MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS IN FOREST FIRE DETECTION

AL-ABBASS Y. AL-HABASHNEH

MAC Protocols for Wireless Sensor Networks in Forest Fire Detection

by

© Al-Abbass Y. Al-Habashneh

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Engineering

Department of Electrical and Computer Engineering

Memorial University of Newfoundland

March 2010

St. John's Newfoundland

Abstract

Power consumption, latency, and complexity are considered to be benchmarks for comparing Medium Access Control (MAC) protocols in Wireless Sensor Networks (WSN). However, the importance of these elements varies according to the application. Furthermore, the reliability of the system is a more specific factor whose importance strongly depends on the application. The term reliability represents the ability of the system to generate authenticated data and transport this data. In this thesis, three MAC protocols are proposed for forest fire detection. Basically, two of these protocols are based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol with some modification to suit the forest fire detection application. These protocols are called Persistent CSMA (P-CSMA) and Per Hop Synchronization CSMA (PHS-CSMA). The third one is a Time Division Multiple Access (TDMA)-based protocol, called Sensor TDMA (S-TDMA). These three protocols are investigated and analyzed by simulation. Moreover, an analytical model is presented for the reliability analysis. Results show that there is no superior protocol which outperforms others in terms of power consumption, delay, reliability and complexity. However, a trade-off does exist. In terms of power consumption, S-TDMA outperforms other protocols but it is not the best in terms of delay, and it has the most complex implementation since a complete time synchronization is required over the entire cluster while PCSMA and PHS-CSMA do not require time synchronization between nodes. PHS-CSMA outperforms P-CSMA in terms of power and delay, but it is more complex since it needs a coordination between nodes to establish a temporary time synchronization at the transmission times. In terms of reliability, all protocols perform similarly. However, at high node failure rates, the presented protocols do not show a great data transport reliability performance. Therefore, route maintenance algorithms are proposed to enhance the reliability performance of the presented protocols. An analytical model is built to examine the reliability performance of the presented algorithm. Analytical results supported by simulation results show that the target is successfully achieved,

where a near unity reliability is attained. Extra delay and power consumption are the price paid for this improvement in reliability.

Acknowledgments

I wishes to express my gratitude to my supervisor, Dr. Mohamed Ahmed who was abundantly helpful and offered invaluable assistance, support and guidance from the initial to the final level enabled me to develop an understanding of the subject. Also I am heartily thankful to my co-supervisor, Dr.Tahir Husain, without whose knowledge and assistance this study would not have been successful.

Special thanks also to my graduate friends, especially Ala'a Al-Habashna and Shadi Alawneh, for their invaluable assistance.

Also, I would like to dedicate this thesis to my beloved family for their understanding and endless love, through the duration of my studies.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Al-Abbass Al-Habashneh

Contents

List of Figures

List of Symbols

Chapter 1

Introduction

1.1 Wireless Sensor Networks (WSNs)

WSNs are wireless networks with a main purpose of monitoring a physical or environmental condition like temperature, humidity, and nuclear radiation. Such networks contain spatially distributed autonomous devices, which are densely distributed inside or near that condition. These devices are called sensor nodes and have two main functions; the first is to collect data from the environment about the condition under surveillance. This job is performed by sensing units attached to these devices. The second function is to send this data to a processing center to be analyzed and manipulated in order to extract useful information about that condition. This job is performed by the transceiver unit attached to the sensor nodes. In each sensor node, these two functions are managed and coordinated by the attached processing unit. The processing capabilities of this unit are usually limited due to the small size of sensor nodes and their limited power recourses. However, based on the application, other units might be needed as well like a location finding system, power generator, and mobilizer [1].

1.2 Design challenges of WSNs

The design of WSNs is a challenging mission, since WSNs have special restrictions and demands that need to be taken into account during the design process. These restrictions make WSN different from any other wireless network. The following are the main points of these demands:

- Low-power consumption: being a microelectronic device, a sensor node can not be equipped with a large power source. Commonly, small size batteries are used. For example, Mica2 [2], which is one of the popular sensor node models by Crossbow brand, uses two AA alkaline batteries as a power supply. Using a sustainable power resource like a solar cell is not suitable for applications with a large number of sensor nodes, like forest fire detection, due to the high cost. Besides that, sensor nodes in most applications are hard to be accessed for maintenance purposes. Therefore, a sensor node's lifetime strongly depends on how long its power supply lasts.

- Scalability: This term represents the ability of a system to manage a large number of nodes. An application may require the use of a huge number of sensor nodes, thus the used algorithms must be able to manage such a large number of sensor nodes.

- Cost: because of the large number of sensor nodes and the high probability of sensor node failure (due to power depletion or environmental stress), minimizing the cost as much as possible is needed to justify the minimum overall cost of the network.

- Small size of sensor nodes: this attitude of manufacturing is adopted in WSNs for easy deployment and installation especially for the applications which has a harsh environment to be accessed. Moreover, for minimizing cost and power consumption, a smaller size device is preferred.

- Fault tolerance: failure of sensor nodes is frequently caused by the environmental stress and power depletion. Hence, the sensor network has to be able to keep

serving the generated traffic even if some of nodes are down. This requires frequent reconfiguration of the network routing.

- Traffic balance: the traffic distribution strongly influences the life time of the sensor networks since unfair power consumption could lead some nodes to fail due to power depletion before others. For example, since generated packets are delivered to data sink using multi-hop transmission, the nodes, which are closer to the data sink, relay more packets than those which are far from the data sink. Therefore, the close ones are expected to consume more power and fail before the far ones.

1.3 Sensor networks architecture and general operation

In most WSNs, sensor nodes lie on the ground. Therefore, the wireless signals sent from sensor nodes suffer a high path loss [3], [4], and the signals' power drops off quickly with high exponents along with the distance (i.e., $Power\ \alpha\ d^{-n}$), where d and n are the distance and the path loss exponent respectively, while $2 \leq n \leq 4$ [1]. Due to this fact, multi-hop transmission strategy is adopted in WSN. This is because sending a message from a source node to a destination node directly over a long distance consumes much more power than sending the same message with multi-hop relaying using intermediate nodes. Therefore, nodes must organize themselves to construct a multi-hop network.

In all WSNs, data is forwarded to an aggregation point called a *data sink* which forwards this data to the processing center. This center is the final destination, where this data is analyzed and manipulated. The idea behind the use of a sink node, which has unlimited power recourses, is that sensor nodes might be far away from the processing center. Thus, nodes will consume a huge amount of power to send the data directly to the far processing center.

Architecturally, the network topology can be either flat or clustered. In the flat topology, nodes send their traffic directly to the data sink using multi-hop transmission without considering any intermediate aggregation points, and the data sink is the only aggregate point in the network. In the clustered topology, nodes are divided into groups of neighbors. Each group is called cluster and has one aggregate node which is called the *cluster head* (CH), whose responsibility is to collect data from the nodes in the cluster and forward it to the sink [5], [6]. Member nodes in a cluster can not communicate with any other nodes out of that cluster. The main advantage of the clustered topology is that it supports a higher scalability to the network [7], [8]. Higher scalability is achieved because the *CH* does not relay all the aggregated packets to the data sink. Instead, a representative packet of the cluster is generated and sent to the data sink. This minimizes latency in the network and minimizes power consumption as well. Hence, the network can accommodate a larger number of nodes. Figure 1.1 shows the topology of the clustered WSN.

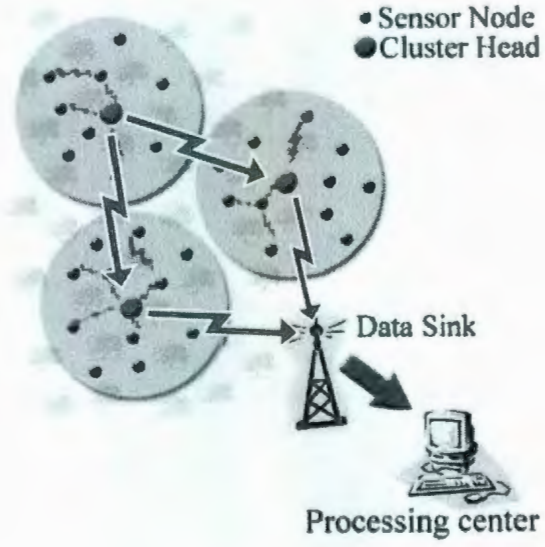


Figure 1.1: Clustered WSN topology.

1.4 WSNs in Forest Fire Detection

Forest fires are considered to be a serious natural disaster, where in some cases they cause a huge threat to public safety and natural resources. For example, McGillivry [9] was one of the disastrous forest fires. It was ignited by lightning on August 15, 2003, and it was situated in north Kamloops, BC, Canada. The wind, whose speed was 50 -70 km/h at that time, caused the fire to spread and to become out of control. The fire lasted for approximately 14 days. the McGillivry fire's record reports that 11,400 hectare were burned, and the total cost was estimated to be $ 25.7 million! Considering the dangerous nature of wildfires and their damage, significant attention has been brought to forest fire detection by researchers. Thus far, fire detection systems have relied on satellite imaging. The Moderate Resolution Imaging Spectroradiometer (MODIS) [10], which is the first moderate-resolution imaging instrument, was launched in outer space in December 1999. MODIS detector measures 36 spectral bands between 0.405 and 14.385 μm , and it acquires data at three spectral resolutions: 250 m, 500 m, and 1000 m. The MODIS takes one to two days to view the entire earth's surface. Generally, such systems take a long time to detect a fire occurrence; this delay time mainly depends on how often images are taken to the forest by the satellite (i.e., imaging rate) and the processing time of the images taken. Truly, time delay is the main drawback of satellite imaging since earlier forest fire detection means earlier intervention or response. Therefore, time is a critical factor in fire fighting.

A fire detection system based on WSNs , which can provide a real time monitoring, is more efficient than a satellite imaging system in terms of delay reporting time. Moreover, WSN-based systems are able to predict a potential fire occurrence based on currently sensed weather parameters [11].

A WSN-based forest fire detection system consists of a large number of sensor nodes, which are deployed in the forest to be monitored. Mainly, these nodes are scattered in a

random fashion covering the entire forest area. One of the possible deployment methods is through airplane dropping. Each sensor node is equipped with a sensing unit which consists of one or more transducers to sense some of the weather parameters. For example, based on the fire weather index (FWI) system [11], which is one of the most comprehensive forest fire danger rating systems, temperature, humidity, wind speed, and rain level are measured to calculate fire occurrence probability. Therefore, many transducers have to be attached to each sensor node. The following are some related issues to this application:

- Limited power: most WSN applications share this restriction, but this application requires extra attention on power saving since sensor nodes are not reachable. Therefore, once the node's power is depleted, the node fails. This fact implies that generally in forest fire detection application, a sensor node's lifetime strongly depends of its batteries' lifetime. Minimizing the consumed power by sensor nodes can significantly extend the network's lifetime.

- Wireless channel: most of the WSN applications, in which sensor nodes are laid on the floor, have a challenging wireless channel. Forest fire detection application implies an even harsher wireless channel since the forest environment is full of trees, which have rough surfaces. Thus, signals are expected to experience high scattering, which weakens the signals and results in a low signal to noise ratio.

- Localization: localization is a concern from which many operations can gain benefits. For example, many routing protocols, which are called location aware routing protocols, are based on the geographical locations of terminals like Two-Tier Data Dissemination (TTDD) [12]. However, in the forest fire detection application, localization is an essential feature because it is crucial to localize the node which has reported a fire occurrence. Using a Global Positioning system (GPS) is applicable but costly. Another approach is to perform localization algorithms. These algorithms implement reference nodes as a backbone. Reference nodes are sensor nodes

with known locations found by either equipped GPS systems or pre-determined location deployment [13]. In this approach, other sensor nodes estimate their locations with respect to the reference nodes. After estimating its location, a node becomes a reference node and helps other nodes to estimate their positions.

- Data reliability: this aspect is a very important one in forest fire detection application which requires high data reliability because any fire mis-reporting could result in a natural disaster. Moreover, sensor nodes have a high failure probability for this application due to the harsh environment. However, two levels to guarantee reliable data should be considered for this application; *data sensing reliability*, which measures the accuracy of sensed data, and *data transport reliability*, which measures the transportation reliability of the generated data. *Data sensing reliability* can be enhanced by using accurate physical transducers (thermometer, humidity meter ...etc.), implementing data aggregation, or by increasing the number of nodes that monitor the same area[14]. Different communication layers can cooperate to improve *Data transport reliability*. For example, in the data link layer, link level retransmission can be used, while multi-path routing can be implemented in the network layer. Later in this thesis, reliability is discussed in greater detail.

- Delay: for this application, packet delay time is not as critical as the other aspects. This assumption is built on the fact that the environmental observations which are monitored (such as temperature and humidity) are originally slow possess.

In [14], the authors present a design of WSN- based forest fire detection system using the FWI system. The main focus in that work is k-coverage problem, which is to calculate the number of nodes to cover the same point to guarantee a certain level of data reliability. In [15], the authors present environmental results about the values and gradients of weather conditions like temperature and humidity during a fire reporting. The presented results are collected from the field during a prescribed test burns. In [16], the authors

present a special hardware circuitry design for forest fire detection based on CC2430 chips.

1.5 Medium Access Control (MAC) protocols for WSN

As indicated earlier, power consumption is one of the primary objectives in WSNs design. Therefore, the MAC protocols for other wireless networks, which do not consider the power consumption as a first priority like cellular and bluethooth networks, could not be used by WSNs. Moreover, WSNs adopt multi-hop transmission rather than direct transmission, which is used in other wireless networks like cellular networks. Because of these differences, WSNs have their own MAC protocols to fulfill their demands, such as considering minimal power consumption as a first priority [1]. In [17], the authors emphasize the main sources of power wastage in the IEEE 802.11 standard as if it is applied in WSNs [18]. These sources are identified as follows:

- Collision: the collide packets are required to be retransmitted which consumes extra power.
- Overhearing: when a node receives a packet which is not destined to it, the packet is discarded. Thus, extra power is consumed.
- Overhead:IEEE 802.11 uses control packets like Request To Send (RTS), Clear To Send (CTS) and Acknowledgment (Ack). These control packets consume extra energy as well.
- Idle listening: IEEE 802.11 standard requires that the node, which is not engaged in any transmission or reception, keeps listening to the channel waiting for a possible transmission. This idle listening consumes extra power.

Based on the IEEE 802.11 standard and its power wastage sources, the authors in the same work [17], proposed a contention-based MAC protocol for WSN called Sensor MAC

(S-MAC), where nodes are switched to the sleep mode (turning their radio off) to save power when they are not participating in transmission or reception.

While the IEEE 802.11 standard provides data rates of 1-54 Mbps with a relatively high power consumption, the IEEE 802.15.4 standard [19] is designed to adapt low rate networks with low power consumption. These networks are called Personal Area Networks (PAN). The IEEE 802.15.4 standard provides data rates of 20, 40, and 250 kbps. Two topologies are provided by this protocol; a centralized one called *the star topology* and a non-centralized or distributed one called *the point-to-point topology.* However, both topologies consider a coordinator within the network. This coordinator has more responsibilities in the star topology where all devices perform their operations through this coordinator. On the other hand, in the point-to-point topology, devices are more independent in their operations and do not rely on a coordinator. Many works have analyzed the performance of the IEEE 802.15.4 [20],[21]. Many other studies have been done on the MAC protocols of WSN under both; the contention based MAC protocols [22],[23], and the contention-free based MAC protocols [24],[25]. In [22], the authors propose an Enhanced-Carrier Sense Multiple Access (E-CSMA) MAC protocol to improve the reliability of the network as well as the power consumption. In [23], the authors present a low duty-cycle MAC protocol called Convergent MAC (CMAC). The main goal for CSMA is to minimize the communication between nodes while there is no traffic without using any kind of time synchronization. In [24], the authors proposed a Time Division Multiple Access (TDMA)-based MAC protocol called Power Aware Clustered TDMA (PACT). This protocol minimizes power consumption by using adapted TDMA duty cycles to user traffic, and by using passive clustering, in which not all nodes in the cluster operate all the time. Instead, a set of the nodes inside the cluster is chosen to operate at a time. In [25], the authors presents a TDMA MAC protocol called Traffic-adaptive medium access protocol (TRAMA). This protocol saves power by minimizing collisions and assigning time slots just to the nodes which need to transmit.

1.6 Problem statement and Contribution.

As previously stated, there is enough justification for the existence of special MAC protocols for WSN. However, a wide variety of WSN applications does exist, which raises the question as to if these MAC protocols fit all WSN applications, or application-oriented MAC protocols are needed. Motivated by this discussion and the importance of forest fire detection, this application is studied intensively to identify its special characteristics, and its main requirement. Therefore, we made the main features of a MAC protocol for this application as follows:

- Power saving is critical; this is valid for most applications, but here it is even more crucial.
- High data transport reliability is required to guarantee reliable fire reporting .
- This application is tolerant to short delay time since weather observations are generally slow.
- Simplicity: this is required for all WSN MAC protocols since sensor nodes are small devices with limited processing capabilities.
- Scalability: the MAC protocols should be able to manage a large number of nodes in the network, where the surveillance area is very large.
- Cost: As forest fire detection is considered as a large scale application of WSNs because of the large number of sensor nodes in this application, the cost should be minimized.

Most of the proposed WSN MAC protocols in the literature do not sufficiently stress the simplicity in the design. For example, Sensor-MAC [17] applies a sophisticated synchronization mechanism, which is quite difficult to implement in large scale networks. The IEEE 802.15.4 [19] standard focuses mainly on the star topology more than the

point-to-point topology. As previously mentioned, star topology requires a single device to coordinate the other nodes operation. This is still hard to implement in large scale networks.

Clearly, the MAC protocol design in this application might benefit from the flexibility in delay time, to support saving power and higher reliability. However, to the best to our knowledge, no application-oriented MAC protocols have been proposed for forest fire detection application.

In this thesis, three MAC protocols are proposed to fit the demands of forest fire detection application. These protocols apply different time synchronization levels; the first, which is called Persistent Carrier Multiple Access (P-CSMA), considers no time synchronization between nodes. The second, which is called Per Hop Synchronization CSMA (PHS-CSMA), considers a temporary time synchronization between nodes. And the last, which is called Sensor Time Division Multiple Access (S-TDMA), considers a full time synchronization between nodes. To support scalability, the network is divided into clusters, where each cluster has an aggregate point called the *CH*, whose responsibilities do not exceed collecting data from other nodes in the cluster and forward it (through other CHs) to the data sink. Limiting the privileges of CHs is considered to maintain the simplicity of our design. The power consumption and delay performances are evaluated by simulation.

On the reliability matter, node density is studied to optimize the number of nodes in the network. Moreover, fault management mechanisms are also proposed to support high data transport reliability. An analytical solution is used to evaluate the reliability performance of the proposed protocols with and without implementing the proposed fault management mechanisms.

The contribution of this thesis can be summarized as follows:

- Forest fire detection application in WSNs is studied intensively in order to highlight its characteristics and main requirements.

- Three MAC protocols are proposed to fulfill this application's requirements. The proposed protocols are studied extensively where different variations are compared to optimize the power consumption and delay time.
- To maximize the reliability of the system:
  - An analytical solution is used to optimize the node density in the network.
  - Three route maintenance mechanisms are proposed to enhance the reliability performance.
  - An analytical solution is used to test the performance of the proposed mechanisms in terms of the reliability. The results of the analytical solution are confirmed by a simulation model.

Chapter 2

System Model and MAC Protocols Design

In this thesis, there MAC protocols are proposed specially for forest fire detection. In this chapter, the system model and the design details of these protocols are presented.

2.1 System Model

As emphasized in the introduction, forest fire detection application is one of the applications that require a large number of nodes (i.e., large scale networks). Therefore, clustered topology is considered in this work to support scalability to the network. A *CH* node with unlimited power resources is considered to be located at the center of the cluster. The cluster's size is a design factor. For this work, a cluster is considered to be a circular area with R_{cl} radius.

