx_cnt  ;
b_data_energy=P_Tx *T_data* b_Tx_cnt + P_Rx *T_data* b_Rx_cnt  ;
d_data_energy=P_Tx *T_data* d_Tx_cnt + P_Rx *T_data* d_Rx_cnt ;
e_data_energy=P_Tx *T_data* e_Tx_cnt + P_Rx *T_data* e_Rx_cnt ;
c_data_energy=P_Rx *T_data* c_Rx_cnt ;
f_data_energy=P_Rx *T_data* f_Rx_cnt ;


%% Hello energy calculations
a_H_energy = P_Tx *T_Hello*9 + P_Rx *T_Hello* a_HM_Rx_Cnt ;
b_H_energy=P_Tx *T_Hello*9 + P_Rx *T_Hello*  b_HM_Rx_Cnt ;
d_H_energy=P_Tx *T_Hello* 9 + P_Rx *T_Hello* c_HM_Rx_Cnt ;
e_H_energy=P_Tx *T_Hello* 9 + P_Rx *T_Hello* e_HM_Rx_Cnt ;
c_H_energy=P_Tx *T_Hello* 9 + P_Rx *T_Hello* c_HM_Rx_Cnt;
f_H_energy=P_Tx *T_Hello* 9 + P_Rx *T_Hello* f_HM_Rx_Cnt;
```

```
    Total_Tx_cnt= a_Tx_cnt+b_Tx_cnt+d_Tx_cnt+e_Tx_cnt ;

    % Delivery precentage calculations,
    deliverd_to_f = (16-numel(f_miss))/16
    deliverd_to_c = (16-numel(c_miss))/16

    delivery_results(m,1)=deliverd_to_f;
    delivery_results(m,2)=deliverd_to_c;


    % Total energy per node calculations.

    a_Total_energy_J = a_data_energy + a_H_energy
    b_Total_energy_J = b_data_energy+ b_H_energy
    c_Total_energy_J = c_data_energy + c_H_energy
    d_Total_energy_J = d_data_energy + d_H_energy
    e_Total_energy_J = e_data_energy + e_H_energy
    f_Total_energy_J = f_data_energy + f_H_energy


    Energy_results(m,1)=a_Total_energy_J;
    Energy_results(m,2)=b_Total_energy_J;
    Energy_results(m,3)=c_Total_energy_J;
    Energy_results(m,4)=d_Total_energy_J;
    Energy_results(m,5)=e_Total_energy_J;
    Energy_results(m,6)=f_Total_energy_J;


    Total_energy_J = (a_Total_energy_J+ b_Total_energy_J+ c_Total_energy_J+ d_Total_energy_J+ e_Total_energy_J+f_To



end

%Calculating mean and SD values for TX, Energy and Delivery

mean_Total_Tranmsissions= mean(TX_results);
std_Tx=std(TX_results);

mean_Total_energy=mean(Energy_results);
std_Eg= std(Energy_results);

mean_Total_delivery= mean(delivery_results);
std_del= std(delivery_results);

%Plotting figures

figure
grid on
hold on
mylabels1={'','','','C','','','','F'};
bar(mean_Total_delivery*100,'m')
errorbar(mean_Total_delivery*100,std_del*100,'.')
xlabel('Node')
```

```
ylabel('Packet delivery percentage')
title('Packet delivery for leaf nodes')
set(gca, 'XTickLabelMode' , 'manual','XTickLabel', mylabels1)


figure
set(gca,'XGrid','off','YGrid','on');
hold on
mylabels={'' ,'A' ,'B' ,'C','D','E','F'};
bar(mean_Total_energy*1000,'m')
errorbar(mean_Total_energy*1000,std_Eg*1000,'.')
xlabel('Node')
ylabel('Average energy consumption(in mJ)')
%title('Average energy consumption per node')
set(gca, 'XTickLabelMode' , 'manual','XTickLabel', mylabels)

figure
set(gca,'XGrid','off','YGrid','on');
hold on
mylabels={'' ,'A' ,'B' ,'C','D','E','F'};
bar(mean_Total_Tranmsissions,'m')
errorbar(mean_Total_Tranmsissions,std_Tx,'.')
xlabel('Node')
ylabel('Average transmissions')
% title('Average transmissions per node')
set(gca, 'XTickLabelMode' , 'manual','XTickLabel', mylabels)
```

A.2 Other Functions of AS

Node Sender mode Functions:

Only the sender function for node D is presented here to save space, other nodes also have similar sender functions.

