# ASHESI UNIVERSITY COLLEGE

## THE PERFORMANCE OF GENETIC ALGORITHMS IN MEDIUM ACCESS CONTROL PROTOCOLS FOR WIRELESS SENSOR NETWORKS

**UNDERGRADUATE APPLIED PROJECT**

B.Sc. Computer Science

**Obed Kobina Nsiah**

**2016**

**ASHESI UNIVERSITY COLLEGE**

# The Performance of Genetic Algorithms in Medium Access Control Protocols for Wireless Sensor Networks

**UNDERGRADUATE APPLIED PROJECT**

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science

**Obed Kobina Nsiah**

**April 2016**

# DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

………………………………………………………………………………………

Candidate's Name:

………………………………………………………………………………………

Date:

………………………………………………………………………………………

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University College.

Supervisor's Signature:

………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………

Date:

………………………………………………………………………………………

# Acknowledgements

I would like to express my sincere gratitude to my advisor, Ms. Grace Sallar, for the support she provided throughout the execution of this project.

I would also like to express special thanks to Dr. Nathan Amanquah for giving me this project idea and guiding me alongside my supervisor.

My final gratitude goes to my family and friends who helped me throughout this project.

# Abstract

This paper evaluates the performance of the Particle Swarm Optimization (PSO) scheme, a genetic algorithm, as used in Medium Access Control (MAC) protocols for Wireless Sensor Networks (WSNs). In doing so, the protocol is implemented and compared to round-robin, a contention-free protocol that makes sensor nodes take turns in sending their data packet to the sink, and a p-persistent CSMA protocol with a p-value of 0.01. The concept of p-persistence as used in the CSMA protocol is employed to tweak the PSO scheme to form a fourth protocol that is used in the evaluation process. All four protocols are simulated in OMNET++, a network simulator, and performance analysis is done based on the throughput and transmission delays of the various protocols. It was discovered that the PSO scheme performed poorly in terms of throughput in highly populated networks with high traffic. The modified PSO scheme with p-persistence performed better than the original PSO scheme in terms of throughput for densely populated networks with high traffic. On low populated networks with low traffic, the original PSO scheme showed the best throughput performance. Round-robin displayed higher throughput performance than the other three protocols for both low and high traffic networks with a high node population.

# Table of Contents

# Chapter 1: Introduction

This chapter briefly explains Wireless Sensor Networks (WSNs) and their importance. It also explains concepts like the Medium Access Control (MAC) protocol and its relevance and briefly explores key terminologies such as a genetic algorithm and others used in this paper.

## 1.1 Introduction & Background to the Project

### 1.1.1 Wireless Sensor Networks and Their Relevance

As technology advances, and with scientists increasingly taking interest in the goal to achieve a smart world, WSNs have become one of the technologies pioneering this smart world. A Wireless Sensor Network (WSN) is a deployment of a collection of autonomous devices with sensors that work collaboratively to achieve a specified goal (Fan & Parish, 2007). There are three main parts of a WSN. The first part consists of the set of sensor nodes with computing capabilities that monitor and measure some physical parameters in the environment (Xu, Shen, & Wang, 2014). The second part of the WSN is the wireless communication channel via which the data gathered from the environment is conveyed. The MAC protocol is used to manage this transmission (Xu, Shen, & Wang, 2014). The final part of the WSN is the central location that receives all the data transmitted from the sensor nodes over the channel to make meaning of the data received (Xu, Shen, & Wang, 2014).

The emergence of WSNs is accompanied by a wide range of endless applications since any kind of sensor can be used on the sensor nodes in the network. Some of the known and implemented applications of WSNs include monitoring of marine environments by measuring the water temperature, pressure, wind direction or wind speed which can be used to detect incoming disasters or threats from the ocean (Krishnamachari, 2005). WSNs with thermal sensing capabilities are also used to detect forest fires for rapid emergency response and also used to detect people in restricted or non-authorized areas to increase security

(Libelium Comunicaciones Distribuidas S.L., 2015). Also, for rapid emergency response, some WSNs monitor seismic activities to detect foreboding earthquakes (Krishnamachari, 2005). There exist some Wireless Sensor Networks that are used in buildings to automatically control a building's ventilation, lighting, air conditioning or heating, based on the climate or environment conditions (Libelium Comunicaciones Distribuidas S.L., 2015). Other possible applications of WSNs are to monitor industrial processes such as wine quality enhancement or measure air pollution and radiation levels to make habitats and ecosystems safer (Libelium Comunicaciones Distribuidas S.L., 2015). The applications of WSNs are endless and result in the enhancement of human lives. To add up to their benefits, wireless sensor networks are easy to deploy and also offer the advantages of real-time monitoring of uncontrollable environments with topological constraints (Krishnamachari, 2005). Sensor nodes are also low cost, low-power and easily manufactured (Fan & Parish, 2007).