Sensor nodes are assumed to have a communication range of R_c, and current dissipation in the active (transmitting), active (receiving) and sleep modes of I_t, I_r and I_s, respectively. Each sensor node generates an information message every T_{wi}, with data rate f_b and message length L.

As mentioned in the introduction, this work has a major concern about the data link

layer. However, minimum number of hops routing is applied in our analysis, and packets re-routing is possible in some cases. Each node is considered to have already created its routing tables and gathered all the needed routing information. This information contains

- Node's level, which is the number of hops needed to reach the *CH*.
- Node's parents, which are the neighbor nodes whose levels are lower than the node's level (i.e., the nodes who have less number of hops to reach the *CH*).
- Node's brothers , which are the neighbor nodes whose level is the same as the nodes level (i.e., the nodes who have the same number of hops to reach the *CH*).
- Node's children, which are the neighbor nodes whose levels are higher than the node's level (i.e., the nodes who have larger number of hops to reach the *CH*).
- The address of the *CH*.

2.2 Persistent-CSMA (P-CSMA)

2.2.1 Basic version

P-CSMA is the simplest MAC protocol among the proposed protocols. Figure 2.1 and Figure 2.2 present the basic operation of this protocol. It can be seen from Figure 2.1 that the timeline of sensor nodes in this protocol is composed of wake-up intervals (T_{wi}), where each T_{wi} contains active time period (T_{ac}) and sleep time period (T_{sl}). During T_{ac}, a node turns its transceiver on to receive possible transmissions from other nodes. On the other hand, the node turns its transceiver off during T_{sl} to save power. Basically, no time synchronization or any previous coordination exists between nodes.

This protocol requires a source node, which has a packet to forward, to send an undestined Request To Send (RTS) message (without addressing the receiver node). Awake parents, who receive this RTS successfully, contend to respond by sending Clear To Send

(CTS) messages back to the source node after a random back-off (BO) time. This BO is necessary to avoid any collision that could happen in case more than one parent node are awake at that time. Once the CTS is received, the packet transmission is initiated and all neighbor nodes go to sleep mode. A positive acknowledgment (ACK) is used to report a successful transmission.

In case of a collision occurrence because of a hidden terminal, which was in the sleep mode when the CTS had been sent, the whole process is repeated after a random BO time. If no CTS message is received by the source node (i.e., all parent nodes were in sleep mode when the RTS was sent), a RTS is sent again after a random BO period time. This BO period should be shorter than T_{wi} to guarantee that at least one of the parents is able to receive a RTS from the source node.

Figure 2.2 shows that after a pre-defined period, which is called *waiting time*, is elapsed with no CTS received at the source node, the source node considers that all parents nodes are not operating. Therefore, the source node decides that it is disconnected from the network (isolated), and drops all the packets in its queue. This node is no longer participating in any activities.

The *waiting time* period is a design parameter, which should be carefully chosen. This period should be long enough to avoid a wrong decision that a node is isolated. Such a wrong decision could be made by the node while there is an available parent node, but the *waiting time* is shorter than what is required to get a response from that parent. On the other hand, *waiting time* should not be very long to avoid unnecessary extra delay time. This extra delay is due to the useless RTS messages, which are sent when no parents nodes are available to respond. These RST messages occupy the channel when it can be used for other transmissions. The optimization analysis for *waiting time* are presented later in this thesis.

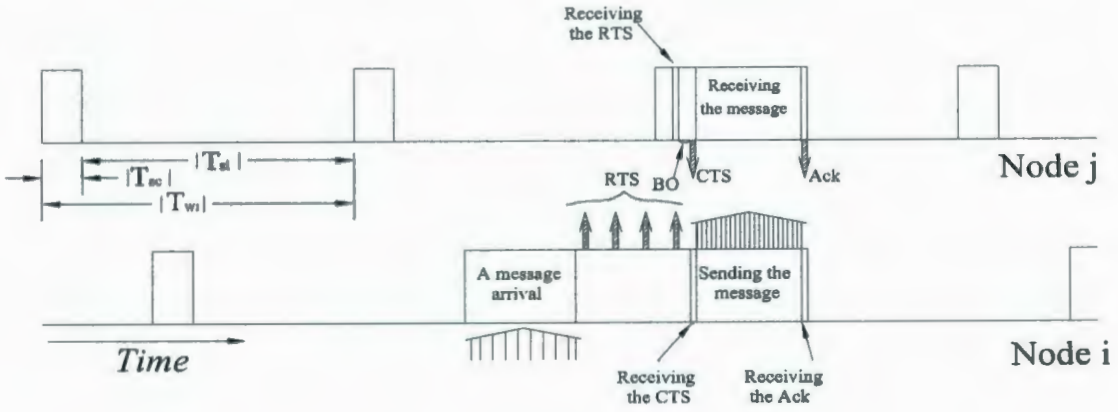


Figure 2.1: Timeline of P-CSMA and its operation.

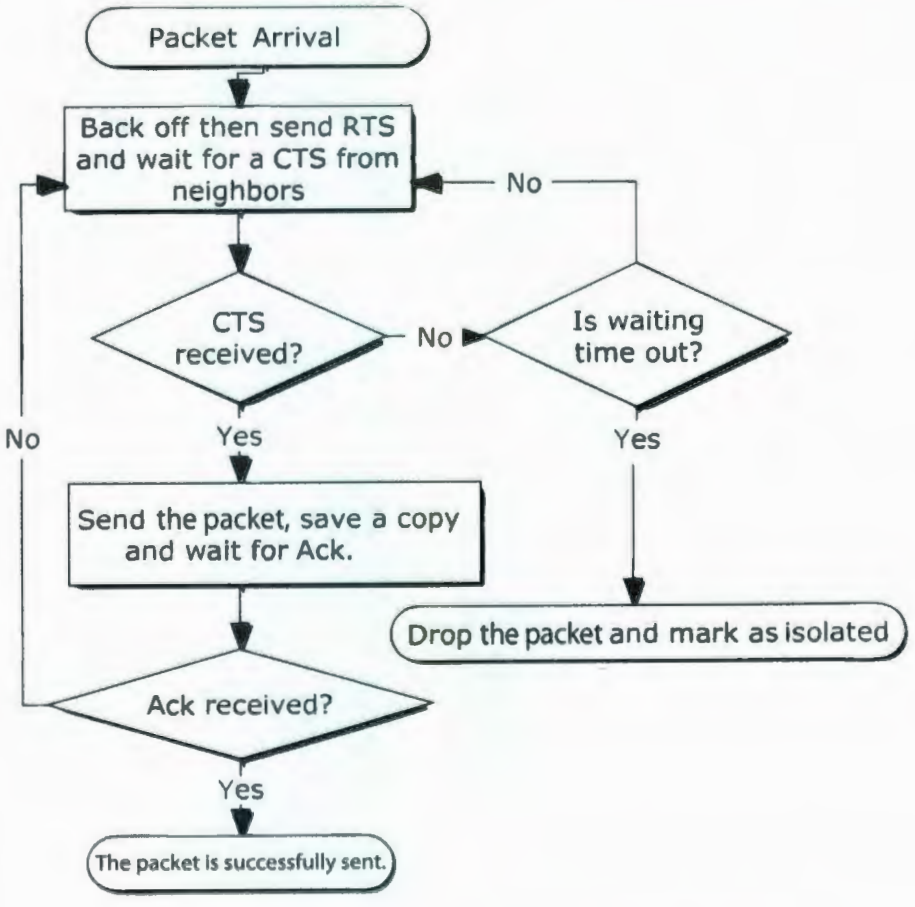


Figure 2.2: Flow chart of P-CSMA: Basic version.

2.2.2 Enhanced version

P-CSMA has a limited reliability performance. Such a low reliability comes from the fact that no route maintenance mechanisms are implemented in this protocol. Therefore, any node that fails to communicate with its parents stops communicating with the network even if it can still communicate through its brothers or its children. For example in Figure 2.3, node $n_2^{(2)}$ has two parents, which are $n_1^{(2)}$ and $n_1^{(3)}$. Both of these parents failed. Therefore, $n_2^{(2)}$ is isolated and does not send its traffic or forward other nodes' traffic. Clearly, this node still can send and forward packets through its brother $n_2^{(1)}$ or its children $n_3^{(1)}$, but this needs to use route maintenance mechanisms, which are not implemented yet in the basic version of P-CSMA.

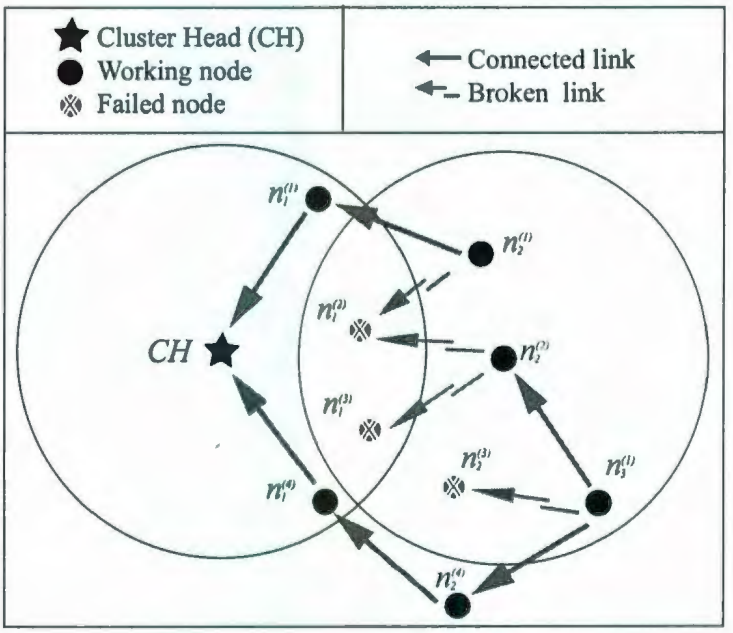


Figure 2.3: Example of node isolation.

Despite being unable to perform route maintenance, it should be emphasized that a node in the basic version of this protocol is able to discover route failure and to decide whether or not it is isolated (disconnected from the network).

As a modification to support route maintenance, a node responds to a received RTS

message based on the level of the sender, and not based on whether the sender node is listed in the children list or not. This approach supports more flexible routing choices for the source node. However, this approach adds a small overhead to keep nodes updated with their neighbors' levels. This can be done simply by including the level of a node in its RTS messages.

When a RTS is sent by a source node, awake neighbors with lower levels, which receive this RTS, contend to send CTS and proceed to receive the packet from the RTS's sender. But, if CTS is not received during that *waiting time*, the source node increases its level by one and restarts the process. This increases the probability that a neighbor node with a lower level than the sender node's level is available. For example in Figure 2.3, node n_2^2, whose level 2, has level 1 neighbors ($n_1^{(2)}$ and $n_1^{(3)}$), level 2 neighbors ($n_2^{(1)}$ and $n_2^{(3)}$), and level 3 neighbor ($n_3^{(1)}$). When node $n_2^{(2)}$ has a packet to send or forward, it sends a RTS, which includes its level (2). Since, $n_1^{(2)}$ and $n_1^{(3)}$, whose levels are lower than 2, are dead, no response (CTR message) will be sent back. After the *waiting time* is elapsed with no response, $n_2^{(2)}$ increases its level by 1 to become 3 and restarts the process. After this modification, $n_2^{(3)}$ is able to respond by a CTS message, since it has a lower level than $n_2^{(2)}$'s. Similarly, if $n_2^{(3)}$ also fails, $n_2^{(2)}$ will increase its level again to become 4 and can communicate through $n_3^{(1)}$.

Figure 2.4 shows the basic operation of the modified version of P-CSMA. However, the modified version still does not allow more than *MAX_LEVEL* hops in the system. Therefore, nodes can not fix route failure after they reach that *MAX_LEVEL* and decide to stop participating in any activities (i.e., become isolated). *MAX_LEVEL* is a design parameter and it is set to 4 in this work because it guaranties 0.999 probability of node connectivity in our system model. This probability is calculated based on our analytical model, which is presented later.

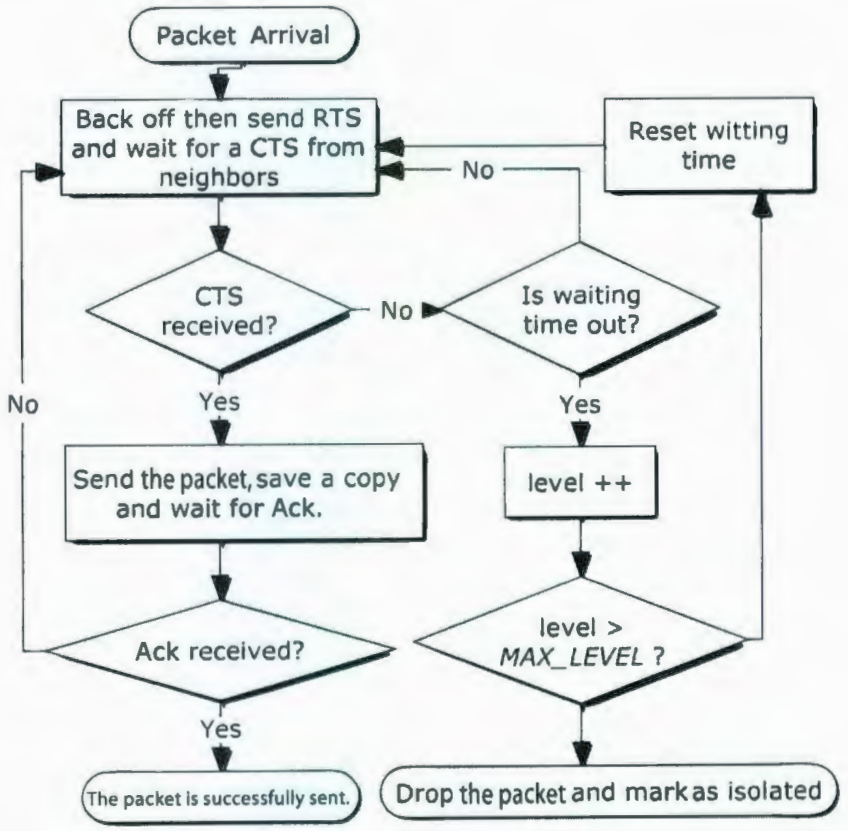


Figure 2.4: Flow chart of P-CSMA: Enhanced version.

2.3 Per Hop Synchronization-CSMA (PHS-CSMA)

2.3.1 Basic version

The second proposed protocol is PHS-CSMA. Figure 2.5 shows the timeline for sensor nodes in this protocol. Clearly, the timeline for this protocol is the same as for P-CSMA except that nodes send beacon signals to announce their active mode at the beginning of each T_{wi}. Each node should sense the channel before sending its beacon to avoid collisions. The operation of this protocol is as follows: once a node has a packet to send, it plans to wake up just before the next parent wake-up interval. At the planned time, the node turns its transceiver on and starts listening to the beacons from that parent. If the expected beacon is received, the source node assumes that this parent is available. Slotted CSMA is used by the source node to send its packet. Ignoring the propagation time, since it is very small compared to the transmission time at low data rates, both the source node and its parent node are considered to be temporarily time synchronized based on the received beacon. Dividing the active period of that parent into time slots, the source node sends its packet at the beginning of one of these slots. There are two main benefits of using slotted-CSMA, which are the following:

- Reducing the probability of collision in case more than one node are interested in sending their packets to the same parent node. These nodes wait for the same parent's beacon. After receiving that beacon, each node chooses a time slot randomly. The one which has the first chosen time slot wins the channel and starts sending. Other nodes, who can hear this transmission, postpone their transmissions to the next frame. However, collisions are still possible in the case of more then one node choosing the same time slot, or in the case that one node does not hear the sending node (hidden terminal).

- Reducing over hearing (a node receives a packet, which is not intended for the node).

This can be explained as that for a node, if a packet is destined to it, the packet should be synchronized with its beacon. Thus, if it happens that a node hears a transmission starting at any time except at the beginning of its time slots, this node assumes that this transmission is not intended for it and goes to sleep.

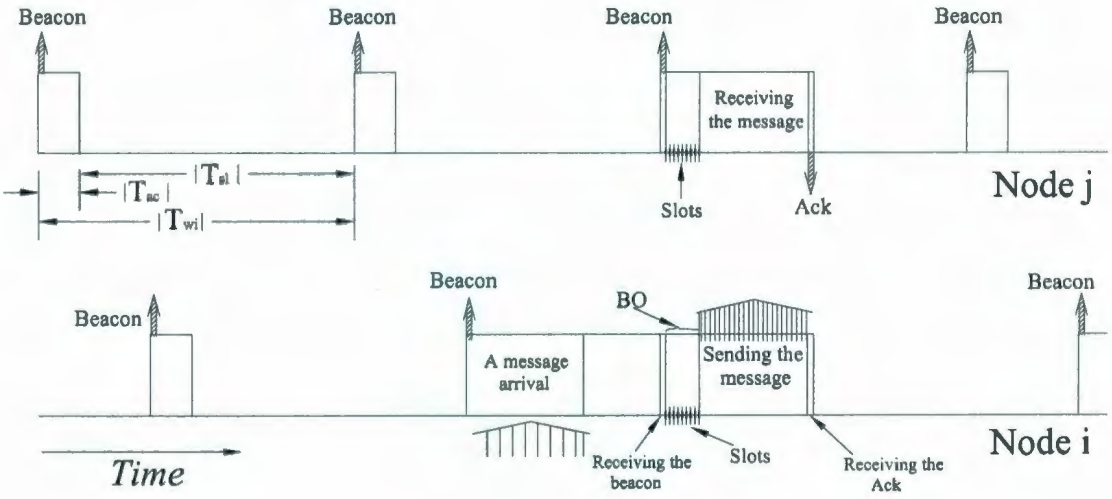


Figure 2.5: Timeline of PHS-CSMA and its operation.

Similar to P-CSMA, Figure 2.6 shows that after a pre-defined period of time, which is called *waiting time*, is elapsed with no CTS received by the source node, the node considers that all parents nodes failed. Therefore, the source node decides that it is disconnected from the network (isolated), and drops all the packets in its queue. This node does not participate in any further activities.