```
function t_boff_d=dSend()

global alpha; global C ; global Qde ; global Qdb; global Qdc ; global Qdf ;global CP_de ; global CP_db ;
global CP_dc ;global CP_df ;global d_Sendermode_for_b ;global d_Sendermode_for_c ; global d_Sendermode_for_f ;
global d_Sendermode_for_e ;global d_Sendermode; d_UN=0;

%CP calvulation using senderCP function
CP_de=SenderCP(CP_de,Qde)
CP_db=SenderCP(CP_db,Qdb)
CP_dc=SenderCP(CP_dc,Qdc)
CP_df=SenderCP(CP_df,Qdf)

% Checking for uncovered niehgbors
if CP_de<alpha
```

```
    d_Sendermode_for_e=1 ;
    d_Sendermode=1;
elseif CP_de>=alpha
    d_Sendermode_for_e=0;
end

if CP_dc<alpha
    d_Sendermode_for_c=1;
    d_Sendermode=1;
elseif CP_dc>=alpha
    d_Sendermode_for_c=0;
end

if CP_db<alpha
    d_Sendermode=1;
    d_Sendermode_for_b=1;

elseif CP_db>=alpha
    d_Sendermode_for_b=0;
end

if CP_df<alpha
    d_Sendermode=1
    d_Sendermode_for_f=1;

elseif CP_df>=alpha

    d_Sendermode_for_f=0;
end

%Updating uncoverd neighbor count
if ( d_Sendermode_for_e==1)

    d_UN =d_UN +1;

else
end
if (d_Sendermode_for_b==1)
    d_UN =d_UN +1;

else
end
if (d_Sendermode_for_c==1)
    d_UN =d_UN +1;

else
end

if (d_Sendermode_for_f==1)
    d_UN =d_UN +1;

else
end
```

```
%Calculating Backoff
t_boff_d= C/(d_UN)  ;

end
```

Node Receiver mode Functions:

Only the Receiver function for node D is presented here to save space, other nodes also have similar Receiver functions.

```
function t_boff_d = dReceive()

global alpha ; global Sender; global C ; global Qbe ; global Qbd ; global Qbc; global Qeb; global Qed
global CP_de; global CP_db; global CP_dc;global CP_df; global be; global bd; global bc; global eb
global ed; global d_Sendermode_for_b; global d_Sendermode_for_c; global d_Sendermode_for_f
global d_Sendermode_for_e; global d_Sendermode;

d_UN=0;

% If the sender is node B
if (Sender == ’b’ )
    CP_db=1;

    if(CP_de < alpha)
%Calling for K factor functions

        kf=kfactor(bd,be);
        if (kf==1)
            if (( Qbd <=Qbe ))
                CP_de=1;
            else
                CP_de =CP_de+ Qbe;
% Node D calculates node B’s ability to cover E and adds that to its CP.
%Idea is to use the available infor as percieved by A
            end
        elseif (kf==-1)
            if ( (Qbe + Qbd) <1 )
                CP_de;

            else
                CP_de =CP_de+ Qbe;

            end
        elseif (kf==0)
            CP_de =CP_de+ Qbe;
        end

    else
    end
    if(CP_dc< alpha)
        kf=kfactor(bd,bc);
        if (kf==1)
            if (( Qbd <=Qbc ))
```

```
                CP_dc=1;
            else
                CP_dc =CP_dc+ Qbc;

            end
        elseif (kf==-1)
            if ( (Qbc + Qbd) <1 )
                CP_dc;

            else
                CP_dc =CP_dc+ Qbc;

            end
        elseif (kf==0)
            CP_dc =CP_dc+ Qbc;
        end

    else
    end

% If the sender is node E
elseif(Sender == 'e')
    CP_de=1;
    if(CP_db<alpha)
        kf=kfactor(eb,ed);

        if (kf==1)
            if (( Qed <=Qeb ))
                CP_db=1;
            else
                CP_db=CP_db+Qeb;
            end
        elseif (kf==-1)
            if ( (Qeb + Qed) <1 )
                CP_db;
            else
                CP_db=CP_db+Qeb;

            end
        elseif (kf==0)
            CP_db=CP_db+Qeb;

        end
    else
    end

end

% Cheking for uncovered neighbors

if CP_de<alpha

    d_Sendermode_for_e=1;
    d_Sendermode=1;
```

```
elseif CP_de>=alpha
    d_Sendermode_for_e=0;
end

if CP_dc<alpha
    d_Sendermode_for_c=1;
    d_Sendermode=1;
elseif CP_dc>=alpha
    d_Sendermode_for_c=0;
end