### 1.1.2 Medium Access Control (MAC) Protocol

This paper takes a keen interest in the Medium Access Control (MAC) protocols used in transmitting the data to the central location. All the sensor nodes in the WSN transmit their data over a shared broadcast channel. Considering this, it is necessary to manage which node gets to use the channel when there is competition for it. Tanenbaum and Wetherall (2011) likens this situation to a conference call where the callers are on different telephones and all the telephones are connected such that one can hear and also talk to everyone else on the call. There will be chaos when two or more people start to talk at the same time. In order to deliver meaningful information and have the others on the telephone line clearly hear one another, only one person has to talk at any particular time. Similarly, some system needs to be instituted in a WSN to manage how the broadcast channel is accessed by the nodes on the network. In networks, the MAC protocol is used to solve this problem (Tanenbaum & Wetherall, 2011). The MAC layer of a network typically uses protocols, which are usually software running on the physical nodes, to ensure that signals sent from different sensor nodes across the same channel do not collide. When a poor MAC protocol is used in the management of the broadcast channel, it defeats the benefits of a WSN that were illustrated above since the data collected is garbled by collisions and is not effectively communicated to the central system to be processed. An effective and ideal MAC protocol avoids collisions while increasing throughput of the network (Stankovic, 2006). A good MAC protocol also requires small memory for execution and is implemented with a small size code (Stankovic, 2006). The different MAC protocols that have been implemented so far have employed several different mechanisms to satisfactorily meet the metrics of becoming an effective MAC protocol. The most commonly used MAC protocols are contention-based (Stankovic, 2006). Contention-based MAC protocols follow a simple algorithm. Each node on the network tries to send the data it has. When two or more nodes send their data at the same time, collision of their data occurs at the central station. The algorithm tries to resolve this collision by making the colliding nodes back off for some amount of time. When the back-off time for a node expires, the node sends its data again. The process of backing off for some amount of time is repeated any time there is a collision. The unique behaviour of the different types of contention-based MAC protocols that exist, their pros and cons, and their performances when the density of the network is high and when the density of the network is low are further discussed in section 2.1. A new wave of attempts to optimize MAC protocols especially for wireless networks have introduced mechanisms where genetic algorithms are being integrated in MAC protocols to improve the standards.

### 1.1.3 Genetic Algorithms

Mitchell (2008) explains a genetic algorithm as a technique for solving optimization problems by a process of natural selection that is akin to genetic evolution. Problems solved with genetic algorithms often have a number of entities trying to find a set of parameters that optimize the solution to the problem. Each entity tries to solve the problem individually and a fitness score is given to each entity based on how well it solves the problem. The algorithm then mutates the set of parameters of all entities with the values of the one with the best fitness score. All entities try to optimize the new set of parameters given and this process is repeated until the problem has been fully optimized. An example of a genetic algorithm is the Swarming Intelligence (SI) motivated by the social behaviour of ants, schools of fishes, flocks of birds, bees and termite colonies and how they interrelate and cooperate with one another (Bonabeau, Dorigo, & Theraulaz, 1999). Bonabeau, Dorigo, & Theraulaz (1999) provide exapmles of how the SI algorithm was implemented to make a swarm of tiny robots successfully transport a load from one location to the other. The SI algorithm has been versioned a number of times to create other algorithms like the Particle Swarm Optimization (PSO) (Kennedy, 2010), the bees algorithm (Pham, et al., 2006) and the Ant Colony Optimization (Dorigo & Birattari, 2010). Mickus, Clarke & Mitchell (2015) developed a new MAC protocol using the Particle Swarm Optimization which is discussed into detail in section 2.1 of this paper alongside some other related works done with genetic algorithms and MAC protocols.