A collision between beacons is a serious problem because such collisions happen frequently during every T_{wi}. To solve this problem, a simple algorithm is applied at the earliest phase of this protocol. In this phase, nodes choose their schedule (i.e., when to transmit a beacon during T_{wi}) as follows: each node has its own address or ID. Simply, if we have 100 nodes for example, the nodes' IDs will be 1, 2, 3, ..., 100. For each one hop neighbors group, the sensor node which has the smallest ID number chooses its schedule randomly. Other nodes, with larger ID, wait until hearing all beacons of other nodes that

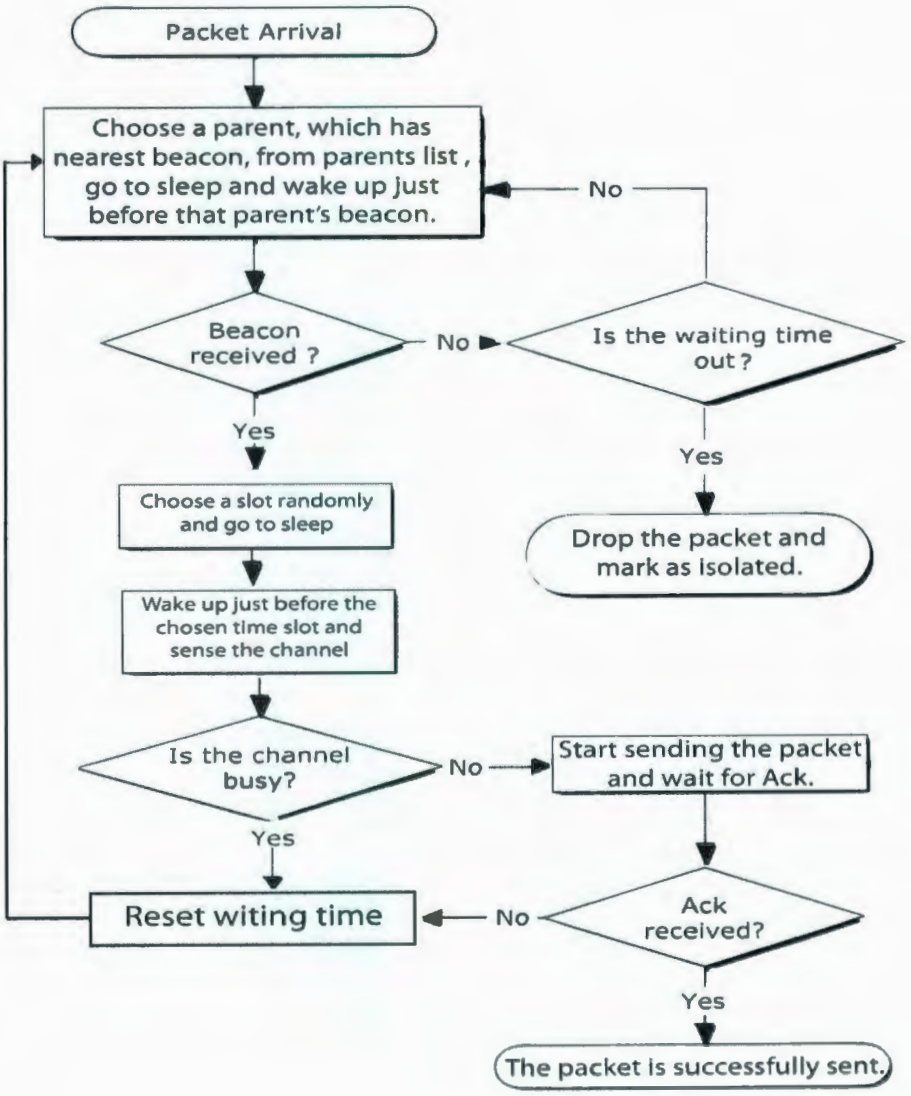


Figure 2.6: Flow chart of PHS-CSMA : Basic version.

have smaller IDs. For each one of these nodes, the maximum delay time between the received beacons is calculated, and the beacon transmission time is randomly chosen around the middle of that maximum delay time. By doing so, an almost uniform distribution of beacons is achieved over T_{wi}. As a result, a collision free beacons system is achieved. Figure 2.7 shows the operation of this mechanism.

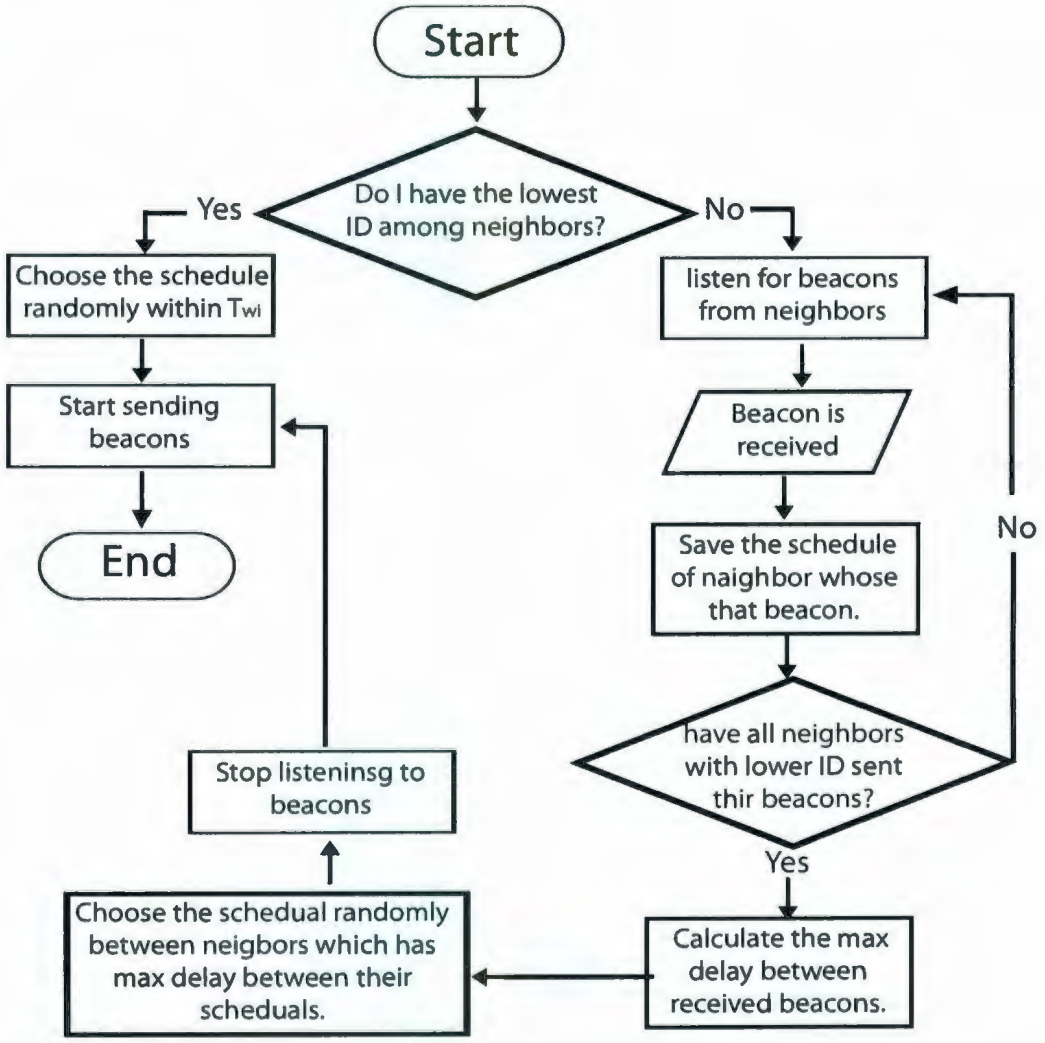


Figure 2.7: Flow chart to choose schedules in PHS-CSMA.

2.3.2 Enhanced version

Based on the original version of this protocol, after a predetermined *waiting time* with no received beacon, the node decides that it is isolated. As discussed before in P-CSMA, this approach limits the reliability of the system. This situation can be dealt with as in P-CSMA. In the modified version of PHS-CSMA protocol, each node includes its level in its beacon. After that *waiting time* is elapsed, the waiting node increases its level by one. Therefore, the source node has more flexibility to send its packet to any of its neighbor nodes as long as that neighbor has a lower level. Figure. 2.8 shows the basic operation of the modified version of this protocol. In the modified version, node levels still do not exceed *MAX_LEVEL*. When a node reaches that limit and detects a route failure, this node is marked as isolated.

2.4 Sensor-TDMA (S-TDMA)

2.4.1 Basic version

The third proposed MAC protocol is S-TDMA. This one is the most complex MAC protocol (in terms of implementation) among the proposed protocols because it requires time synchronization between sensor nodes in the same cluster. Many protocols have been proposed for time synchronization. Based on the IEEE 802.15.4 standard [19], time synchronization can be implemented using a coordinator sensor node. This coordinator periodically broadcasts a beacon signal to allow the other nodes in the cluster to synchronize themselves with the network. In the case of multi-hop topology, which is the case in our scenario, more than one coordinator can be used. Timing-synchronization Protocol for Sensor Networks (TPSN) protocol [26] does not consider such a coordinator, where time synchronization is achieved by exchanging synchronization packets between neighbors. A synchronization packet contains time stamps for its source sensor node. Using

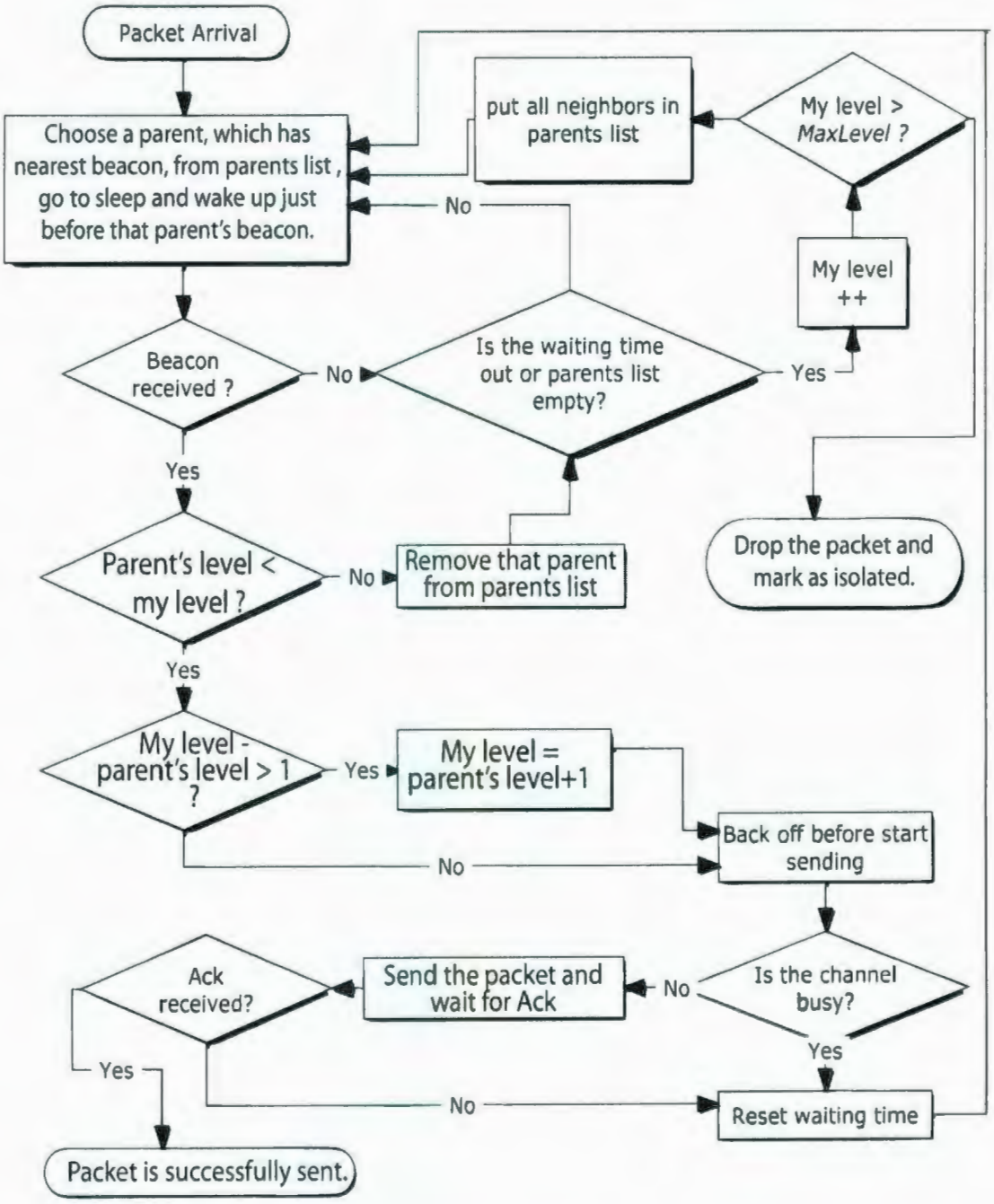


Figure 2.8: Flow chart of PHS-CSMA: Enhanced version.

these time stamps, neighbors calculate the time drift between their clocks. However, we assume that the implementation details of the time synchronization algorithm is out of the scope of this thesis, and sensor nodes are assumed to be time synchronized.

Figure 2.9 shows the time line of sensor nodes using S-TDMA protocol. This time line is divided into cycles or wake up (T_{wi}) intervals. Since sensor nodes are required to be synchronized for this protocol, T_{wi} intervals for all nodes start at the same time. T_{wi} is composed of sleep time T_{sl} and active time T_{ac}. During T_{sl}, all nodes go to sleep mode to save power. T_{ac} is divided into N time-slots, where N is the number of nodes in the cluster. A node goes to active mode just in particular time slots (i.e., a node is not up all the time during T_{ac}).

According to this protocol, time-slots are distributed over the nodes in the cluster, where each node has one time slot. This time slot is called the *outgoing time slot.* Using its outgoing time slot, the node sends its messages to its parents. However, a node can not receive traffic during its outgoing time slot. Therefore, the node goes to the sleep mode in that time slot if it does not have traffic to send. On the other hand, each node should be awake in the outgoing time slots that are related to its children to serve their possible traffic. Each node considers a certain time-slot as an incoming time-slot if that slot is an outgoing time-slot to one of its children.

Figure 2.10 shows the basic operation of sensor nodes in this protocol. When a sensor node has a packet to forward, it chooses one of its parents as a receiver of this packet and sends the packet in its outgoing time-slot. This parent has to be awake at this time-slot since the source node is one of its children. To confirm a successful transmission, the source node listens to the next outgoing time-slot of the receiver for Contention Window (CW) times. If any transmission activities are detected, the source node assumes that the receiving node is operating (not failed) and that it has received the packet successfully. In the case where no transmission activities from the receiving node are detected, the source node assumes that this parent is failed and removes it from the parents list. If there are

more parents left in the parents list, the source node tries with another node from the parents list. Ending up with an empty parents list, the source node decides that it is isolated and stops participating in any activities. A new parent can be chosen to receive the packet in one of two ways:

- Randomly: this approach implies that all parents of the source node have to be awake all the time during its outgoing time-slot.
- Sequentially: this approach implies that just one parent is awake per each T_{wi}. In this case, different parents alternate in getting up in the incoming time slot. For example, if the source node has 2 parents, then in the first cycle (i.e.,T_{wi} #1) just the first parent is up and the second one is in sleep mode. Similarly, in the second cycle, the second parent is up and the first one sleeps. In the third cycle, the first parent is up again while the second one sleeps and so on. Based on this discussion, choosing the receiving parents is basically done based on the cycle number. This approach is better in terms of power saving, since it reduces the number of time-slots during which parent nodes are awake, which saves nodes power.

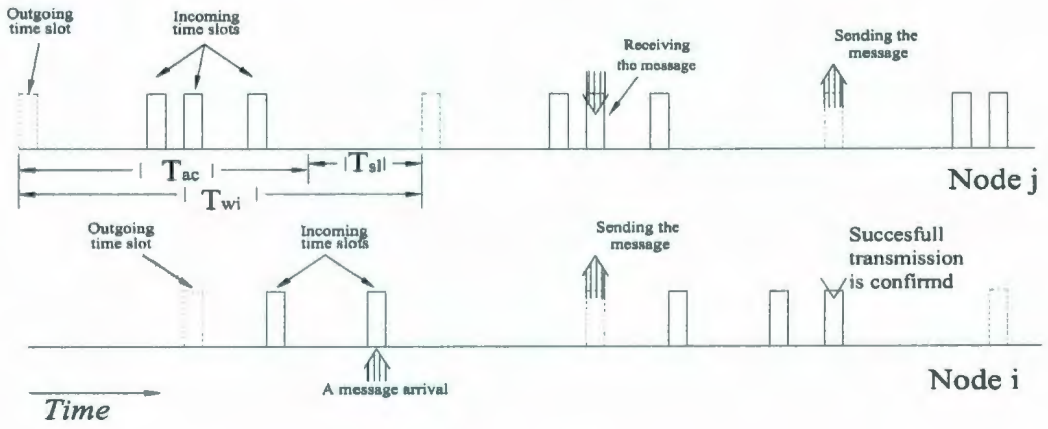


Figure 2.9: Time line of S-TDMA and its operation.

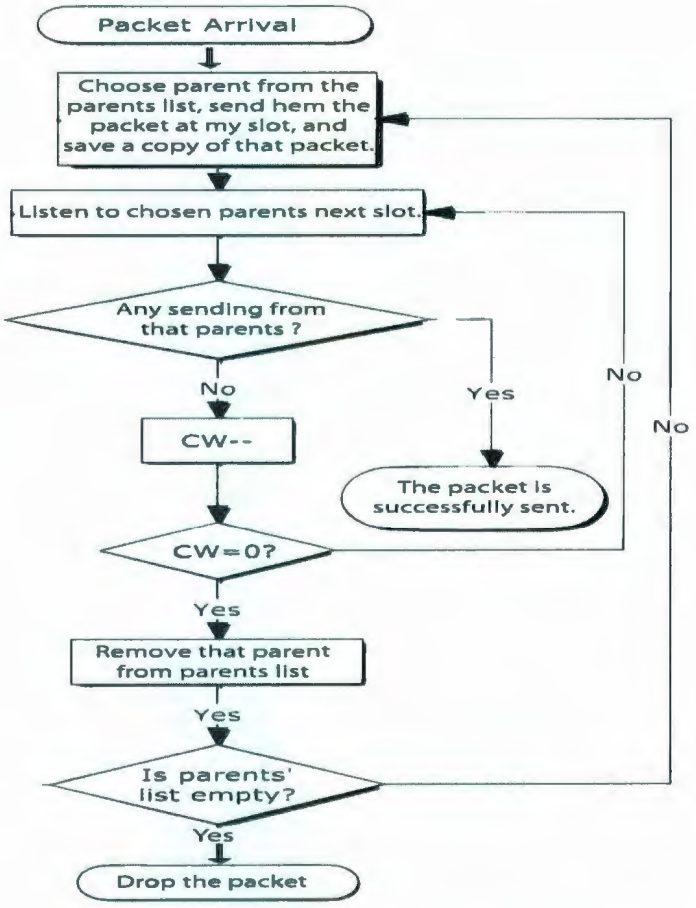


Figure 2.10: Flow chart of S-TDMA: Basic version.

2.4.2 Enhanced version

Based on the original version of this protocol, to end up with an empty parents list, the node assumes its isolation from the network. However, this limits the reliability of the system, since a node assumes its isolation when it still can communicate through its other neighbors. This situation can be fixed as shown in Figure 2.11. After ending up with an empty parents list, a new node can be added to the parent list out of the brothers list. This can be done by sending the packet to the brother who has the nearest outgoing time-slot using its outgoing time-slot. If a collision happens, the node tries with another brother. If no collision is detected, this brother is added to the parents list and is removed from brothers list. However, the isolation decision is taken when both brothers list and parents list are empty.

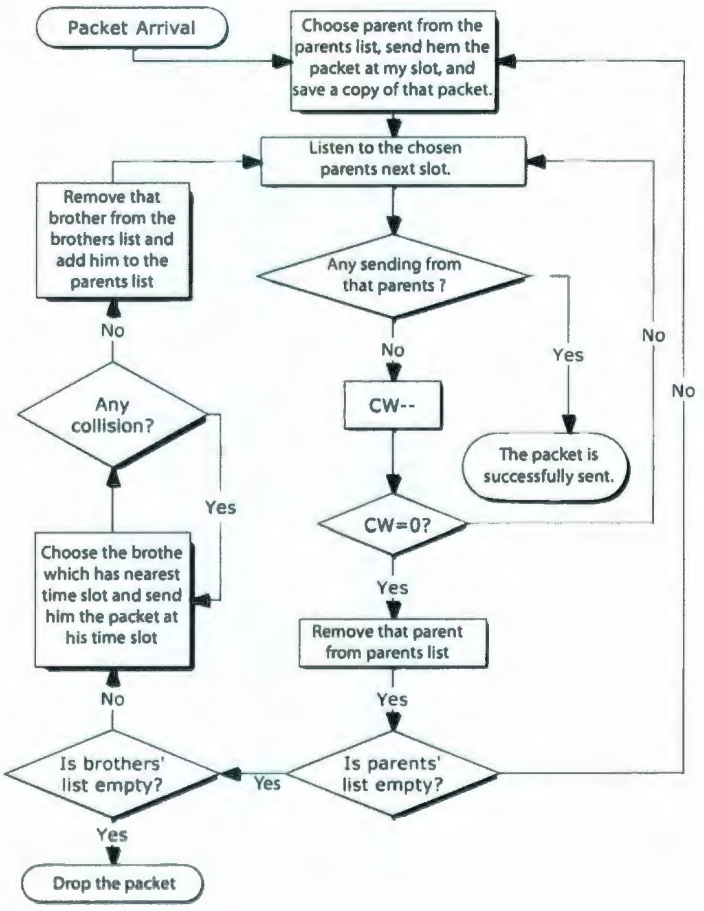


Figure 2.11: Flow chart of S-TDMA: Enhanced version.