if CP_df<alpha
    d_Sendermode_for_f=1;
    d_Sendermode=1;
elseif CP_df>=alpha
    d_Sendermode_for_f=0   ;
end

if CP_db<alpha

    d_Sendermode_for_b=1;
    d_Sendermode=1;
elseif CP_db>alpha

    d_Sendermode_for_b=0;
end

% Updating uncoverd nodes
if ( d_Sendermode_for_e==1)
    d_UN =d_UN +1;

else
end
if (d_Sendermode_for_b==1)
    d_UN=d_UN+1;
else
end
if (d_Sendermode_for_c==1)
    d_UN=d_UN+1;
else
end
if (d_Sendermode_for_f==1)
    d_UN=d_UN+1;
else
end
% Calculating back-off time
t_boff_d= C/(2*(d_UN))  ;

end
```

Function for receiving state determination:

```
function option()

global alpha;global Sender;global k;global a;global b;global c;global d;global e
global f;global Qab;global Qae;global Qba;global Qbe;global Qbd;global Qbc;global Qde
global Qdb;global Qdc;global Qdf;global Qcb;global Qcd;global Qfd;global Qea;global Qeb
global Qed;global a_Rx;global b_Rx;global c_Rx;global d_Rx;global e_Rx;global f_Rx;
global ChQab;global ChQae;global ChQba;global ChQbe;global ChQbd;global ChQbc;global ChQde;
global ChQdb;global ChQdc;global ChQdf;global ChQea;global ChQeb;global ChQed
global a_Rx_cnt;global b_Rx_cnt;global c_Rx_cnt;global d_Rx_cnt;global e_Rx_cnt
global f_Rx_cnt;global t_Boff_a;global t_Boff_b;global t_Boff_e;global t_Boff_d



if ( Sender=='a')
    % determining wether the transmission is sucessfully received or not.
    if (ChQab> rand )
        % Calling receive function to update CP values and calculate Backofftimes
        t_Boff_b=bReceive();
        b_Rx_cnt=b_Rx_cnt+1
        b_Rx(end+1)=k ;
        % Updates the received packet sequence number K representes the sequence number of current packet
    else
        b_Not_Rx =1;
    end

    if( ChQae> rand )
        t_Boff_e=eReceive();
        e_Rx_cnt=e_Rx_cnt+1;
        e_Rx(end+1)=k;

    else
        e_Not_Rx=1;
    end
elseif (Sender=='b')
    %  bSend()
    if (ChQba>rand)
        t_Boff_a=aReceive();
        a_Rx_cnt=a_Rx_cnt+1 ;
        a_Rx(end+1)=k;

    else
        a_Not_Rx=1;
    end

    if (ChQbe > rand)
        t_Boff_e=eReceive();
        e_Rx_cnt=e_Rx_cnt+1;
        e_Rx(end+1)=k;
```

```
    else
        e_Not_Rx=1;
    end

    if (ChQbd > rand )
        t_Boff_d=dReceive();
        d_Rx_cnt=d_Rx_cnt+1;
        d_Rx(end+1)=k;

    else
        d_Not_Rx=1;
    end
    if (ChQbc > rand )
        % t_Boff_c=cReceive();
        c_Rx_cnt=c_Rx_cnt+1;
        c_Rx(end+1)=k;

    else
        c_Not_Rx=1;
    end

elseif ( Sender=='d')
    %dSend();
    if (ChQdb>rand)
        t_Boff_b=bReceive();
        b_Rx_cnt=b_Rx_cnt+1 ;
        b_Rx(end+1)=k;

    else
        b_Not_Rx=1;
    end

    if (ChQde > rand)
        t_Boff_e=eReceive();
        e_Rx_cnt=e_Rx_cnt+1;
        e_Rx(end+1)=k;

    else
        e_Not_Rx=1;
    end

    if (ChQdc > rand )
        %t_Boff_c=cReceive();
        c_Rx_cnt=c_Rx_cnt+1;
        c_Rx(end+1)=k;

    else
        c_Not_RX=1;
    end
    if (ChQdf > rand)
        f_Rx_cnt=f_Rx_cnt+1;
        f_Rx(end+1)=k;
```

```
    else
        f_Not_Rx=1;
    end
elseif( Sender==’e’)
    %  eSend();
    if ( ChQea> rand)
        t_Boff_a=aReceive();
        a_Rx_cnt=a_Rx_cnt+1;
        a_Rx(end+1)=k;

    else
        a_Not_Rx=1;
    end
    if (ChQeb> rand)
        t_Boff_b=bReceive();
        b_Rx_cnt=b_Rx_cnt+1;
        b_Rx(end+1)=k;

    else
        b_Not_Rx=1;
    end
    if (ChQed> rand)
        t_Boff_d=dReceive();
        d_Rx_cnt=d_Rx_cnt+1;
        d_Rx(end+1)=k;

    else
        d_Not_Rx=1;
    end
else
end

%Calling for competition function
competition();

end
```

Function to handle node competition:

```
function competition(
global alpha ;global Sender;global k;global a;global b;global c;global d;global e;global f
global Qab;global Qae;global Qba;global Qbe;global Qbd;global Qbc;global Qde;global Qdb;
global Qdc;global Qdf;global Qcb;global Qcd;global Qfd;global Qea;global Qeb;global Qed

global a_Rx;global b_Rx;global c_Rx;global d_Rx;global e_Rx;global f_Rx;

global ChQab;global ChQae;global ChQba;global ChQbe;global ChQbd;global ChQbc;global ChQde
global ChQdb;global ChQdc;global ChQdf;global ChQcb;global ChQcd;global ChQfd;global ChQea
global ChQeb;global ChQed

global qab;global qae;global qba;global qbe;global qbd;global qbc;global qea;global qeb
```

```
global qed;global qdb;global qdc;global qde;global qdf;global qcb;global qcd;global qfd

global ab;global ae;global ba;global be;global bd;global bc;global de;global db;global dc
global df;global ea;global eb;global ed;global cb;global cd;global fd

global CP_ab;global CP_ae;global CP_be;global CP_bd;global CP_bc;global CP_de;global CP_db
global CP_dc;global CP_df;global CP_eb;global CP_ed;

global a_Tx_cnt;global a_Rx_cnt;global b_Tx_cnt;global b_Rx_cnt;global c_Rx_cnt;global d_Tx_cnt
global d_Rx_cnt;global e_Tx_cnt;global e_Rx_cnt;global f_Rx_cnt;global t_Boff_a;global t_Boff_b
global t_Boff_e;global t_Boff_d;global a_Sendermode_for_b;global a_Sendermode_for_e;global b_Sendermode_for_e
global b_Sendermode_for_d;global b_Sendermode_for_c;global e_Sendermode_for_b;global e_Sendermode_for_d
global d_Sendermode_for_b;global d_Sendermode_for_c;global d_Sendermode_for_f;global d_Sendermode_for_e

global winner;global Total_Boff;global a_Sendermode;global b_Sendermode ;global d_Sendermode;
global e_Sendermode;

% Keep on competing untill all nodes exit competition.
while(a_Sendermode_for_b˜=0 |a_Sendermode_for_e˜=0 |b_Sendermode_for_e˜=0 ...
        | b_Sendermode_for_c˜=0 | b_Sendermode_for_d˜=0|d_Sendermode_for_b˜=0 ...
        | d_Sendermode_for_e˜=0 | d_Sendermode_for_c˜=0 | d_Sendermode_for_f˜=0 ...
        | e_Sendermode_for_b˜=0 | e_Sendermode_for_d˜=0  )


 winner= min(min(t_Boff_a,t_Boff_e),min(t_Boff_b,t_Boff_d))

    if winner==t_Boff_a

        a_Tx_cnt=a_Tx_cnt+1;
        Sender=’a’

        t_Boff_a=aSend();
        if(winner˜=inf)
            Total_Boff=Total_Boff+winner
        else
        end
        option();



    elseif winner==t_Boff_b

        b_Tx_cnt=b_Tx_cnt+1;
        Sender=’b’
        t_Boff_b ;

        t_Boff_b=bSend();
        if(winner˜=inf)
            Total_Boff=Total_Boff+winner
        else
        end
        option();
        % S, N2,N3 receive mode
```

```
    elseif winner==t_Boff_e

        e_Tx_cnt=e_Tx_cnt+1 ;
        Sender='e'
        t_Boff_e ;

        t_Boff_e=eSend();
        if(winner~=inf)
            Total_Boff=Total_Boff+winner
        else
        end
        option();

    elseif winner==t_Boff_d

        Sender='d'
        t_Boff_d ;
        d_Tx_cnt=d_Tx_cnt+1 ;
        t_Boff_d=dSend();
        if(winner~=inf)
            Total_Boff=Total_Boff+winner
        else
        end
        % Calling option() to check the receive status;
        option();
    else
    end


end
end
```

Functions to represent the SR and ER phases:

```
function recovery()

global alpha;global Qab;global Qae;global Qbe;global Qbd;global Qbc;global Qdc;
global Qdf;global Qeb;global Qed;global k;global sender_to_b;global sender_to_c;
global sender_to_d;global sender_to_e;global sender_to_f;global Boff_aRec;
global Boff_bRec;global Boff_eRec;global Boff_dRec;global a_sendTo_b;global a_sendTo_e;
global b_sendTo_c;global d_sendTo_c;global b_sendTo_d;global d_sendTo_f;global e_sendTo_b;
global e_sendTo_d;global b_sendTo_e;global Seq_no;global a_sendermode_b;global a_sendermode_e;
global CP_ab;global CP_ae;global CP_be;global CP_bd;global CP_bc;global CP_de;global CP_db;
global CP_dc;global CP_df;global CP_eb;global CP_ed;
global d_Tx_cnt;global b_Tx_cnt;global e_Tx_cnt ;global a_Tx_cnt ;global   a_Sendermode;
global   b_Sendermode;global  e_Sendermode;global   d_Sendermode;a_Sendermode=0;

winner=0;CP_ab =0; CP_ae=0; CP_be=0; CP_bd=0; CP_bc=0; CP_de=0; CP_db=0; CP_df=0;
CP_dc =0; CP_eb=0; CP_ed=0;
b_Sendermode=0;e_Sendermode =0;d_Sendermode=0;global TP;global Sender;
```

```
%Total Packets sent
TP=[1:k]  ;

%Calling missing function to calculate missing packets
missing();
% Calling Hello message receive functions
aRxHello();
bRxHello();
dRxHello();
eRxHello();

% Competition at SR and ER phases
while(a_Sendermode˜=0 | b_Sendermode˜=0 |d_Sendermode˜=0 |e_Sendermode˜=0)
    winner= min(min(Boff_aRec ,Boff_bRec),min( Boff_eRec,Boff_dRec))


    if(winner==Boff_bRec)
        Sender='b';
        % Check if the nominated sender is the self node.
        if((sender_to_e == 'b') & (sender_to_d == 'b') & (sender_to_c=='b'))

            %Transmit all the packets in missing array
            for(j=1 : 1 :numel(b_sendTo_e))

                % Check for CP threshold for each packet transmission
                while(CP_be < alpha)
                    Seq_no=(b_sendTo_e(j))

                    % Increase transmission count
                    b_Tx_cnt =b_Tx_cnt+1 ;
                    % Determine receive status
                    option_rec()
                    % Calculate sender CP values.
                    CP_be=SenderCP(CP_be,Qbe);

                end
                CP_be =0;
            end

            b_sendTo_e=0;
            for(j=1 : 1 :numel(b_sendTo_d))
                while(CP_bd < alpha)
                    Seq_no=(b_sendTo_d(j))
                    b_Tx_cnt =b_Tx_cnt+1 ;
                    option_rec()
                    CP_bd=SenderCP(CP_bd,Qbd);
                end
                CP_bd=0;
            end
            b_sendTo_d=0;
            for(j=1 : 1 :numel(b_sendTo_c))
                while(CP_bc < alpha)
                    Seq_no=(b_sendTo_c(j))
```

```
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bc=SenderCP(CP_bc,Qbc);
            end
            CP_bc=0;
        end
        b_sendTo_c=0;
    elseif((sender_to_e == ’b’) & (sender_to_d == ’b’))
        for(j=1 : 1 :numel(b_sendTo_d))
            while(CP_bd < alpha)
                Seq_no=(b_sendTo_d(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bd=SenderCP(CP_bd,Qbd);
            end
            CP_bd=0;
        end
        b_sendTo_d=0;
        for(j=1 : 1 :numel(b_sendTo_e))
            while(CP_be < alpha)
                Seq_no=(b_sendTo_e(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_be=SenderCP(CP_be,Qbe);

            end
            CP_be=0;
        end
        b_sendTo_e=0;
    elseif((sender_to_e == ’b’) & (sender_to_c == ’b’))

        for(j=1 : 1 :numel(b_sendTo_e))
            while(CP_be < alpha)
                Seq_no=(b_sendTo_e(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_be=SenderCP(CP_be,Qbe);

            end
            CP_be=0;
        end
        b_sendTo_e=0;

        for(j=1 : 1 :numel(b_sendTo_c))
            while(CP_bc < alpha)
                Seq_no=(b_sendTo_c(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bc=SenderCP(CP_bc,Qbc);
            end
            CP_bc=0;
        end
        b_sendTo_c=0;
    elseif((sender_to_c == ’b’) & (sender_to_d == ’b’))
```