## 1.2 Objectives

The aim of this project is to evaluate the performance of some common genetic algorithms when implemented as MAC protocols for wireless sensor networks. The performance of these genetic algorithms will be compared to other existing MAC protocols to find out their strengths and weaknesses. Three MAC protocols will be compared to one another to find out their strengths and weaknesses. The goal is to find out how genetic MAC protocols behave at low load and high load on the network. The evaluation of the implemented MAC protocols will be done based on two measures outlined below:

- The efficiency of the algorithm in terms of throughput (the number of packets successfully delivered to the base station/sink node in a given interval)
- The delay of the protocol (the amount of time each node waits to send a data packet to the sink node)

### 1.3 Outline of Dissertation

The organization of this paper is as follows: Chapter 2 presents the research done on the topic at hand with section 2.1 outlining existing studies and investigations done on the subject matter and problem. Chapter 3 discusses the methodology regarding the research. The implementation procedure followed for each protocol is also given in this section. Chapter 4 discusses and analyses the results generated from the study and chapter 5 concludes the paper with recommendations for other interesting fields of research in relation to the subject matter of this paper and based on the results of the study.

# Chapter 2: Research

This chapter explores the current literature and studies in MAC protocol and wireless sensor networks research. It provides information on some genetic algorithms used as Medium Access Control protocols including essential discoveries, theoretical and methodological contributions made to the field and knowledge base of both MAC and WSNs research.

## 2.1 Literature Review

Many different MAC protocols have been implemented and analysed to coordinate how a number of users competing for a single shared channel may use the channel. All of these protocols each have their unique strengths and weaknesses depending on the conditions of the network. In the 1970's, the first contention-based protocol called Pure ALOHA was developed by Norman Abramson (Tanenbaum & Wetherall, 2011). In this protocol, any node on the network that is ready to send a data packet is allowed to do so at any time. Collisions may occur during transmission. When this happens, the colliding nodes are to back-off and wait for a random amount of time, and try again when their specific waiting times are over. This process of backing-off and waiting for a random amount of time is repeated for any number of nodes whose packets collide until all the nodes are successfully able to deliver their data to the base station (Qiao, Yang, Ma, Zhang, & Dong, 2013). Pure ALOHA was simple to implement but its efficiency was low in that in a high density network where nodes had a lot of traffic to send around, there were a lot of collisions (Tanenbaum & Wetherall, 2011). Throughput (number of successfully transmitted packets over an interval) was low despite the delay (how long a node waits before transmitting) being low. With this protocol, a node that was already sending data to the base station could have its data jumbled when a new node immediately started to send out its data. To combat the issue, a variation of ALOHA called the Slotted ALOHA was introduced where the nodes on a network were only allowed to transmit their data at discrete time periods when it was guaranteed that any already transmitting node would have completely delivered its packet. The only problem with this protocol was that at any particular allowable time slot to transmit, a number of nodes could attempt to send data at the same time. However, collisions were less with this protocol compared to Pure ALOHA (Tanenbaum & Wetherall, 2011).

It was also realized that some of the collisions could be avoided if a ready node listened to the network before transmitting. This concept of listening to the network before transmitting introduced a protocol called Carrier Sense Multiple Access (CSMA) (Tanenbaum & Wetherall, 2011). In this protocol, a node that has data to transmit tests the channel to see if the channel is busy. The message is then transmitted immediately if the channel is not busy. When two or more nodes transmitting data at the same time collide, each node waits a random amount of time and tries again later to avoid re-colliding. The process of waiting for a random amount of time to avoid collision is repeated for any nodes whose data collide with one another (Stankovic, 2006). It was found that this protocol performed better than ALOHA in terms of throughput since nodes refrain from sending immediately when they have data to transmit but only do so when no node is transmitting. Several alterations of the Carrier Sense Multiple Access (CSMA) protocol have been introduced by different researchers. In one of the alterations of the CSMA, all nodes in the network transmit their data with some amount of probability after realising that the channel is idle (Egea-López & Vales-Alonso, 2007). CSMA protocols that implement this mechanism are called p-persistent CSMA where p represents the probability of a node transmitting (Egea-López & Vales-Alonso, 2007). Therefore, a p-persistent CSMA protocol with a p of 0.5 will allow the nodes on the network who are ready to transmit data transmit with a probability of 0.5 when the network is idle. This implies that even less number of nodes will transmit when the network is idle to reduce the number of collisions compared to the original CSMA implementation which in theory has a p value of 1 since all nodes will attempt to transmit on an idle channel. Experimental results have shown that CSMA generally performs better in terms of throughput at high loads and traffic on the network than ALOHA (Tanenbaum & Wetherall, 2011). P-persistent CSMA with lower p values also offer high throughput compared to p-persistent CSMA with high p values. Delay is high with the CSMA protocol and so on low loads where slotted ALOHA show seemingly similar behaviour to CSMA in terms of throughput, it is often preferred owing to its low delay (Tanenbaum & Wetherall, 2011). Figure 2.1 is graph from Tanenbaum & Wetherall (2011) showing the performance of the various ALOHA and CSMA protocols in terms of the throughput across an increasing load on the network as discussed above.