Chapter 3

Reliability Analysis

As mentioned before, reliability is an important factor in fores fire detection to a void false fire reporting or miss fire detection. In this chapter, reliability is defined as a performance measuring metric. An analytical model is presented to evaluate the reliability for the proposed protocols.

3.1 Reliability definition

Generally, the term *reliability* in communication systems represents the ability of the system to transport a message from its source up to its destination. In forest fire detection based on WSNs and other wireless systems, the reliability concept goes beyond just measuring the ability of data transportation. It is extended to include the accuracy of the data itself, which is generated by the sensor nodes in the system. Therefore, two different terms of reliability are commonly used in WSNs [27][14]. The first is *data transport reliability*, which measure the system's ability to transport data. The second is *data sensing reliability* which measures the accuracy of the generated data. *Data sensing reliability* can be enhanced by different ways, such as using high quality transducers, implementing data aggregation, or increasing the number of sensors that monitors the same area spot. However, *data sensing reliability* is out of the scope of this thesis, while we focus on the

data transport reliability. For simplicity, from now on, the term *reliability* will stand for the *data transport reliability.*

Forest fire detection requires high *reliability* because any fire miss-reporting could result in a natural disaster. Also, false alarm warnings can waste money and efforts. Moreover, sensor nodes have high failure probability for this application due to the harshness of the environment. This implies the necessity of implementing fault management methods.

Different definitions of the reliability are proposed in the literature. The term event-to-sink reliability is proposed in [28], where the reliability is defined by the number of transported packets to the sink during a *decision interval.* The authors in [27] propose a different reliability definition which is the probability that at least one operational node in the cluster is still connected to the sink with an operational link. Motivated by these definitions, *we consider the reliability in this thesis as the percentage of received packets out of the total number of generated packets by all nodes.*

3.2 Data transport reliability of the presented MAC protocols

As explained earlier, the proposed protocols have a lack in reliability performance. This lack may be counted for the harsh nature of the forest fire detection environment, where nodes may have high failure rates. Therefore, many routes could break down and at the worst case some nodes might end up with no working route and get isolated from the network. In order to overcome this issue, enhanced versions of our protocols are proposed, where route maintenance mechanisms are implemented to support an adaptive routing approach. Both, the basic and enhanced versions of each protocol are explained in the previous chapter (System Model and MAC Protocols Design).

In this chapter, an analytical model is developed to evaluate the reliability performance

of the three MAC protocols for both the basic and the enhanced version. Figure 3.1 shows the described topology, where CH is located at the center of the cluster. The total cluster area T is composed of the two ring areas B and C plus the circle area A (i.e., $T = A \cup B \cup C$). T contains the complete set of nodes $\mathbf{N}$ randomly and uniformly distributed. Note that the general nodes n_j and n_v, where $(n_j, n_v) \in \mathbf{N}$, have communication ranges J and V, respectively. Considering a possible node failure, a node is in failure state (*dead*) with a probability P_f or in working state with a complementary probability P_w. Therefore, if a node is in working state, then it can be either isolated (*iso*) or connected and reach CH in 1, 2, 3 or 4 hops. We call these possibilities *States* (S) where $S = \{1, 2, 3, 4, iso, dead\}$. Here are some notations used in the rest of this chapter:

- H: the cardinality of the set $\mathbf{H}$, where $\mathbf{H}$ is a variable and can be any subset of $\mathbf{N}$ (i.e., $\mathbf{H} \subseteq \mathbf{N}$).
- $\mathbf{N^x}$: the set of nodes located in the area x, where x is a variable and could be any area, (e.g., A, B or C).
- $\mathbf{N_s}$: the set of nodes in the state s, where $s \in S$.
- $\mathbf{N_s^x}$: the set of nodes in the area x and in the state s.
- $\widehat{\mathbf{H}}$: the original set of $\mathbf{H}$, where $\mathbf{H}$ is a variable and could be any subset of $\mathbf{N}$ (i.e., $\mathbf{H} \subseteq \mathbf{N}$). 'Original' means exactly after creating the routing table and before the network operation where no nodes have failed yet.
- $\mathbf{N_{s,iso}}$: a set of nodes out of $\widehat{\mathbf{N}_\mathbf{s}}$. These nodes are not originally isolated but become isolated because of node failures.
- $\overline{x}$: the complement area of x (i.e., $\overline{x} = T - x$).
- $\beta_{k,p}^{M}$: the binomial probability of K successes out of M trials with p probability of success (i.e., $\beta_{k,p}^{M} = \binom{M}{K} p^K (1-p)^{M-K}$).

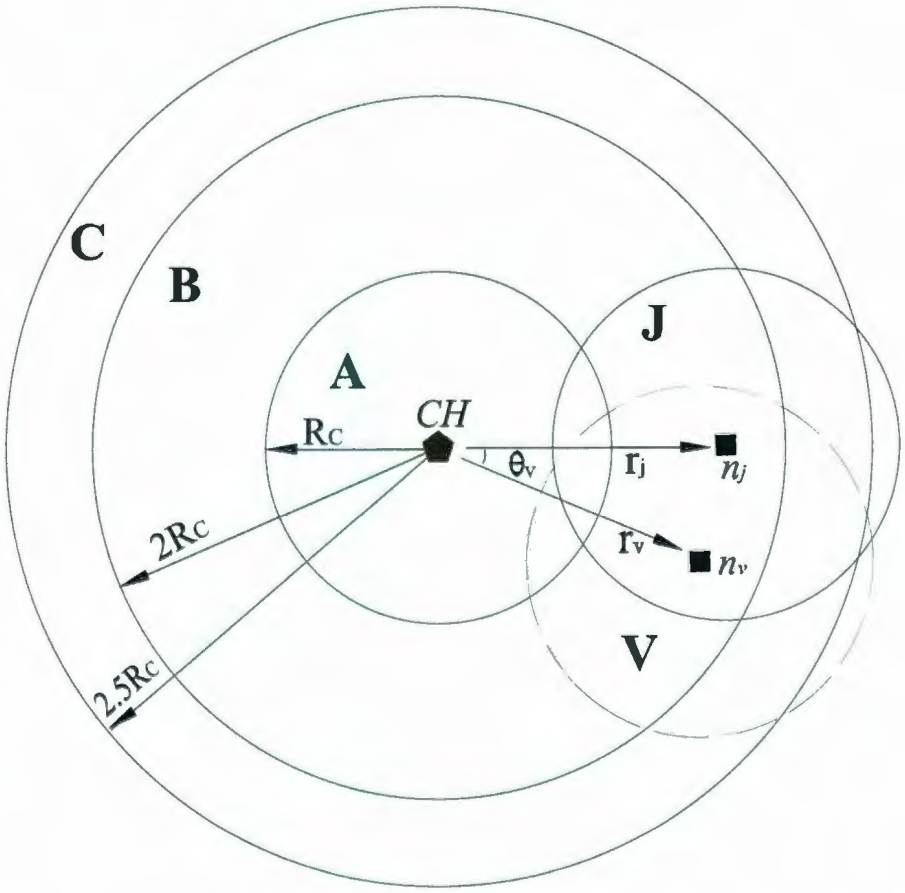


Figure 3.1: Cluster topology.

3.2.1 Enhanced version (With route maintenance)

As mentioned before, the route maintenance mechanisms enable nodes to find alternative routes in case of route failure. This statement is valid as long as there is a connected node with a level less than 4 in the neighborhood of the node that needs to recover from route failure. This means that the node can join this neighbor and reach CH in 4 hops or less. Otherwise, the node will be isolated and all of its generated packets will be lost. Going back to our definition of the reliability, which is the percentage of delivered packets out of the generated ones, it can also be represented by the percentage of connected nodes out of the working nodes since the generated traffic rates by nodes are the same. Starting from this point to calculate the reliability, we need to calculate the number of nodes that can reach CH in one (N_1), two (N_2), three (N_3) and four hops (N_4) first. Then, the reliability can be found as

$$Reliability = \frac{N_1 + N_2 + N_3 + N_4}{P_w \times N} \tag{3.1}$$

Since all nodes are uniformly distributed, the number of nodes in areas A, B and C are given by

$$N^A = N\frac{A}{T}, \quad N^B = N\frac{B}{T}, \quad N^C = N\frac{C}{T}. \tag{3.2}$$

From Figure 3.1, N_1, N_2, N_3 and N_4 are expressed as

$$\begin{aligned} N_1 &= N_1^A, \\ N_2 &= N_2^B, \\ N_3 &= N_3^B + N_3^C, \\ N_4 &= N_4^B + N_4^C. \end{aligned} \tag{3.3}$$

From (3.3), it can be shown that

$$N_1 = P_w \times N^A. \tag{3.4}$$

N_2 can be calculated as follows. In Figure 3.1 if n_j is located in the area B, then the probability that n_j reaches CH in two hops ($P_2^B \equiv Pr(n_j \in \mathbf{N_2} | n_j \in \mathbf{N^B})$) implies that n_j

is not dead and there is at least one connected node located in area $A \cap J$. The connected node has to be in $A \cap J$ in order to reach CH in one hop. Thus, P_2^B can be given by

$$P_2^B = P_w \int_1^2 \left(1 - \frac{\beta_{0,P_\chi}^N}{1 - \beta_{N,P_\chi}^N}\right) f_R(r_j)\, dr_j, \tag{3.5}$$

where P_χ is the probability that one working node is located in the area $A \cap J$. This node is definitely connected since it is in the communication range R_c of CH. The function $f_R(.)$ is the probability density function (*pdf*) of the radius, where nodes are uniformly randomly distributed over the radius $0 \rightarrow 2.5R_c$. Thus P_χ and f_R can be expressed as

$$P_\chi = P_w \frac{A \cap J}{T}. \tag{3.6}$$

$$f_R(r) = \frac{2r}{2.5R_c}. \tag{3.7}$$

In (3.5), the term (β_{0,P_χ}^N) represents the probability that the area $A \cap J$ has no connected nodes. Since we know that n_j is already out of $A \cap J$, it is impossible that all $\mathbf{N}$ is inside $A \cap J$. Therefore, the term (β_{0,P_χ}^N) is divided by $(1 - \beta_{N,P_\chi}^N)$. In the same equation, the term between brackets represents the probability that at least one connected node is located in the area $A \cap J$ at a specific r_j, while the integration is to average that probability over the area B. Therefore, N_2 can be easily calculated, where $N_2 = P_2^B \times N^B$.

N_3 can be calculated by determining N_3^B and N_3^C. For N_3^B, in Figure 3.1 if n_j is located in area B, the probability that n_j reaches CH in three hops ($P_3^B \equiv Pr(n_j \in \mathbf{N_3} | n_j \in \mathbf{N^B})$) implies that the area $A \cap J$ has no connected nodes and there is at least one connected node which reaches CH in two hops in the neighborhood. Therefore, the following probabilities are required to be calculated:

- $P_{3,I} \equiv Pr(\mathbf{N_1^{A \cap J}} = \mathbf{\Phi})$: the probability that the area $A \cap J$ has no connected nodes.
- $P_{3,II} = 1 - \sum_{n_v \in \mathbf{N^{B \cap J}}} Pr(n_v \notin \mathbf{N_2} | \mathbf{N_1^{A \cap J}} = \mathbf{\Phi})$: the probability that there is at least one connected node like n_v, which can reach CH in two hops given that the area $A \cap J$ has no connected nodes.

Hence, P_3^B can be given by

$$P_3^B = P_w \int_{R_c}^{2R_c} (P_{3,I} \times P_{3,II})\, f_R(r_j) dr_j, \tag{3.8}$$

where $P_{3,I}$ and $P_{3,II}$ can be expressed as

$$P_{3,I} = \frac{\beta_{0,P_\chi}^N}{1 - \beta_{N,P_\chi}^N}, \tag{3.9}$$

$$P_{3,II} = 1 - \frac{\beta_{0,P_o}^N}{1 - \beta_{N,P_o}^N}, \tag{3.10}$$

where P_o represents the probability that a node like n_v which is located in the area $B \cap J$ is connected and reaches CH in two hops given that the area $A \cap J$ has no connected nodes. This probability can be obtained as follows:

$$P_0 = \iint_{B \cap J} \left(1 - \frac{\beta_{0,P_\eta}^N}{1 - \sum_{e_0} \beta_{e_0,P_\eta}^N}\right) f_R(r_v) f_\theta d\theta_v dr_v, \tag{3.11}$$

where $e_0 \in \{N-1, N\}$, P_η is the probability that a node is not dead and located in the area $A \cap \overline{J} \cap V$ and f_θ is the *pdf* of θ. P_η and f_θ are given by

$$P_\eta = P_w \frac{A \cap \overline{J} \cap V}{T}, \tag{3.12}$$

$$f_\theta = \frac{1}{\theta_U - \theta_L}, \tag{3.13}$$

where θ_U and θ_L are the upper and lower limits of θ in the area $B \cap J$, respectively. These limits are functions of r_v and r_j, and can be given by

$$\theta_U = cos^{-1}\left(\frac{r_j^2 + r_v^2 - 1}{2 \times r_j \times r_v}\right). \tag{3.14}$$

$$\theta_L = 2\pi - \theta_U. \tag{3.15}$$

Now N_3^B is ready to be calculated, where $N_3^B = P_3^B \times N^B$.

Similarly, we can find N_3^C. Therefore, $P_3^C \equiv Pr((n_j \in \mathbf{N_3})|(n_j \in \mathbf{N^C}))$ can be calculated as

$$P_3^C = P_w \int_{2R_c}^{2.5R_c} P_{3,II} f_R(r_j) dr_j, \tag{3.16}$$

Equation (3.16) is quit similar to equation (3.8) except that the term $P_{3,I}$ is omitted since $A \cap J = 0$. N_3^C is ready to be calculated, where $N_3^C = P_3^C \times N^B$. After finding N_3^B and N_3^C, N_3 can be calculated using (3.3).

N_4 can be determined from N_4^B and N_4^C. For N_4^B, if n_j in Figure 3.1 is located in the area B, the probability that n_j reaches CH in four hops ($P_4^B \equiv Pr(n_j \in \mathbf{N_4}|n_j \in \mathbf{N^B})$) implies that the node n_j itself can not reach CH in two or three hops and there is at least one connected node which reaches CH in three hops in the neighborhood. Therefore, the following probabilities are required:

- $P_{\notin N_{23}^B} = Pr(n_j \notin (\mathbf{N_2^B} \cup \mathbf{N_3^B})|n_j \in \mathbf{N^B})$: the probability that n_j can not reach CH in 2 or 3 hops given that it is located in the area B.
- $P_{in} = 1 - \sum_{n_v \in \mathbf{N}^{\overline{A} \cap J}} Pr(n_v \notin \mathbf{N_3}|\mathbf{N_2^{B \cap J}} = \mathbf{\Phi})$: the probability that there is at least one node like n_v which can reach CH in three hops given that the area $B \cap J$ is empty of $\mathbf{N_2}$ nodes.

Thus, P_4^B can be expressed as

$$P_4^B = P_w \times P_{\notin N_{23}^B} \int_{R_c}^{2R_c} P_{in} \times f_R(r_j) dr_j, \tag{3.17}$$

where $P_{\notin N_{23}^B}$ is equal to $1 - P_2^B - P_3^B$, while P_{in} can be obtained by

$$P_{in} = \sum_{n_{in}=2}^{N-3} \left(\frac{\beta^N_{n_{in}, \frac{\overline{A} \cap J}{T}}}{1 - \sum_{e_1} \beta^N_{e_1, \frac{\overline{A} \cap J}{T}}} \times P_{inb} \right), \tag{3.18}$$

where $e_1 \in \{0\}$[1]. P_{inb} is the probability that there is in the neighborhood at least one node like n_v, which reaches CH in three hops given the number of nodes in the area $\overline{A} \cap J$ (n_{in}) and given that the area $B \cap J$ is empty of $\mathbf{N_2}$ nodes. The summation in this equation is for averaging P_{inb} over n_{in}. From the definition given above, P_{inb} is expressed as

$$P_{inb} = \sum_{n_{inb}=1}^{n_{in}} \left(\frac{\beta^{n_{in}}_{n_{inb}, \frac{B \cap J}{A \cap J}}}{1 - \sum_{e_2} \beta^{n_{in}}_{e_2, \frac{B \cap J}{A \cap J}}} \times P_p \right), \tag{3.19}$$

where $e_2 \in \{0\}$[1], P_p represents the probability that there is at least one connected node that reaches CH in three hops in the area $\overline{A} \cap J \equiv (B \cap J) \cup (C \cap J)$ given n_{in}, the number of nodes in $B \cap J$ (n_{inb}) and that the area $B \cap J$ is empty of $\mathbf{N_2}$ nodes. The main summation in this equation is for averaging P_p over n_{inb}. From the definition given above, P_p is found from

$$P_p = 1 - \beta^{U_1}_{0, P_\iota(B \cap J)} \times \beta^{W_1}_{0, P_\iota(C \cap J)}, \tag{3.20}$$

where $U_1 = n_{inb} - 1$ and $W_1 = n_{in} - n_{inb}$, while $P_\iota(B \cap J)$ and $P_\iota(C \cap J)$ are the probabilities that given a node is located in the area $B \cap J$ and $C \cap J$, respectively; then this node reaches CH in three hops. $P_\iota(x)$, where x is a variable and could be any area in $\overline{A}$, can be expressed as

$$P_\iota(x) = P_w \iint_x (1 - \beta^{N_1}_{0, \frac{B \cap \overline{J} \cap V}{B}}) \times f_R(r_j) f_{\theta_v} d\theta_v dr_v. \tag{3.21}$$

Accordingly, N_4^B is ready to be calculated as $N_4^B = P_4^B \times N^B$. In order to calculate N_4^C, $P_4^C \equiv Pr(n_j \in \mathbf{N_4} | n_j \in \mathbf{N^C})$ is needed. Similarly as for N_4^B, P_4^C can be expressed as

$$P_4^C = P_w \times P_{\notin N_3^C} \times \int_{2R_c}^{2.5R_c} P_{in} \times f_R(r_j) dr_j. \tag{3.22}$$

[1]This set will contain more than one member in the next section.

In this equation, $P_{\notin N_3^C} = 1 - P_3^C$, P_{in} is given by (3.18) taking into account that in (3.19), $e_2 \in \{n_{in}\}$, and in (3.20), $U_1 = n_{inb}, W_1 = n_{in} - n_{inb} - 1$.

N_4^C is ready to be calculated as $N_4^C P_4^C \times N^C$. Thus, N^4 can be found from (3.3). Finlay, the reliability is ready to be calculated as well from (3.1).

3.2.2 Basic version (without route maintenance)

Without implementing route recovery mechanisms, a node becomes isolated when its parents are either dead or isolated. Thus, the probability of a node being isolated is larger compared with the case of implementing route maintenance mechanisms, where the node can use other neighbors to relay its traffic if all of its parents are dead. We need to calculate the number of connected nodes to find the reliability as in (3.1). The procedure to find the number of connected nodes is as follows.

- Find $\widehat{N_s^x}$ where $x \in \{A, B, C\}$ and $s \in \{1, 2, 3, 4, iso, dead\}$. These numbers can be found exactly with the same way as in the previous analysis considering $P_w = 1$.
- Find $N_{\ell,iso}$, where $\ell \in \{1, 2, 3, 4\}$. These values have to be found sequentially starting from $\ell = 1$ and ending with $\ell = 4$ because the probability of a connected node in some level to get isolated depends on the probability of isolation and failure of its parents, which have a one degree lower level.

Obviously, it is impossible for any node that has a direct link with the CH to become isolated; it is either dead or connected. Therefore, $N_{1,iso} = 0$.