```
        for(j=1 : 1 :numel(b_sendTo_d))
            while(CP_bd < alpha)
                Seq_no=(b_sendTo_d(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bd=SenderCP(CP_bd,Qbd);
            end
            CP_bd=0;
        end
        b_sendTo_d=0;
        for(j=1 : 1 :numel(b_sendTo_c))
            while(CP_bc < alpha)
                Seq_no=(b_sendTo_c(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bc=SenderCP(CP_bc,Qbc);
            end
            CP_bc=0;
        end
        b_sendTo_c=0;
    elseif(sender_to_e == 'b')
        for(j=1 : 1 :numel(b_sendTo_e))
            while(CP_be < alpha)
                Seq_no=(b_sendTo_e(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_be=SenderCP(CP_be,Qbe);

            end
            CP_be=0;

        end
        b_sendTo_e=0;
    elseif(sender_to_d == 'b')
        for(j=1 : 1 :numel(b_sendTo_d))
            while(CP_bd < alpha)
                Seq_no=(b_sendTo_d(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bd=SenderCP(CP_bd,Qbd);
            end
            CP_bd=0;
        end
        b_sendTo_d=0;
    elseif(sender_to_c == 'b')
        for(j=1 : 1 :numel(b_sendTo_c))
            while(CP_bc < alpha)
                Seq_no=(b_sendTo_c(j))
                b_Tx_cnt =b_Tx_cnt+1 ;
                option_rec()
                CP_bc=SenderCP(CP_bc,Qbc);
            end
            CP_bc=0;
```

```
            end
            b_sendTo_c=0;

        end


        b_Sendermode=0;

    elseif(winner==Boff_dRec)
        Sender='d';

        if((sender_to_c == 'd') & (sender_to_f == 'd'))

            for(j=1 : 1 :numel(d_sendTo_c))
                while(CP_dc < alpha)
                    Seq_no=(d_sendTo_c(j))
                    d_Tx_cnt =d_Tx_cnt+1 ;
                    option_rec()
                    CP_dc=SenderCP(CP_dc,Qdc);
                end
                CP_dc=0;
            end
            d_sendTo_c=0;
            for(j=1 : 1 :numel(d_sendTo_f))
                while(CP_df < alpha)
                    Seq_no=(d_sendTo_f(j))
                    d_Tx_cnt =d_Tx_cnt+1
                    option_rec()
                    CP_df=SenderCP(CP_df,Qdf);
                end
                CP_df=0;
            end
            d_sendTo_f=0;
        elseif(sender_to_c == 'd')
            for(j=1 : 1 :numel(d_sendTo_c))
                while(CP_dc < alpha)
                    Seq_no=(d_sendTo_c(j))
                    d_Tx_cnt =d_Tx_cnt+1 ;
                    option_rec()
                    CP_dc=SenderCP(CP_dc,Qdc);
                end
                CP_dc=0;
            end
            d_sendTo_c=0;

        elseif(sender_to_f == 'd')

            for j=1:1:numel(d_sendTo_f)

                while(CP_df <alpha)

                    Seq_no=(d_sendTo_f(j))
                    d_Tx_cnt =d_Tx_cnt+1
                    option_rec()
```

```
                    CP_df=SenderCP(CP_df,Qdf);
                end
                CP_df=0;
            end
            d_sendTo_f=0;

        end
        d_Sendermode=0;
    elseif(winner==Boff_eRec)
        Sender=’e’;
        if((sender_to_d== ’e’) & (sender_to_b == ’e’))

            for j=1 : 1 :numel(e_sendTo_b)
                while(CP_eb < alpha)
                    Seq_no=(e_sendTo_b(j))
                    e_Tx_cnt =e_Tx_cnt+1 ;
                    option_rec()
                    CP_eb=SenderCP(CP_eb,Qeb);
                end
                CP_eb=0;
            end
            sendTo_b-0;
            for(j=1 : 1 :numel(e_sendTo_d))
                while(CP_ed < alpha)
                    Seq_no=(e_sendTo_d(j))
                    e_Tx_cnt =e_Tx_cnt+1 ;
                    option_rec()
                    CP_ed=SenderCP(CP_ed,Qed);

                end
                CP_ed=0;
            end
            e_sendTo_d=0;
        elseif(sender_to_b == ’e’)
            for(j=1 : 1 :numel(e_sendTo_b))
                while(CP_eb < alpha)
                    Seq_no=(e_sendTo_b(j))
                    e_Tx_cnt =e_Tx_cnt+1 ;
                    option_rec()
                    CP_eb=SenderCP(CP_eb,Qeb);
                end
                CP_eb=0;
            end
            e_sendTo_b=0;

        elseif(sender_to_d == ’e’)
            for(j=1 : 1 :numel(e_sendTo_d))
                while(CP_ed < alpha)
                    Seq_no=(e_sendTo_d(j))
                    e_Tx_cnt =e_Tx_cnt+1 ;
                    option_rec()
                    CP_ed=SenderCP(CP_ed,Qed);
                end
                CP_ed=0;
```