For N_2, we need to find $P_{2,iso}$ which is the probability that a node like n_j gets isolated if it originally reaches CH in two hops (i.e., $P_{2,iso} = Pr(n_j \in \mathbf{N_{2,iso}} | n_j \in \widehat{\mathbf{N_2}})$). This probability can be expressed as the probability that all node's parents are either dead or isolated and can be given by

$$P_{2,iso} = \int_{R_c}^{2R_c} \left(\sum_{n=1}^{N-1} (P_f)^n \frac{\beta_{n,P_X}^N}{1 - \sum_{e_3} \beta_{e_3,P_X}^N} \right) f_R(r_j) dr_j. \tag{3.23}$$

where $e_3 \in \{0, N\}$. N_2 can be calculated as

$$N_2 = \widehat{N_2} \times (1 - P_{2,iso}). \tag{3.24}$$

For N_3, $P^B_{3,iso}$ and $P^C_{3,iso}$ should be calculated first, where $P^B_{3,iso} = Pr(n_j \in \mathbf{N_{3,iso}} | n_j \in \widehat{\mathbf{N_3^B}})$ and $P^C_{3,iso} = Pr(n_j \in \mathbf{N_{3,iso}} | n_j \in \widehat{\mathbf{N_3^C}})$. $P^B_{3,iso}$, is expressed as

$$P^B_{3,iso} = \int_{R_c}^{2R_c} \left(\sum_{n=1}^{N-1} (P_f)^n \frac{\beta^N_{n,P_o}}{1 - \sum_{e_4} \beta^N_{e_4,P_o}} \right) f_R(r_j) dr_j. \tag{3.25}$$

where $e_4 \in \{0, N-1, N\}$. Heading to the next step, $P^C_{3,iso}$ can be obtained by

$$P^C_{3,iso} = \int_{2R_c}^{2.5R_c} \left(\sum_{n=1}^{N-1} (P_f)^n \frac{\beta^N_{n,P_o}}{1 - \sum_{e_5} \beta^N_{e_5,P_o}} \right) f_R(r_j) dr_j. \tag{3.26}$$

where $e_5 \in \{0, N-1, N\}$. Then, N_3 can be calculated as

$$N_3 = \widehat{N_3} - P^B_{3,iso} \times \widehat{N_3^B} - P^C_{3,iso} \times \widehat{N_3^C}. \tag{3.27}$$

Similarly for N_4, $P^B_{4,iso}$ and $P^C_{4,iso}$ are needed, where $P^B_{4,iso} \equiv Pr(n_j \in \mathbf{N_{4,iso}} | n_j \in \widehat{\mathbf{N_4^B}})$ and $P^C_{4,iso} \equiv Pr(n_j \in \mathbf{N_{4,iso}} | n_j \in \widehat{\mathbf{N_4^C}})$. $P^B_{4,iso}$ can be expressed as

$$P^B_{4,iso} = \int_{R_c}^{2R_c} P'_{in} \times f_R(r_j) dr_j. \tag{3.28}$$

where $P'_{in}$is given by the same equation of P_{in} given by (3.18) except that $e1 \in \{0, 1, N-2, N-1, N\}$, and P_{inb} is replaced by P'_{inb}, where P'_{inb} is similar to P_{inb} given by (3.19) except that P_b is replaced by P'_b, while P'_b is given by

$$P'_p = \sum_{u=0}^{U_2} \sum_{w=0}^{W_2} \beta^{U_2}_{u,P_i(B \cap J)} \beta^{W_2}_{w,P_i(C \cap J)} (P_f)^{u+w} - \beta^{U_2}_{0,P_i(B \cap J)} \beta^{W_2}_{0,P_i(C \cap J)}. \tag{3.29}$$

where $U_2 = n_{inb-1}$ and $W_2 = n_{in} - n_{inb}$. Similarly, $P^C_{4,iso}$ can be found from

$$P_{4,iso}^{C} = \int_{2R_c}^{2.5R_c} P_{in}'' \times f_R(r_j) dr_j \tag{3.30}$$

where P_{in}'' is similar to P_{in} given by (3.18) except that $e_1 \in \{0, 1, N-2, N-1, N\}$, and P_{inb} is replaced by P_{inb}'', where P_{inb}'' is similar to P_{inb} given by (3.19) except that $e_2 \in \{n_{in}\}$ and P_b is replaced by P_b'', where P_b'' is similar to P_b' given by (3.29) except that $U_2 = n_{inb}$ and $W_2 = n_{in} - n_{inb} - 1$.

Accordingly, N_4 can be determined as

$$N_4 = \widehat{N_4} - P_{4,iso}^{B} \times \widehat{N_4^B} - P_{4,iso}^{C} \times \widehat{N_4^C}. \tag{3.31}$$

After finding the numbers of connected nodes with different levels, the reliability can be calculated using (3.1).

Chapter 4

Results

In this chapter node density and the waiting window design parameters are discussed and optimized. Moreover, the proposed MAC protocols are compared comprehensively by applying both regular and emergency transmission. The performance metrics considered are the average power consumption per node, packet delay, and system reliability. An added metric is considered just in the case of emergency reporting. This metric is called 'event to cluster head delay'.

As mentioned in the system model that the clusters have a circular shape. In this work, the cluster's radius (R_{cl}) is considered to be 2.5 km. This number is chosen based on the assumption that the total area of the forest under surveillance is 200 km^2. Thus, this area can be divided into 100 clusters with an area of 20 km^2. Considering a circular shape for the clusters, the radius is about 2.5 km.

Sensor nodes features are assumed to be similar to those of TinyNode 584 sensor nodes [29]. For this model, the current dissipation values in the transmitting (I_t), receiving(I_t) and sleep (I_s)modes are 46 mA, 16 mA and 6.5 μA, respectively. The data ratef_b is 1.2 kbps, which is the lowest available data rate for this model. The lowest data rate is chosen to support the highest transmission distance(R_c), which is 1 km. Each sensor node generates an information message every $T_{wi} = 30$ minutes. Message length (L) is considered to be 15 bytes long (i.e., 100 ms long), and the message contains information about tem-

perature, humidity, wind speed, Cyclic Redundancy Check (CRC), sender address, and next node address. These messages are generated regularly and called *Regular checkup*. *Emergency Reporting* mode, in which a message is generated once a fire is detected to report that event, is supported by our MAC protocols and considered in our analysis.

4.1 Node connectivity and node density

As emphasized earlier, cost is an important aspect in forest fire detection because of the huge area to be covered. Therefore, the number of the nodes represented by the node density should me minimized. However, this optimization problem is constrained with the condition that any node should be connected to the network (i.e., any node should have at least one route to the *CH*). In this thesis the term *connectivity* of a node will be used to represent the number of *vertex disjoint* paths, in which that node can reach the *CH*. In graph theory, any two paths are called *vertex disjoint* when there is no common node between them. Since nodes have different *connectivities*, three metrics can be used to represent the robustness of the network. These metrics are the *MinConnectivity*, *AvgConnectivity*, and *MaxConnectivity*. The *MinConnectivity* is the connectivity of the nod which has the lowest connectivity among all nodes in the network. This figure is a good indicator of the robustness but it is not the best since just one node is used to find this metric. However, *MinConnectivity* is important in choosing a minimum node density, which guaranties no isolated node in the network. Such a density is found at *MinConnectivity* = 1. The second, *AvgConnectivity*, is more meaningful in representing the robustness of the network since all nodes participate in calculating this metric. The last, *MaxConnectivity*, is not very usefully in the sense that it is calculated based on just one entry, which is the connectivity of the node which has maximum connectivity among the networks. Moreover, there are no restrictions on the maximum number of paths. However, this metric can be used to give a better understanding about the network.

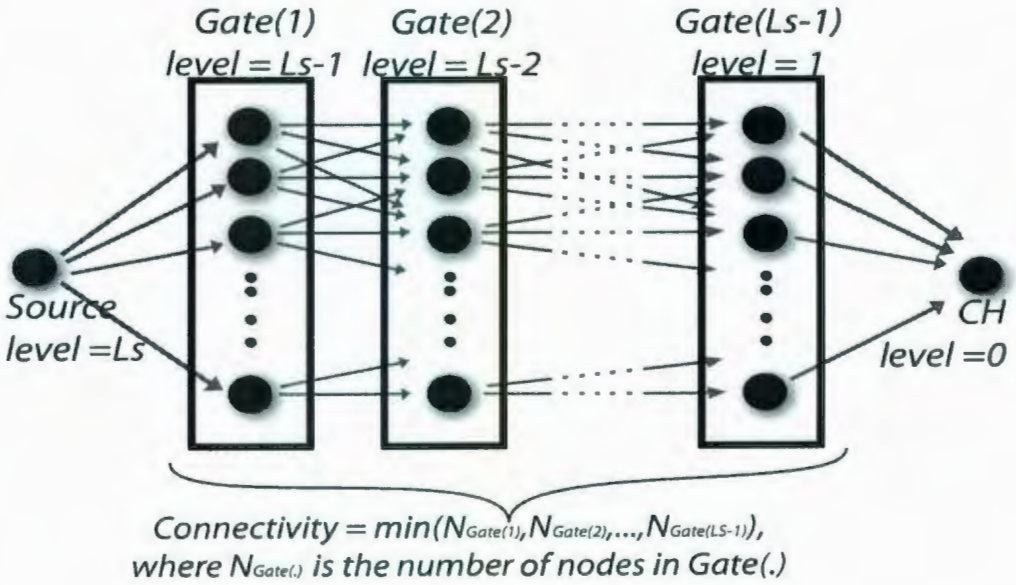


Figure 4.1: Connectivity model.

Excluding the possibility of node failure, *MinConnectivity*, *AvgConnectivity*, and *MaxConnectivity* are evaluated (by a C++ simulation) for different values of node density. One simulation approach to find the connectivity of a certain node is to use graph theory. However, for the sake of simplicity, another approach is used. Figure 4.1 shows this approach, where all the intermediate nodes, that could relay packets from the source up to the *CH*, are divided into *Gates* based on their levels. In other words, the intermediate nodes, which have the same level, comprise one *Gate*. The size of a *Gate* is the number of the nodes N_{Gate} that comprises this *Gate*. Under this vision, when a source node sends a packet, the packet has to pass all the *gates*, which have different sizes, one by one until reaching the *CH*. The packet can pass through a certain *Gate* using any of the nodes inside that *Gate*. Since we have defined the *connectivity* as the number of *vertex disjoint* paths, the number of paths through any *Gate* equals the size of that *Gate* (N_{Gate}). The *connectivity* of a certain node then is the minimum gate size of all gates.

For the presented system model and following the presented approach, Figure 4.2 shows *MinConnectivity*, *AvgConnectivity*, and *MaxConnectivity* at different node densi-

ties. Clearly, at almost 5 $node/\text{km}^2$ density, the *MinConnectivity* hits 1 connectivity. Therefore, this value could be considered as the optimum node density under the constraint that there are no isolated nodes.

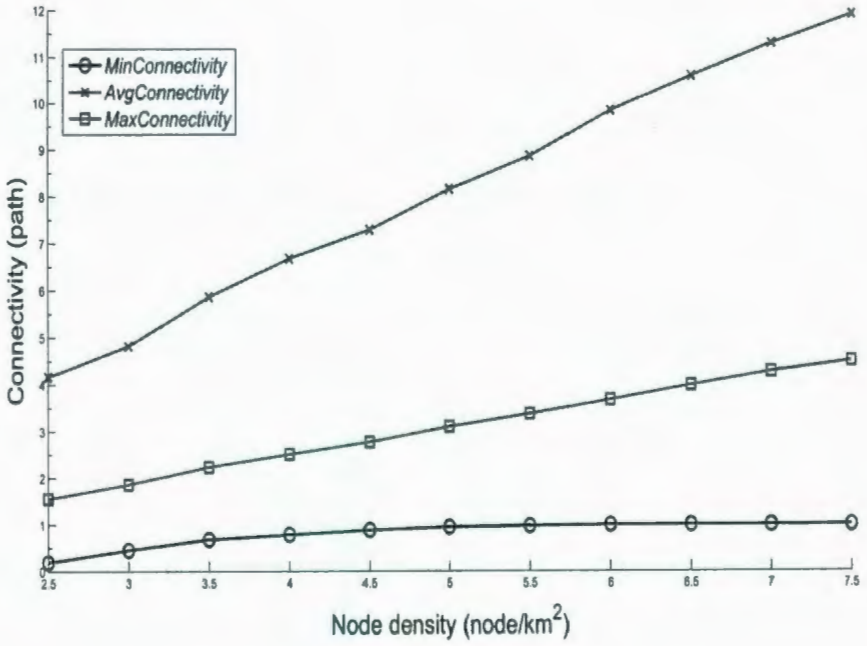


Figure 4.2: Simulation results of the connectivity model in Figure 4.1.

In [30], the author proves a relationship between the transmission range of sensor nodes and network connectivity. This relationship is given by

$$r_0 \geq \sqrt{\frac{-ln(1-p^{1/n})}{\rho\pi}} \tag{4.1}$$

where ρ is the node density, n is the total number of nodes in the WSN (in the cluster as we consider clustered topology), and p is the probability that the network is connected. For the network, being connected means any two nodes can reach each other directly or with multi-hop connection. Figure 4.3 shows this relationship. Obviously from the figure, to achieve a connection probability of 1 with 5 $node/\text{km}^2$ node density, the minimum communication range required is 1 km, which is employed in our model. This confirms

the results in Figure 4.2.

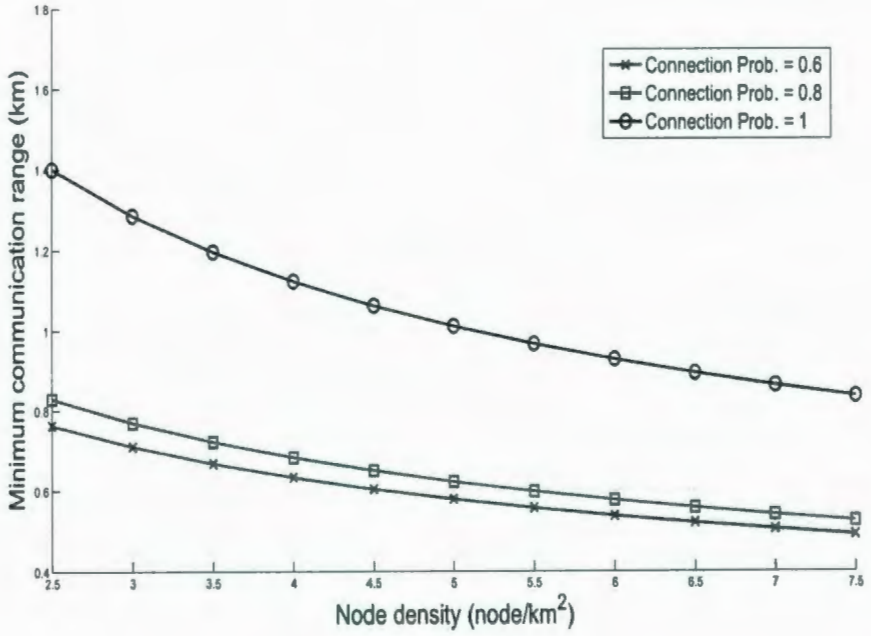


Figure 4.3: Results from equation (4.1).

In [31], the authors prove another relationship, which calculates the probability (f_a) of a spot to be covered with at least one sensor node given the node density (λ) and the sensing range (r_s). This equation is given by

$$f_a = 1 - e^{-\lambda \pi r_s^2} \tag{4.2}$$

Figure 4.4 shows equation (4.2) for different values of the sensing range. Clearly the coverage probability is more affected by the sensing range rather than the node density. However, the coverage probability is not as critical as the connection probability. This can be explained as follows. If a fire is started in an area, which is not covered by sensor nodes, this fire will be detected after a period of time as the fire moves into the coverage area of one of the surrounding sensors. Therefore, the fire will be detected after some delay.

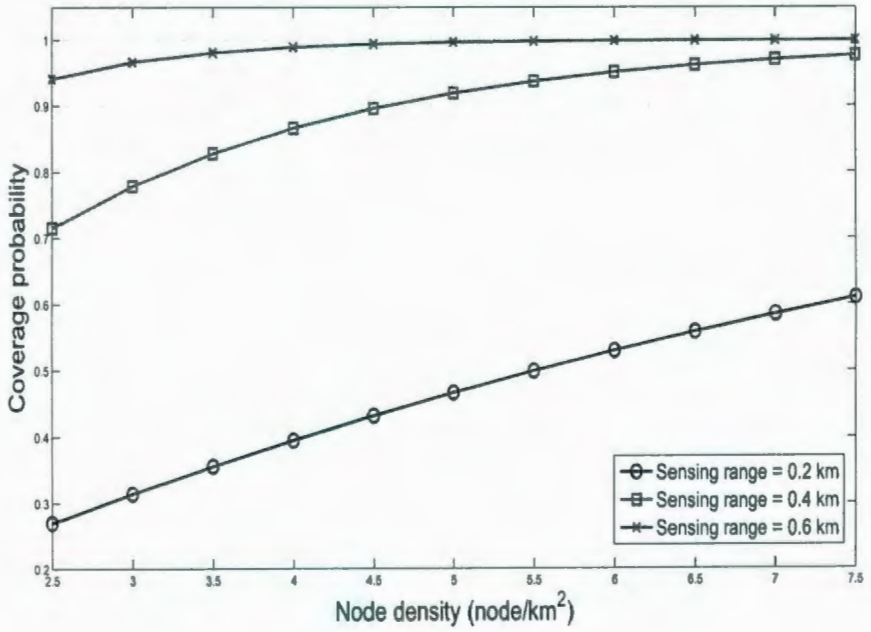


Figure 4.4: Results from equation (4.2).

4.2 Waiting window analysis

As explained earlier in the design of the proposed protocols, P-CSMA and PHS-CSMA have a design parameter called the *waiting window* which is the period of time during which the source node keeps trying to transmit its packet before it gives up, drops off the packet, and considers isolation from the network in the basic version or initiates route maintenance in the enhanced version. Clearly, a node can make more attempts to send the packet in longer values of *waiting window*. This parameter is a design factor which has to be chosen carefully since small values of this parameter may lead to an unnecessary decision of isolation or rerouting. Therefore, the reliability of the system is affected by this factor. Obviously, larger values for this parameter lead to better reliability since increasing the number of trails of transmissions can reduce the probability of an incorrect isolation decision. However, reliability is not the only factor affected by this parameter.

Power consumption and packet delay are expected to vary by changing this parameter. Power consumption is proportional to *waiting window* since a large number of attempts to send the packet obviously causes more power consumption. The same trend is also valid for the packet delay as more attempts to send the packet makes the time elapsed to deliver the packet longer.

4.2.1 P-CSMA

In order to find the optimum *waiting window* value, power consumption,packet delay, and reliability are tested at different values of the *waiting window*. Simulation results in Figures 4.5, 4.6, and 4.7 show that power consumption, packet delay, and reliability in P-CSMA increases as the *waiting window* increases (as expected). Clearly in Figure 4.7, the relationship is not linear and after a certain value of *waiting window*, which is $40 \times T_{wi}$ the reliability increases very slightly. This can be explained in that, after this value is reached almost no wrong decision of isolation is taken by nodes.

Power consumption and packet delay have the same trends as the reliability in their dependence on the *waiting time*. Figure 4.5 shows that power consumption saturates at *waiting time* of $(40 \times T_{wi})$. This is due to the fact that, only the nodes which are really isolated need longer *waiting times* than this value to transmit. Therefore, the power consumed by those nodes is basically consumed by isolated nodes. Since isolated nodes are not included in the power performance calculation, a slight increase in the average power consumption is expected after this point. Figure 4.6 shows that almost no more delay is added to the system performance after this point ($waiting\ window = 40 \times T_{wi}$) because any packet does not need more waiting time than $40 \times T_{wi}$ to be delivered to the next node as long as the source node is not isolated. Based on the last discussion, $waiting\ window = 40 \times T_{wi}$ is the optimum point.

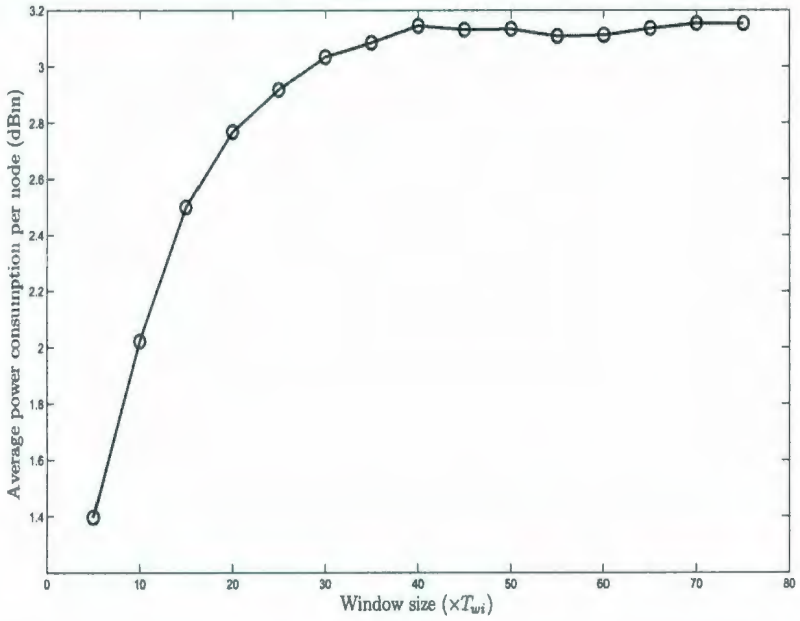


Figure 4.5: Dependence of power consumption of P-CSMA on *waiting window*, at N=100.

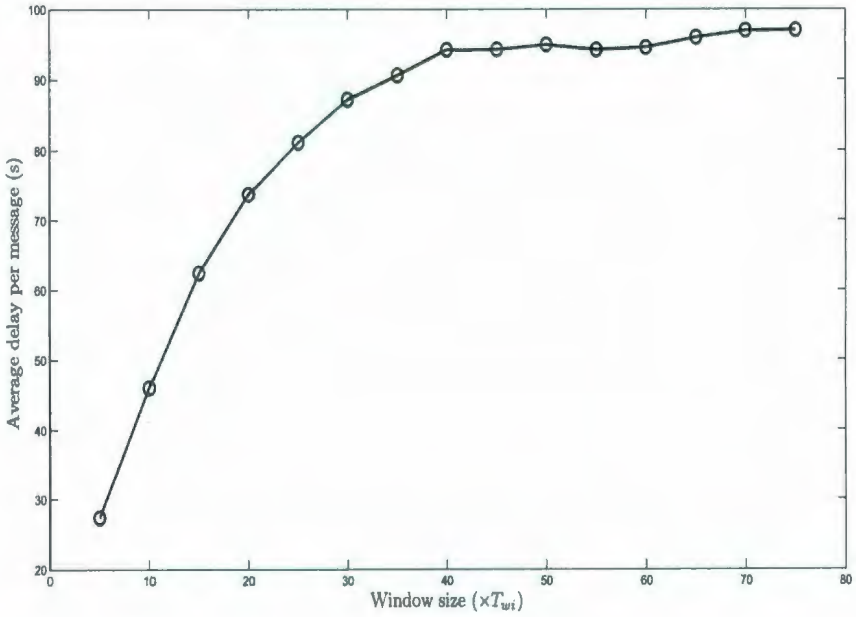


Figure 4.6: Dependence of power consumption of P-CSMA on *waiting window*, at N=100.

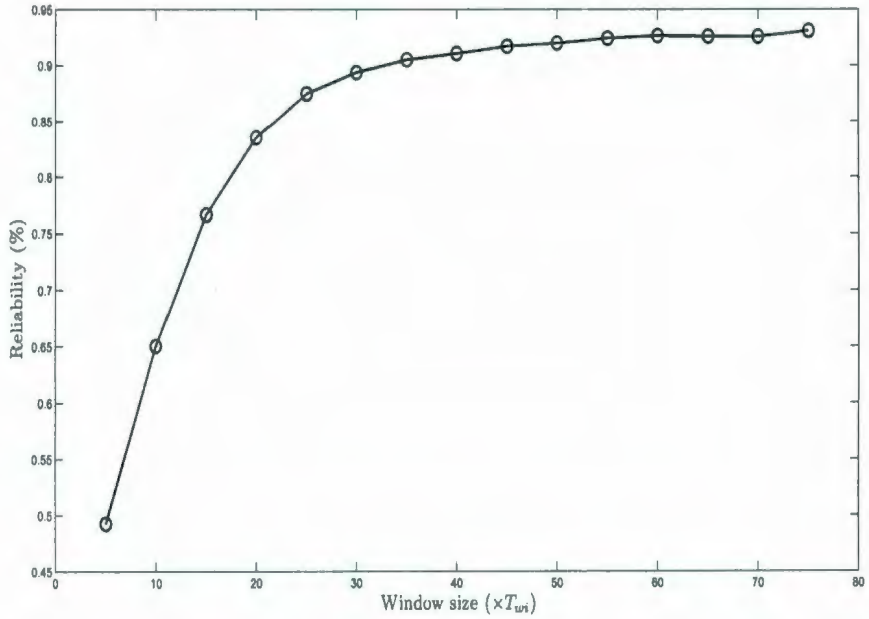


Figure 4.7: Dependence of power consumption of P-CSMA on *waiting window*, at N=100.

4.2.2 PHS-CSMA

Similarly as for P-CSMA, Figures 4.8, 4.9, and 4.10 show that the optimum *waiting window* value for PHS-CSMA is $8 \times T_{wi}$. Clearly, the optimum *waiting window* value for PHS-CSMA is much smaller than for P-CSMA. This is due to the higher collision rate in P-CSMA.

4.3 Performance comparison of the proposed MAC protocols

Using computer simulation implemented in C++, the proposed MAC protocols are analyzed. The results obtained for average power consumption per node, average packet delay, and system reliability versus the wake-up interval (T_{wi}) are used to compare the performance of the proposed protocols. Since the active time (T_{ac}) and the slot time (T_s)

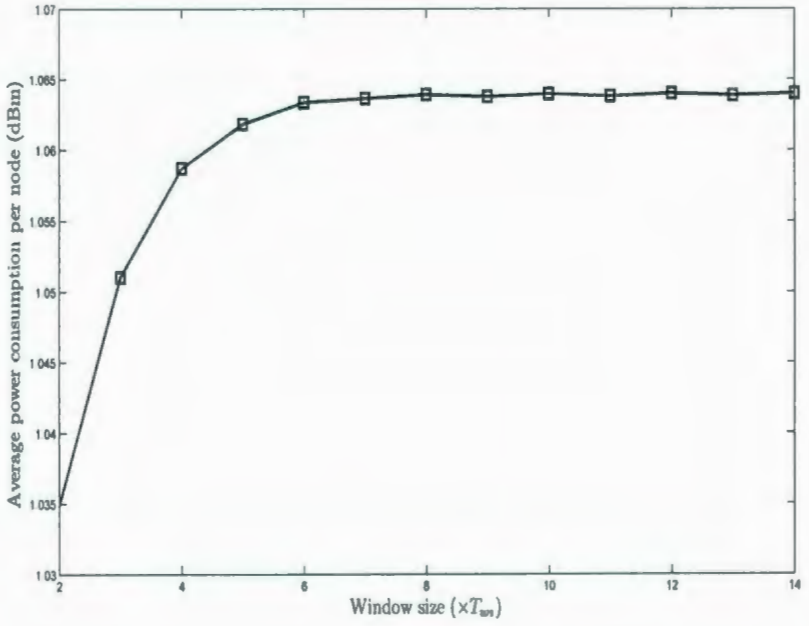


Figure 4.8: Dependence of power consumption of PHS-CSMA on *waiting window*, at N=100.

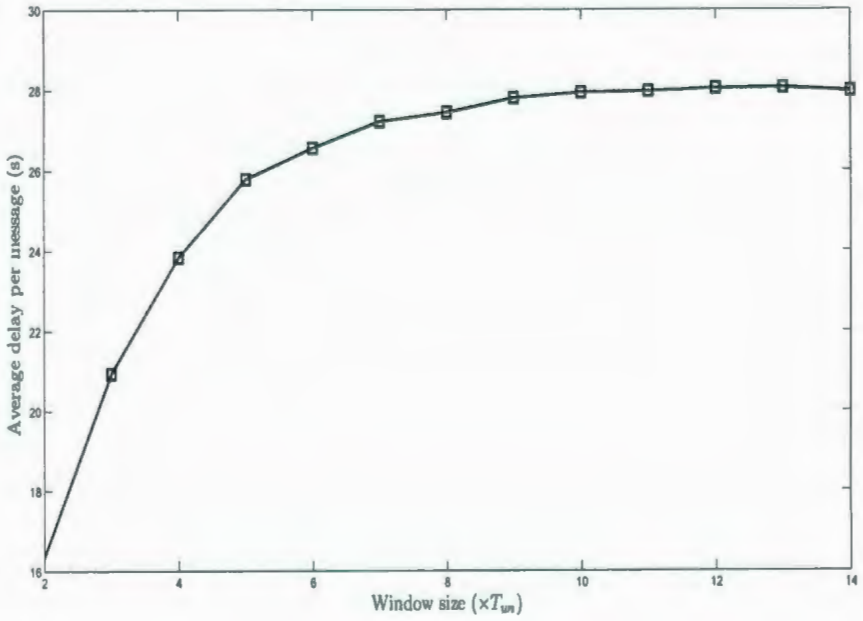


Figure 4.9: Dependence of packet delay of PHS-CSMA on *waiting window*, at N=100.

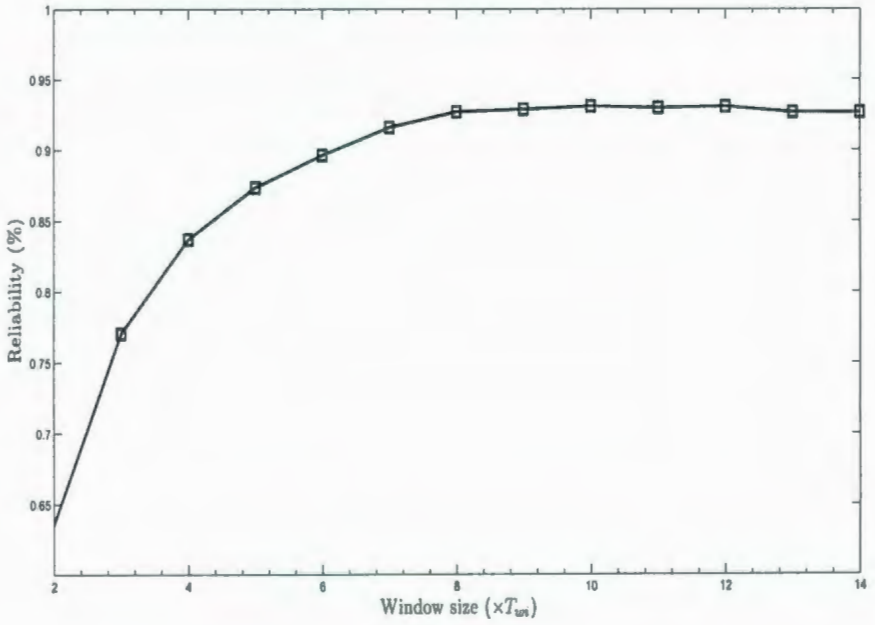


Figure 4.10: Dependence of reliability of PHS-CSMA on *waiting window*, at N=100.

are taken to be fixed values, the variable term is the sleep time (T_{sl}). Generally, the minimum power consumption (P_{min}) is achieved when nodes just wake up to transmit or receive a packet and when there is no possibility for collisions and retransmission. Achieving this P_{min} is not feasible in practice, but it is worthy to be calculated as a reference level. On the other hand, we can assume that the maximum power dissipation (P_{max}) is applied when nodes are all the time in active mode .

According to the analytical model with $N = 100$ and $P_f = 0$, the number of nodes that reach CH by 4 hops (N_4) is found to be 6, while the nodes that reach CH by 3 hops (N_3) is found to be 45. Also, N_2 is equal to 33 nodes, and N_1 is equal to 15 nodes. The total number of transmissions every 30 minutes is N_{tr}, where

$$N_{tr} = N_1 + N_2 \times 2 + N_3 \times 3 + N_4 \times 4 = 255.$$

For each transition, the time needed to transmit one information message is T_{msg}, where

$$T_{msg} = \frac{15\ bytes}{1.2\ kbps} = 0.1\ sec.$$

Since the TinyNode 584 [29] model is designed to work with 2 AA alkaline batteries, the operation voltage is 3 volts and the energy needed to send one information message and to receive one information message are $e_{s,msg}$ and $e_{r,msg}$ respectively, where

$$e_{s,msg} = T_{msg} \times 46\ mA \times 3\ V = 13.8\ mJ.$$

$$e_{r,msg} = T_{msg} \times 16\ mA \times 3\ V = 4.8\ mJ.$$

Consequently, the theoretical minimum total consumed energy ($E_{t,min}$) by all nodes during 30 minutes is

$$E_{t,min} = (e_{s,msg} + e_{r,msg}) \times N_{tr} + ((60 \times 30) \times N - 2 \times N_{tr} \times T_{msg}) \times 6.5\mu A \times 3V = 8.117J.$$

On the other hand, the theoretical maximum total consumed energy ($E_{t,max}$) by all nodes during 30 minutes is assumed when no sleep mode implemented. Control messages overhead is not taken into account in this calculation. $E_{t,max}$ is calculated as follows:

$$E_{t,min} = (e_{s,msg} + e_{r,msg}) \times N_{tr} + ((60 \times 30) \times N - 2 \times N_{tr} \times T_{msg}) \times 16mA \times 3V = 8642.3J.$$

Finally, the average minimum consumed power level per node is p_{min}, and the average maximum consumed power level per node is p_{max}, where

$$p_{min} = \frac{E_{t,min}}{N \times (30 \times 60)} = 45.09\mu W \equiv -13.46dBm.$$

$$p_{max} = \frac{E_{t,max}}{N \times (30 \times 60)} = 45.01mW \equiv 16.81dBm.$$

These boundaries give a rough indication about range of power consumption of MAC protocols for our system model presented earlier. However, in the calculations of the upper bound, collisions and retransmissions are not considered.

The proposed protocols are tested and analyzed in terms of power consumption,packet delay, and reliability. Simulation results are used for the comparison between the proposed protocols. Analytical results are used along with simulation results to test the reliability performance of the proposed protocols.

Figures 4.11 4.12 and 4.13 show the average power consumption per node, average packet delay, and reliability respectively, at different values of T_{wi}. In the figures, S-TDMA can not take values less that 10 s for T_{wi}. This is due to the fact that each node in the system has its own time slot in T_{ac}. Therefore, T_{ac} should be at least equals to $N \times T_s = 100 \times 0.1 = 10$ s.

In terms of power consumption, as shown in Figure 4.11, power consumption for all protocols decreases as T_{wi} increase. This trend is expected as longer T_{wi} means that a node stays more in the sleep mode, which saves power. However, power consumption of P-CSMA saturates after a certain value of T_{wi}, which is 10 s. This behavior may be explained in that for longer T_{wi} than 10 s, the power saved in longer sleeping time is consumed with more attempts to send or forward packets. On the other hand, this is not the case in the other two protocols, where in PHS-CSMA nodes know the schedules of neighbors and there is no need to keep trying to send while the next node is in sleep mode. Also, in TDMA, nodes are fully synchronized and know when the next node is awake to receive traffic. Therefore, it is expected that for PHS-CSMA and S-TDMA to be monitically decreasing versus T_{wi}.

Clearly, among the proposed protocols, P-CSMA has the worst performance in terms of power consumption, while S-TDMA outperforms P-CSMA and PHS-CSMA. This can be explained in that P-CSMA is a contention-based MAC protocol and has no type of time synchronization. Therefore, a node spends a longer time trying to communicate

with the next node before sending or forwarding a packet. Consequently, more collisions in the system are expected. Similarly, S-TDMA outperforms PHS-CSMA in terms of power consumption because nodes in S-TDMA are fully time synchronized and the power wasted trying to communicate before packet transmission is much less than what it is in the other two protocols.

Figure 4.12 shows that packet delay in all protocols increases as T_{wi} increases. This trend is expected as longer T_{wi} makes a node waits more before it sends or forwards its packet to the next node. The same figure shows that P-CSMA still has the worst performance because of collisions, which a packet experiences the most in P-CSMA. Despite being a contention-free MAC protocol, S-TDMA does not outperform PHS-CSMA, which has the best performance, in terms of packet delay. This is due to the fact that S-TDMA needs a longer time to confirm a successful transmission. For example, in S-TDMA, if node A has sent a packet to node B, node A waits until the next time slot of node B and listens to confirm successful transmission. During that time, node A can not send the packet to any other node. On the other hand, in PHS-CSMA, node A can assume successful transmission immediately after sending the packet by an Acknowledgment (Ack) packet. Therefore, in case of unsuccessful transmission, a packet can be retransmited faster in PHS-CSMA than in S-TDMA.

In terms of reliability, Figure 4.13 shows that all the basic version of protocols have almost the same performance. Among the Enhanced versions, S-TDMA slightly excels the others. This is due to the fact that a node in the enhanced version of S-TDMA performs route maintenance regardless of the node's level. An interesting note here is that the reliability of the system is independent of T_{wi}. Another interesting note is that the power performance of the PHS-CSMA protocol did not change by implementing the route maintenance mechanism. This can be explained in that, for a node with a packet to send, every attempt it makes to send the packet, it wakes up for a short time to listen to its parent beacon. Therefore, the power consumed in this process is very low compared with

that of the other two protocols, where every attempt means sending RTS as in P-CSMA or even sending the complete packet as in S-TDMA.

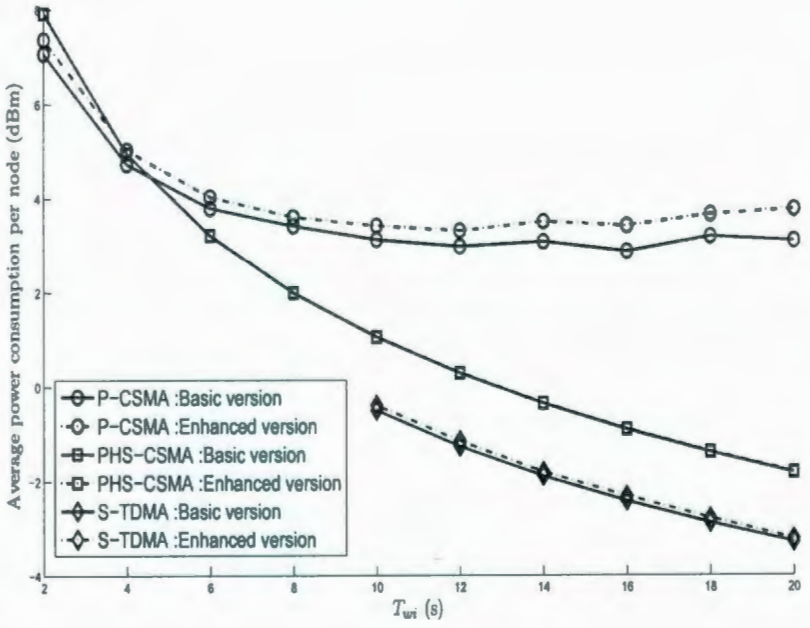


Figure 4.11: Power consumption against T_{wi}, at $N = 100$, $P_f = 0.2$.

Figures 4.14, 4.15 and 4.16 show the average power consumption per node, average packet delay, and reliability, respectively, at different values of node failure probability P_f. Simulation results are used to verify the analytical calculations of the reliability, and to evaluate the power consumption and packet delay. In the analytical model, the communication range R_c is considered to be 1 km as indicated in the system model. Figure. 4.16 shows good agreement between the analytical and simulation results, and shows that the main target of applying the presented route maintenance mechanisms is efficiently achieved. This claim is strongly supported by the significant enhancement on the reliability performance of the three protocols where a near unity reliability is achieved.