```
            end
            e_sendTo_d=0;
        end
        e_Sendermode=0;
    elseif(winner==Boff_aRec)
        Sender=’a’

        if( (a_sendermode_b ==1) & ( a_sendermode_e ==1))

            Sender=’a’

            for j=1:1:numel(a_sendTo_e)
                Sender=’a’
                while(CP_ae < alpha)
                    Sender=’a’
                    Seq_no=(a_sendTo_e(j))
                    a_Tx_cnt =a_Tx_cnt+1
                    option_rec()
                    CP_ae=SenderCP(CP_ae,Qae);

                end
                CP_ae=0;
            end
            a_sendTo_e=0;
            for j=1 : 1 :numel(a_sendTo_b)
                while(CP_ab < alpha)
                    Seq_no=(a_sendTo_b(j))
                    a_Tx_cnt =a_Tx_cnt+1 ;
                    option_rec()
                    CP_ab=SenderCP(CP_ab,Qab);
                end
                CP_ab=0;
            end
            a_sendTo_b=0;


        elseif(sender_to_e == ’a’)
            for(j=1 : 1 :numel(a_sendTo_e))
                while(CP_ae < alpha)
                    Seq_no=(a_sendTo_e(j))
                    a_Tx_cnt =a_Tx_cnt+1 ;
                    option_rec()
                    CP_ae=SenderCP(CP_ae,Qae);
                end
                CP_ae=0;
            end
            a_sendTo_e=0;
        elseif(sender_to_b == ’a’)
            for(j=1 : 1 :numel(a_sendTo_b))
                while(CP_ab < alpha)
                    Seq_no=(a_sendTo_b(j))
                    a_Tx_cnt =a_Tx_cnt+1 ;
                    option_rec()
                    CP_ab=SenderCP(CP_ab,Qab);
```

```
                end
                CP_ab=0;
            end
            sendTo_b=0;
        end
        a_Sendermode=0;
    end

    % Back-off calculations for SR and ER phase
    if( a_Sendermode˜=0)
        Boff_aRec=(25-10)*rand()+10 ;
    else
        Boff_aRec=inf;
    end
    if( b_Sendermode˜=0)
        Boff_bRec=(25-10)*rand()+10 ;
    else
        Boff_bRec=inf ;
    end
    if( e_Sendermode˜=0)
        Boff_eRec=(25-10)*rand()+10 ;
    else
        Boff_eRec=inf ;
    end
    if( d_Sendermode˜=0)
        Boff_dRec=(25-10)*rand()+10 ;
    else
        Boff_dRec= inf ;
    end

end

end
```

Function to Calculate missing packets:

```
 function missing()



global Qab;global Qae;global Qbe;global Qbd;global Qbc;global Qdc;global Qeb;
global Qed;global b_Rx;global c_Rx;global d_Rx;global e_Rx;global f_Rx;global k;
global b_miss;global c_miss;global d_miss;global e_miss;global f_miss;global sender_to_b;
global sender_to_c;global sender_to_d;global sender_to_e;global sender_to_f;global CP_ab;
global CP_ae;global CP_be;global CP_bd;global CP_bc;global CP_de;global CP_db;global CP_dc;
global CP_df;global CP_eb;global CP_ed;global Best_link;

winner=0; CP_ab =0; CP_ae=0;

CP_be=0;CP_bd=0;CP_bc=0;CP_de=0;CP_db=0;CP_df=0;CP_dc =0;CP_eb=0;CP_ed=0;

global   a_Sendermode;global   b_Sendermode;global  e_Sendermode;global   d_Sendermode;
```