As shown in Figures. 4.14 and 4.15, the penalty of the reliability enhancement is the slight increase in power consumption and packet delay. Fortunately, the increase in

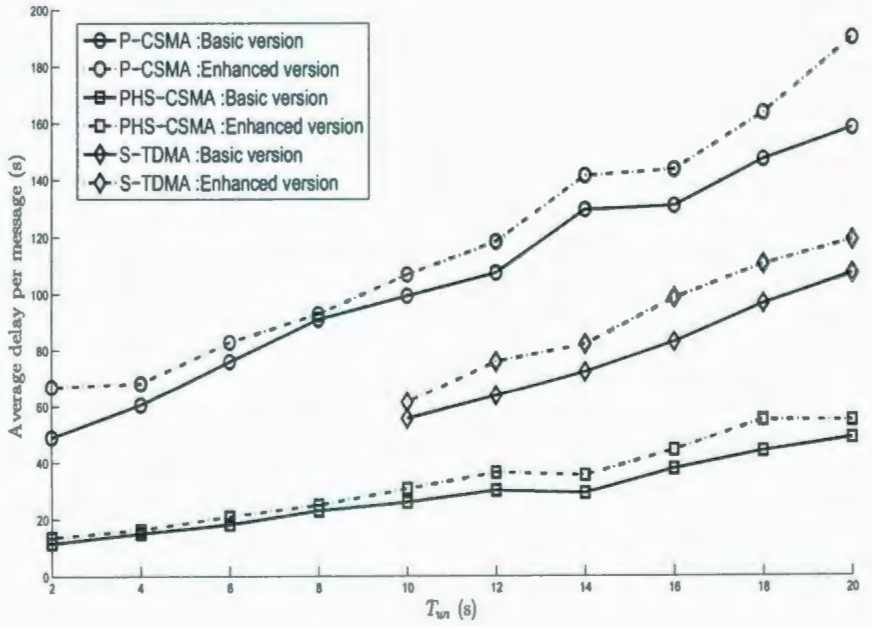


Figure 4.12: Delay against T_{wi}, at $N = 100$, $P_f = 0.2$.

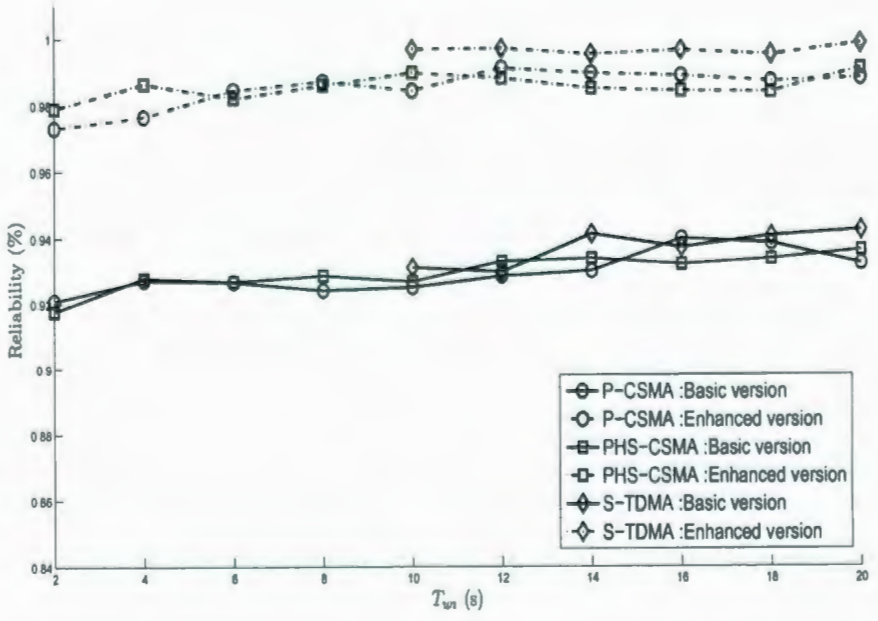


Figure 4.13: Reliability against T_{wi}, at $N = 100$, $P_f = 0.2$.

power consumption and packet delay is limited, where at 0.4 probability of node failure, power consumption is increased by 25%, 0% and 6%, and the delay is increased by 35%, 25% and 30% for P-CSMA, PHS-CSMA and S-TDMA, respectively. Also, the increase in the power consumption and packet delay is to be expected since the node makes more attempts to send its packet and could use a longer route.

Clearly, power consumption and delay in the basic versions are almost constant as P_f increases, except for P-CSMA where the delay decreases as P_f increases. This is due to the fact that a larger P_f means fewer working nodes in the system and consequently less channel contention.

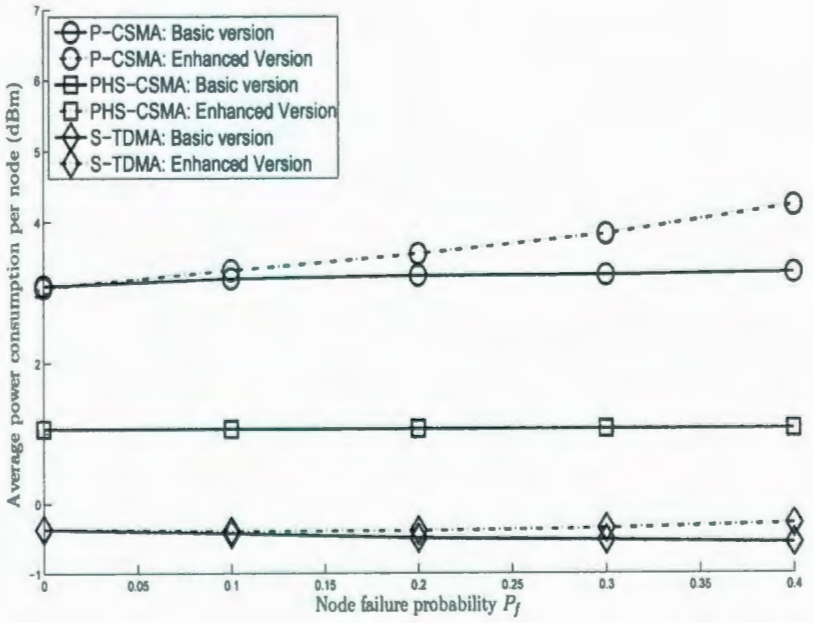


Figure 4.14: Power consumption at different values of node failure probability P_f, at $N = 100$, $T_{wi} = 10$.

The presented protocols support two types of transmissions or reporting. The first is called *regular checkup*, in which each node generates a packet every certain period of time. The second is called *emergency reporting*, in which emergency messages are just

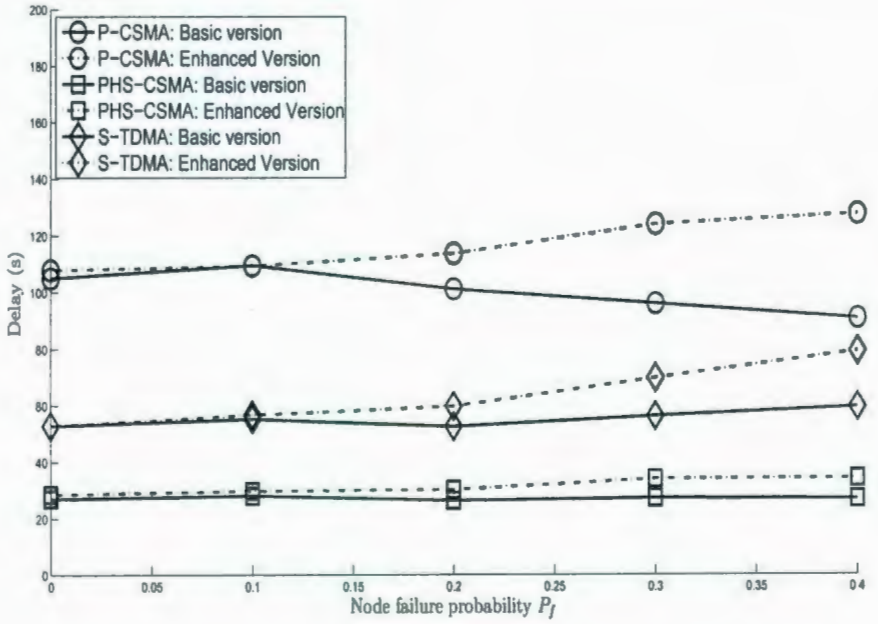


Figure 4.15: Delay at different values of node failure probability P_f, at $N = 100$, $T_{wi} = 10$.

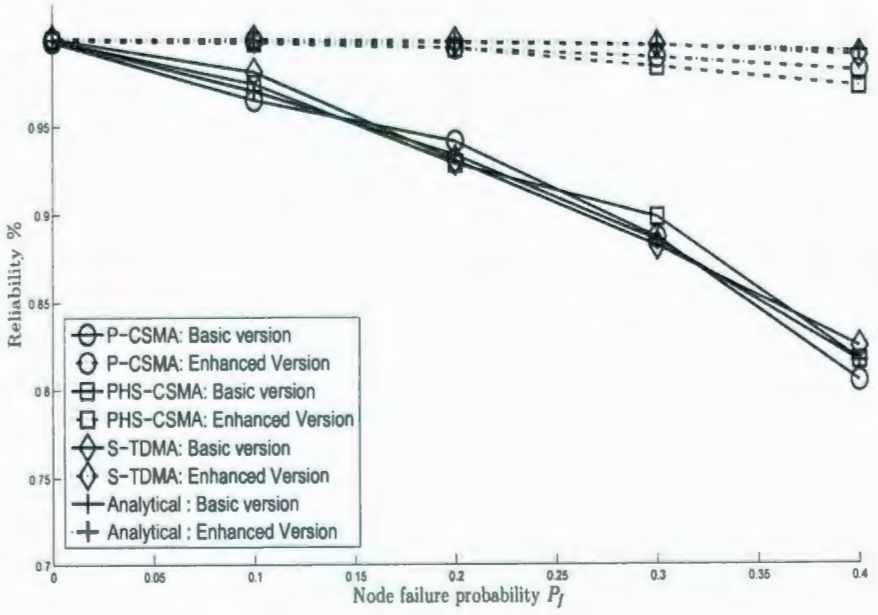


Figure 4.16: Reliability at different values of node failure probability P_f, at $N = 100$, $T_{wi} = 10$.

generated in case of fire detection. To this point, all the presented results are based on the *regular checkup*. For the sake of comparison between the proposed MAC protocols in the *emergency reporting* mode, the following scenario is considered.

- The system model is the same as for *regular checkup*, where 100 nodes are uniformly distributed in a circular cluster with 2.5 km^2 radius.
- No *regular checkup* is considered for these analysis (i.e., nodes do not generate a packet every half an hour.)
- A fire occurrence happens at a random time and a random location in the cluster.
- The fire is considered to expand with a speed of 0.5 km/h [11] in the all directions.
- A node is able to detect the fire if the fire is in the sensing range of that node. Sensing range is considered to be 100 m [32].
- The system stops working when the fire reaches *CH*. Before that, all nodes, for which the fire reaches their sensing range, generate an emergency message and send it to *CH*.
- Nodes burning is considered: when the fire reaches a node, then node burns (not working any more).

For *emergency reporting*, the performance measuring metrics are conceptually slightly different than what they are in the *regular checkup*. Power consumption is calculated by averaging the power consumed by the participating nodes only (not all nodes in the cluster).

Packet delay is calculated in the same way as in the *regular checkup* by averaging the time needed to transport a packet (emergency message in *emergency reporting* mode) from its source up to the *CH*. Along with the packet delay, we present a new metric to measure delay performance under the *emergency reporting* mode only. This term is called

event to CH delay, which represents the average time needed to report a fire from the time of its occurrence until the *CH* receives the first reporting message.

In terms of reliability, the concept here is different since we are talking about an event not a message, and this event can be detected by more than one node. Thus, an event can be reported by more than one message. Moreover, this event is going to be detected sooner or later. For example, if a fire occurred at a location where the closest node is not working, the fire expands to reach the sensing range of another node. Therefore, the metric of transport reliability is not meaningful in the *emergency reporting*.

Figure 4.17 shows the average power consumption per node. If we take the same number of participating nodes for the same period of time (e.g., 30 min), the number of generated messages will be less in the *emergency reporting* than in the *regular checkup* because all the participating nodes in the *regular checkup* generate packets, while in the *emergency reporting* mode, the participating nodes which detect the fire generate packets and the other nodes may participate in forwarding those packets. Therefore, it is expected for the power consumption to be less in the *emergency reporting* mode. S-TDMA still has the best performance in power consumption, but P-CSMA comes in the second place, and PHS-CSMA has the worst performance. This change in the trend is due to the very light traffic, under which a node in P-CSMA just wakes up for T_{ac} during T_f most of the time. On the other hand, under the same light traffic, a node in PHS-CSMA sends a beacon besides waking up for T_{ac} during T_f most of the time.

In terms of packet delay, Figure 4.18 shows that PHS-CSMA is the fastest to transport emergency packets to the *CH*. P-CSMA comes in the second place, while S-TDMA needs the longest time among the proposed protocols. It is interesting how the trend is changed from that in the *regular checkup*, where P-TDMA had the worst performance. This is again due to the low collision rate under light traffic in *emergency reporting* mode, which makes P-CSMA able to deliver the packet faster.

Clearly, from Figure 4.19 and Figure 4.18, Event-CH delay is much more longer than

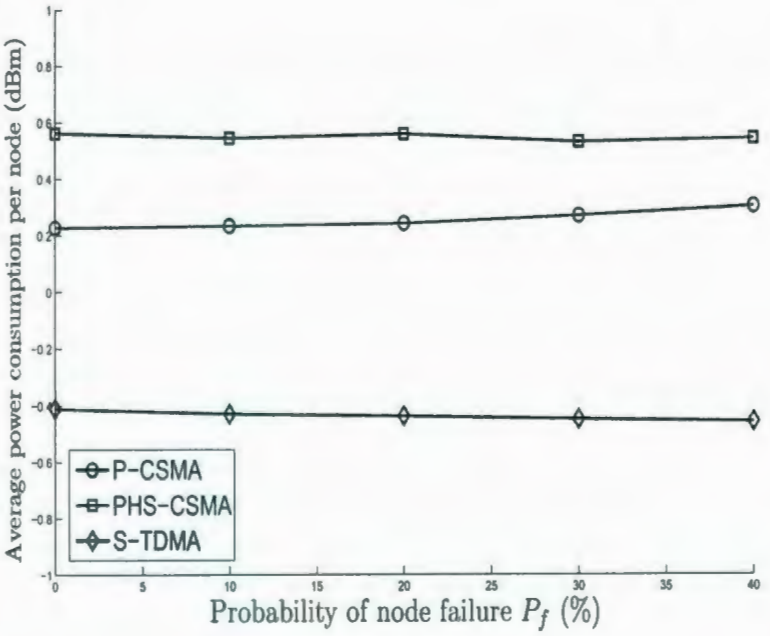


Figure 4.17: Power consumption in *emergency reporting*.

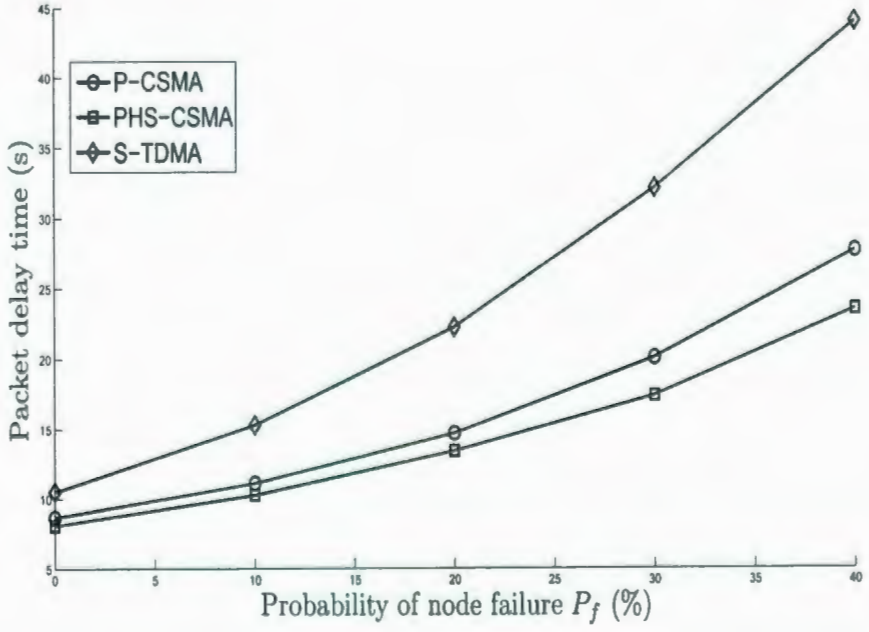


Figure 4.18: Packet delay in *emergency reporting*.

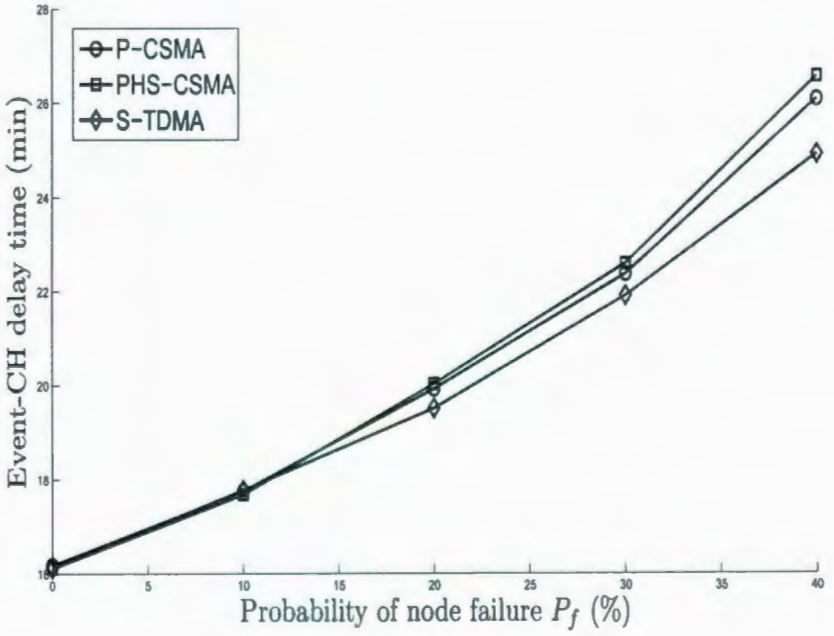


Figure 4.19: Event-CH delay in *emergency reporting*.

packet delay. For example, the three protocols with no failure probability need about 16 minutes to report a fire, while just 10 s at maximum of this time is needed to transport the packet. This supports our assumption that packet delay is not as critical as other aspects for this application, where the observed conditions are originally slow and packet delay could be negligible compared to Event-CH delay. Figure 4.19 shows that the proposed protocols have close Event-CH delays. For example, at the worst case with 0.4 node failure probability, S-TDMA needs 24.5 minutes to report a fire, while P-CSMA needs 25 minutes, and PHS-CSMA needs 25.2 minutes. This difference in Event-CH delays is due to the difference in the reliability of the protocols. Figure 4.16, shows that for the enhanced versions of the proposed protocols. S-TDMA has a slight better reliability than the other two protocols, and P-CSMA comes in the second place. Thus, for PHS-CSMA in case of fire, it's more probable in PHS-CSMA than the other protocols for the first emergency packet, which is generated by the nearest node, to be lost, and the fire is reported by other nodes. Therefore, more delay is expected in PHS-CSMA than the others. Similarly,

S-TDMA needs a shorter time to report a fire than the others. However, the probability of the first message to be lost is extremely small, which makes the difference in Event-CH delay between the proposed protocols very small.

4.4 Comparison with IEEE 802.15.4

To evaluate the efficiency of the proposed protocols, the IEEE 802.15.4 standard is simulated using Omnet++ considering the *regular checkup* under the system model presented earlier. Figures 4.20, 4.21, and 4.22 represent the average power consumption by node, average packet delay, and reliability performance of the IEEE 802.15.4 standard, respectively.

In terms of power consumption, comparing the power consumption of the proposed protocols in Figure 4.14 with the power consumption of the IEEE 802.15.4 in Figure 4.20, the proposed protocols show a significantly better performance, with an average gain in power saving of 14 dBm. However, this gain is at the price of high increase in the packet delay performance as shown in the Figures 4.15 and 4.21. For example, at probability of failure of 0.2, the IEEE 802.14.5 standard needs around 15 ms to deliver a packet, while, S-TDMA PHS-CSMA and P-CSMA need 25 s, 60 and 115 s, respectively. This gain in power saving is due to the fact that the IEEE 802.15.4 standard does not implement sleep mode. Therefore, the power consumption of the IEEE 802.15.4 standard matches with upper bound, which is calculated in the section 4.3, of power consumption for MAC protocols. This also explains the high difference in packet delay. In terms of reliability, as shown is the Figures 4.16 and 4.22, the proposed protocols show a better performance than the IEEE standard IEEE 802.15.4, which has a best reliability of 0.945, while the proposed protocols achieve a reliability of almost 1.

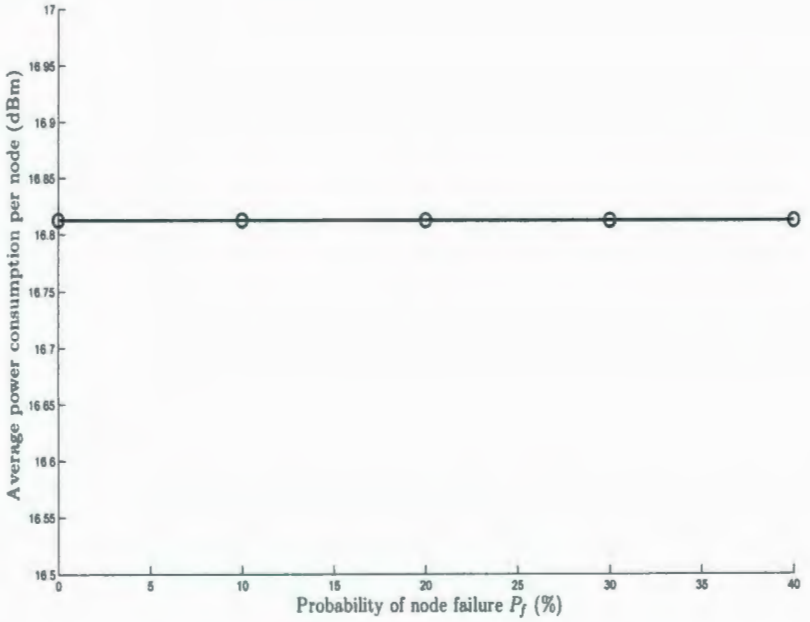


Figure 4.20: Power consumption in the IEEE 802.15.4.

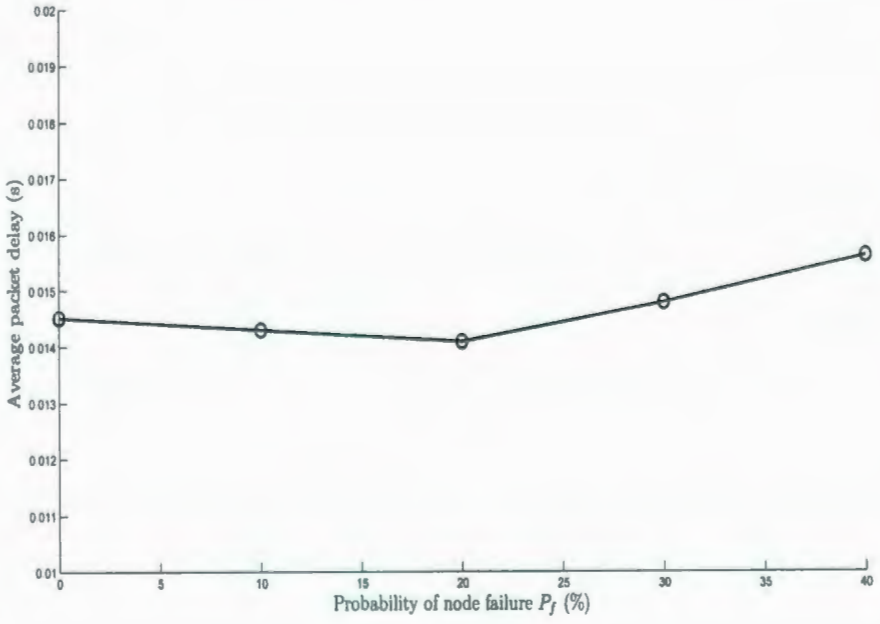


Figure 4.21: Packet delay in the IEEE 802.15.4.

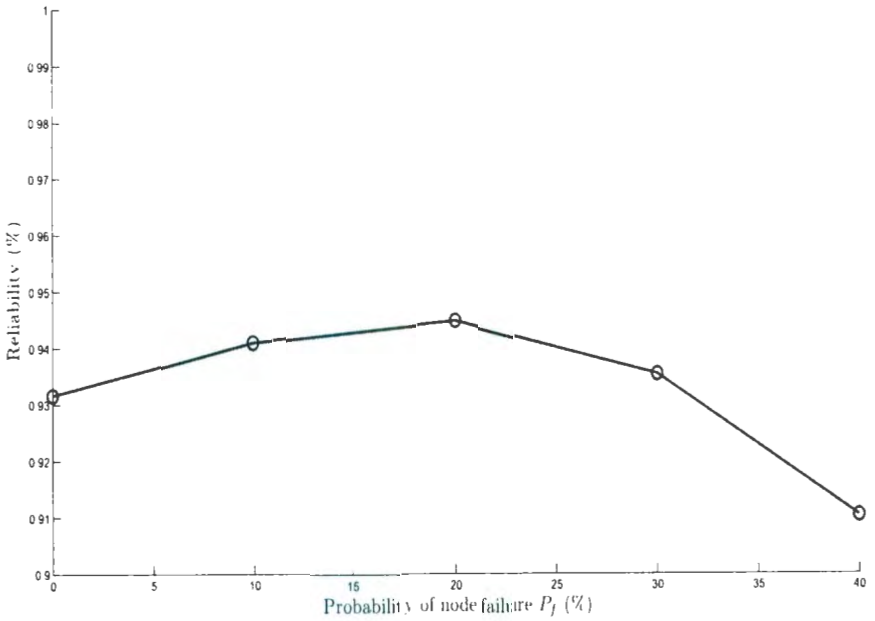


Figure 4.22: Reliability in the IEEE 802.15.4.

Chapter 5

Conclusions

Forest fires are serious natural disasters because of their threat to public safety and natural resources. Thus far, wild fire detection systems have relied on satellite imaging. The main drawback of these systems is the long delay in fire reporting. As a substitute to satellite imaging systems, WSN-based fire detection supports almost real time monitoring. Moreover, WSN-based systems can predict fire occurrences based on weather conditions, such as temperature and humidity.

For forest fire detection application, as with many other applications of WSNs, power saving takes the first priority. On the other hand, a small packet delay is not critical since the weather observation used to indicate a fire occurrence changes slowly. Data reliability is a very important factor to improve the accuracy of fire detection because any missed fire detection can cause a disaster. Also, false fire reporting can cause a huge waste in efforts.

In this thesis, forest fire detection application of WSNs is studied extensively, and the main priorities and restrictions are highlighted for this application. Node density is one of the important factors in the network design since the performance of the system is influenced by this factor. Many other factors, like connectively and coverage area, depend directly on node density as well. Therefore, node density is studied deeply to determine the optimum node density in the network. The optimum value is considered to

be the lowest number of nodes that guarantees connectivity for all nodes in the network. A simulation model is applied to find that optimum density. Results shows that 5 node/km^2 of node density is the optimum value.

Also, in this thesis, three MAC protocols are proposed to satisfy the priorities and demands of forest fire detection. Two of the proposed protocols which are Persistent-Carrier Sense Multiple Access (P-CSMA) and Per Hop Synchronization-Carrier Sense Multiple Access (PHS-CSMA) are contention-based MAC protocols. However, PHS-CSMA implements a temporary time synchronization between nodes just at the time of communication between them. P-CSMA does not implement any type of time synchronization between nodes. The third protocol is called Sensor-Time Division Multiple Access (S-TDMA). This one is a time division multiple access based protocol, in which all the nodes in the cluster are completely time synchronized. In terms of implementation complexity, S-TDMA is the most complex one since it needs a global synchronization among the cluster. PHS-CSMA comes in the second place, and P-CSMA is the simplest one since it does not require time synchronization.

The proposed MAC protocols are simulated for *regular checkup* and *emergency reporting* to compare their performances adaptively with the forest fire detection application. In the *regular checkup*, nodes generate messages periodically to report the current reading of the observed conditions, while in the *emergency reporting*, a message is generated just in the case of a fire being detected. Three basic metrics are used to compare the proposed protocols. These metrics are: Average power consumption per node, average packet delay and reliability. A metric is added to these metrics just for the *emergency reporting*. This metric is called Event-CH delay.

Results show lack of data transport reliability in these protocols, especially at high node failure probabilities. This is because of the absence of route maintenance mechanisms. Therefore, route maintenance mechanisms are proposed as well at the cost of small increase in the power consumption and packet delay. An analytical model is developed to

evaluate the reliability performance. Results show a good agreement between analytical results and simulation results, which both show that a near unity reliability is achieved using the proposed mechanisms with a slight increase in power consumption and delay.

In terms of power consumption, results show that S-TDMA outperforms other protocols in both *regular checkup* and *emergency reporting* mode. Also, S-TDMA is the fastest protocol to report a fire, and the second one in terms of packet delay under the *regular check up*. However, as mentioned before, S-TDMA is the most complex among the presented protocols in terms of implementation. On the other hand, P-CSMA is the simplest protocol but has the worst power consumption and packet delay performance. PHS-CSMA has the best packet delay performance and comes in the second place after S-TDMA in terms of power consumption. Even though PHS-CSMA has the worst power consumption performance in *emergency reporting*, it still can be considered better than P-CSMA in terms of power consumption. This is due to the fact that *regular checkup* is the normal condition which nodes operate all the time, while *emergency reporting* is an exceptional event. However, PHS-CSMA still needs more complex implementation than P-CSMA, and needs takes longer time to report a fire than P-CSMA. In terms of reliability, all the proposed protocols have almost the same performance.

Comparing the proposed protocols with the IEEE 802.15.4 standard. a significant enhancement is achieved in terms of the power consumption. This price paid for this gain is the longer data transport delay.

Therefore, we can conclude that there is no superior protocol which outperforms others in terms of power consumption, packet delay, Event-CH delay, reliability and complexity. However, a trade-off does exist.

Future work should be considered as follows:

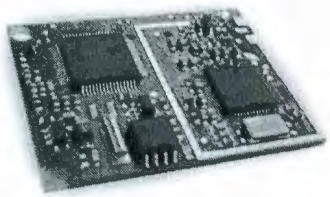
- Extend the proposed protocols to include communications between cluster heads up to the sink.

- Extend the analytical model to measure the power consumption and delay.
- Study the performance of the proposed protocols in terms of different aspects like power consumption fairness and scalability.
- Extend the reliability analysis to include the sensing data reliability along with the data transport reliability.
- Enhance the reliability performance by taking advantage of the essential localization feature. This can be done by confirming a fire occurrence by the readings of more than one node which are located close to the fire.
- Extend the analysis to include different types of node failures other than the random failures, such as a group of neighbor nodes fail at once.

Appendix

TinyNode 584 *Embedded Wireless Network Node*

The TinyNode 584 is an ultra-low power OEM module that provides a simple and reliable way to add wireless communication to sensors, actuators, and controllers. TinyNode 584 is optimized to run TinyOS and packaged as a complete wireless subsystem with 19 configurable I/O pins offering up to 6 analog inputs, up to 2 analog outputs as well as serial interface.



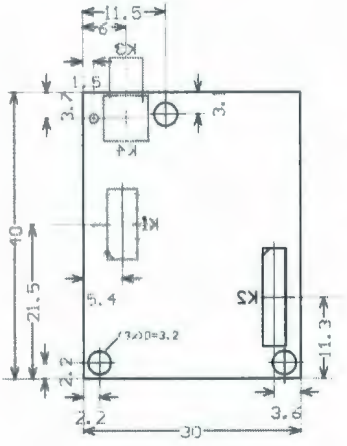
KEY FEATURES

- Ultra Low Power 3 V design: > 5 years battery life on 2/3AA Lithium batteries (using sleep modes)
- Easy to integrate with a wide variety of sensors and actuators
- 8MHz Texas Instruments MSP430 microcontroller
- 868 MHz Xemics XE1205 ultra-low power multi channel wireless transceiver
- On-board temperature sensor
- Small: 30x40 mm

Software adjustable for long range low bandwidth or short range high bandwidth connections

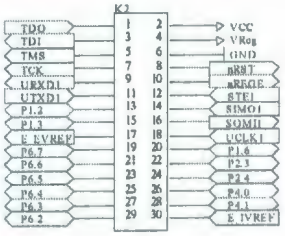
- High sensitivity (down to -121 dBm) RF receiver
- Transmitter output power up to +12 dBm
- 30 pin Molex 52465-3071 board-to-board connector
- On-board 1/4 wave wire antenna, footprint for SMA and MMCX connector, footprint for MMBX board-to-board connector (MMBX-S50-0-1)
- Analog, digital and serial interfaces
- Fast wakeup from sleep (<6µs)
- Out-of-the-box TinyOS support: mesh networking and communication implementation

DRAWINGS



Dimensions and connector position, view from above (the connector is behind the printed circuit board.

Pinout K2
Molex 52465-3071



TINYOS

A small, open source, energy efficient, software operating system developed by UC Berkeley, TinyOS supports large scale, self-configuring sensor networks. The source code and software development tools are publicly available at:

http://webs.cs.berkeley.edu/tos

shockfish /

TinyNode 584 Embedded Wireless Network Node

INTERFACE SPECIFICATIONS

Analog inputs/outputs	Six 0 to 2.5 V 12-bit analog inputs Two 0 to 2.5 V 12-bit analog outputs
Digital inputs/outputs	max. 19 digital I/O, LVTTL (3 V)
Serial port	[z] bps UART, LVTTL (3V) signaling SPI interface
Hardware Interface	Molex 52465-3071

CPU PERFORMANCE

Bus speed	
Maximum	8 MHz
Standard	4 MHz
RAM	10K bytes
Program Space	48K bytes
External Flash	512K bytes
Configuration Flash	256 bytes
Flash reprogramming cycles	>100'000

WIRELESS SPECIFICATIONS

Operating frequency	868-870 MHz 2-10 channels
RF output power	0 to +12 dBm
Data rate	1.2-152.3 kbps
Receiver sensitivity	
@ 1.2 kbps	-121 dBm
@ 76.8 kbps	-104 dBm
@ 152.3 kbps	-101 dBm
Range @ 76.8 kbps	
Outdoor (1m elevation)	200 m (+5 dBm)
Indoor	40 m (+5 dBm)
Current consumption	
Transmit @ +5 dBm	33 mA
Receive	14 mA
Sleep	< 1µA

RECOMMENDED OPERATING CONDITIONS

Parameter	Maximum
Digital input/output current	
Output source	6 mA
Output sink	6 mA
Total output	6 mA
Analog input	
Input MUX ON resistance	2 kΩ
Input capacitance	40 pF
Load current on Vref	1 mA
Analog output	
Max load capacitance	100 pF
Max load current	1 mA

TOTAL CURRENT CONSUMPTION

State	mA
Sleep, Timer off	0.004
Sleep, Timer on	0.007
µC only (@ 4 MHz)	2
Receive (inc. µC)	16
Transmit (inc. µC)	
0dBm/1mW	25
10dBm/10mW	46
12dBm/16mW	62

ABSOLUTE MAXIMUM RATINGS

Parameter	Min	Max	Units
Supply voltage			
VBAT (VCC = VBAT)	2.4	3.6	V
during flash memory programming	2.7	3.6	V
VSUP (VCC = 2.8 V)	2.9	5.5	V
Voltage on any pin	-0.3	VCC+0.3	V
Storage temperature	-40	+85	°C
Operating ambient temperature	-40	+85	°C

shockfish /

Tel. +41 21 693 85 15
shockfish@shockfish.com
www.shockfish.com

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] Crossbow. Mica2 datasheet. `http://www.xbow.com`.

[3] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *IEEE Communications Magazine*, 43(5):51–58, 2002.

[4] K. Sohrabi, B. Manriquez, and G. J. Pottie. Near ground wideband channel measurement in 800-1000 mhz. In *Proceedings of the IEEE Vehicular Technology Conference*, pages 571–574, Jul 1999.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, volume 2, page 10, Jan 2000.

[6] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In *Proceedings of the IEEE INFOCOM*, volume 1, March 2004.

[7] O. Younis, M. Krunz, and S. Ramasubramanian. Node clustering in wireless sensor networks: recent developments and deployment challenges. *IEEE Network*, 20(3):20–25, 2006.

[8] Y. Chen and C. Wen. Adaptive cluster-based scheduling management for wireless ad-hoc sensor networks. In *Proceedings of the International Conference on Sensor Technologies and Applications*, volume 1, pages 256–263, June 2009.

[9] The British Columbia Forest Service. Protection branch-ministry of forests and range- province of british columbia. `http://bcwildfire.ca/`.

[10] NASA. Modis website. `http://modis.gsfc.nasa.gov`.

[11] Natural Resources Canada. Canadian wildland fire information system - canadian forest fire weather index (FWI) system. `http://cwfis.cfs.nrcan.gc.ca/en_CA/background`.

[12] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th Annual International Conference on Mobile computing and networking*, pages 148 – 159, Sep 2002.

[13] J. Albowicz, A. Chen, and L. Zhang. Recursive position estimation in sensor networks. In *Proceedings of the 9th International Conference on Network Protocols*, pages 35 – 41, Nov 2001.

[14] M. Hefeeda and M. Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[15] David M. Doolin and Nicholas SITAR. Wireless sensors for wildfire monitoring. In *Proceedings of the Conference on Sensors and smart structures technologies for civil, mechanical, and aerospace systems*, volume 5765 (2), July 2005.

[16] Junguo Zhang, Wenbin Li, Ning Han, and Jiangming Kan. Forest fire detection system based on a zigbee wireless sensor network. *Frontiers of Forestry in China*, 3(3):369–374, Sep. 2008.

[17] Wei Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE INFOCOM*, volume 3, pages 1567–1576, 2002.

[18] IEEE Standard 802.11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999).

[19] IEEE Standard 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) (2003).

[20] J. Lee. Performance evaluation of IEEE 802.15.4 for low-rate wireless personal area networks. *IEEE Transactions on Consumer Electronics*, 52(3):742–749, Aug 2006.

[21] Zhijia Chen, Chuang Lin, Hao Wen, and Hao Yin. An analytical model for evaluating IEEE 802.15.4 CSMA/CA protocol in low-rate wireless application. In *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops*, volume 2, pages 899–904, May 2007.

[22] S. Eisenman and A. Campbell. E-CSMA: Supporting enhanced CSMA performance in experimental sensor networks using per-neighbor tran. In *Proceedings of the IEEE INFOCOM*, pages 1208 – 1216, May 2007.

[23] S. Liu, K-W Fan, and P. Sinha. CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor. In *Proceedings of the IEEE SECON*, pages 11 – 20, June 2007.

[24] G. Pei and C. Chien. Low power TDMA in large wireless sensor networks. In *Proceedings of the Military Communications Conference (MILCOM'01)*, volume 3, pages 347 – 351, Oct 2001.

[25] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (Sensys'03)*, volume 3, pages 181 – 192, Nov 2003.

[26] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138 – 149, 2003.

[27] H. AboElFotoh, S. Iyengar, and K. Chakrabarty. Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failure. *IEEE Transactions on Reliability*, 54:145 – 155, March 2005.

[28] O. Akan and I. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 13:1003–1016, Oct. 2005.

[29] TinyNode. Tinynode 584 fact sheet. `http://www.tinynode.com`.

[30] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking*, pages 80 – 91, 2002.

[31] Benyuan Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 475 – 483, Oct. 2004.

[32] M Hefeeda and M Bagheri. Wireless sensor networks for early detection of forest fires. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 1–6, Oct. 2007.