```
a_Sendermode=0;b_Sendermode=0;e_Sendermode =0;d_Sendermode=0;

global TP

TP=[1:k]  ;
% Total number of packets sent so far

% Calculating Missing packets at each node


b_miss=setdiff(TP,b_Rx)
if(numel(b_miss)>0)
    Best_link= max(Qab, Qeb)
    if (Best_link==Qab & Best_link==Qeb)
        sender_to_b= 'a'
    elseif (Best_link==Qab)

        sender_to_b= 'a'
    elseif(Best_link==Qeb)
        sender_to_b= 'a'
    end
else
end


c_miss=setdiff(TP,c_Rx)
if(c_miss>0)
    Best_link= max(Qbc, Qdc)
    if (Best_link==Qbc & Best_link==Qdc)
        sender_to_c= 'b'
    elseif (Best_link== Qdc)

        sender_to_c= 'd'
    elseif(Best_link== Qbc)
        sender_to_c= 'b'
    end
else
end


d_miss=setdiff(TP,d_Rx)
if(d_miss>0)
    Best_link= max(Qbd, Qed)
    if (Best_link==Qbd & Best_link==Qed)
        sender_to_d= 'e'

    elseif (Best_link== Qbd)

        sender_to_d= 'b'
    elseif(Best_link== Qed)
        sender_to_d= 'e'
    end
else
end
```

```
e_miss=setdiff(TP,e_Rx)

if(e_miss>0)
    Best_link= max(Qae, Qbe)
    if (Best_link==Qae & Best_link==Qbe)
        sender_to_e= 'a'
    elseif (Best_link==Qbe)

        sender_to_e= 'a'
    elseif(Best_link==Qae)
        sender_to_e= 'a'
    end
else
end


f_miss=setdiff(TP,f_Rx)
if(f_miss>0)
    sender_to_f= 'd'
else
end
end
```

Function for κ Factor calculation

```
 function kf=kfactor(link1, link2)

L1= link1;
L2 =link2;

px= sum(L1)/4 ;
py= sum(L2)/4;

p_xy= ( (L1(1)& L2(1)) + (L1(2)& L2(2)) +(L1(3)& L2(3))+ (L1(4)& L2(4)))/4 ;
px_py = px * py ;
b=sqrt( px*(1-px)*py*(1-py));

phi= (p_xy - px_py) / b  ;

phi_max= (min(px,py)-px*py)/b;

if (px+py <= 1)
    phi_min= -(px*py)/b ;
else
    phi_min = (px+py-1-px*py) /b;
end

if (phi >0)
    kf= phi/phi_max ;
elseif (phi <0)
```

```
    kf= (-(phi))/(phi_min);
else
    kf=0;
end
end
```

Function to mange hello message reception process

The code for node D is presented here. Other nodes also have similar code structure and logic.

```
 function dRxHello()

global d_Rx;global k;global z;global c_miss;global f_miss;global sender_to_c
global sender_to_f;global Boff_dRec;global d_Sendermode;Boff_dRec= inf ;
Tosend_d=[];global cd_bm;global fd_bm;global  d_sendTo_c;global   d_sendTo_f

d_sendTo_c=[];
d_sendTo_f=[];

%If the hello message receives nodes assess the required packets to send.
    if(k==4)
    if(fd_bm(6)==1)

    d_sendTo_f = intersect(f_miss,d_Rx) ;

    else
    end

    if(cd_bm(6)==1 & c_miss ~=0)
        d_sendTo_c=intersect(c_miss,d_Rx)
    else
    end

elseif(k==8)
    if(fd_bm(7)==1)
    d_sendTo_f = intersect(f_miss,d_Rx);

    else
    end

    if(cd_bm(7)==1)
        d_sendTo_c=intersect(c_miss,d_Rx) ;
    else
    end

elseif(k==12)
   if(fd_bm(8)==1)
    d_sendTo_f = intersect(f_miss,d_Rx);

    else
    end

    if(cd_bm(8)==1)
```

```
        d_sendTo_c=intersect(c_miss,d_Rx) ;
    else
    end

elseif(k==16 & z==0)
   if(fd_bm(9)==1)
    d_sendTo_f = intersect(f_miss,d_Rx) ;

    else
    end

    if(cd_bm(9)==1)
        d_sendTo_c=intersect(c_miss,d_Rx);
    else
    end

elseif(k==16 & z>=10)
  if(numel(fd_bm)==z)
   if(fd_bm(z)==1)
    d_sendTo_f = intersect(f_miss,d_Rx) ;

    else
    end
  else
  end
 if(numel(cd_bm)==z)
      if(cd_bm(z)==1)
        d_sendTo_c=intersect(c_miss,d_Rx);
      else
      end
 else
 end
end

if(sender_to_c == 'd' & sender_to_f == 'd')
Tosend_d= union(d_sendTo_c , d_sendTo_f) ;
elseif(sender_to_c == 'd')
    Tosend_d= d_sendTo_c ;

elseif(sender_to_f == 'd')
    Tosend_d= d_sendTo_f
end

if (isempty(Tosend_d))
   Boff_dRec= inf;
else
      d_Sendermode=1;
    Boff_dRec=(10-5)*rand()+5 ;

end
end
```