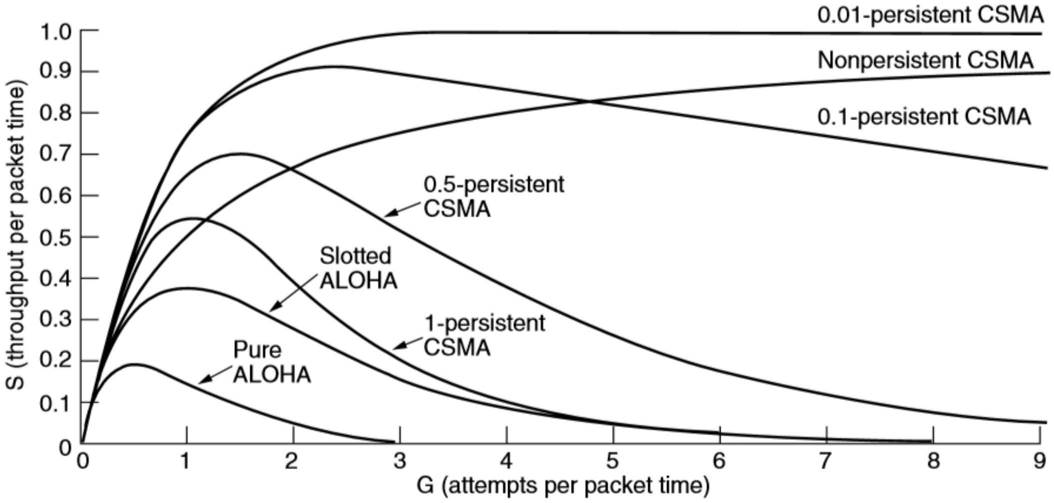


Figure 2.1: A graph of throughput per packet time against attempts per packet time

Other variations of CSMA employ an exponential back-off algorithm instead of the usual random wait times (Yassein, Manaseer, & Al-Turani, 2009). There also exists a Fibonacci increment back-off algorithm that Yassein, Manaseer, & Al-Turani (2009) reported to offer higher throughputs in mobile ad hoc networks than the exponential back-off. In addition to exponential back-off times, some CSMA schemes try to avoid collisions entirely before they happen (Mahalik, 2007). This is called the CSMA/CA where the CA stands for collision avoidance. In this protocol, a node that needs to transmit data to another node first listens to the channel to ensure that the channel is idle. It then sends a “request to send” (RTS) packet to the destination node. If the destination node is ready to receive any data, it sends a “clear to send” (CTS) packet back to the initial node as an indicator to send the packet (Mahalik, 2007). Any node that does not receive a CTS packet by the destination node when that node has sent an RTS packet to the sink never sends its packet until it receives the CTS packet. RTS packets may still collide if two or more nodes send to a particular node but this is not as disadvantageous as having the main payloads colliding. There is a field of protocols where no collision occurs. These kinds of protocols are called contention free protocols. Round-robin is an example. In the round-robin scheme, dedicated time slots are given to each node in the network to send a data packet. At any particular time slot, only one node is allowed to send data. Whether or not a node has data to send in its allocated time, the time slot is specifically reserved for that node and no other node in the network will attempt to send any data. The next set of MAC protocols to be discussed are ones integrated with genetic algorithms.

Mickus, Clarke, & Mitchell (2015) introduced a new MAC scheme grounded in the concept of swarm intelligence for Wireless Sensor Networks. By combining control engineering theory and Particle Swarm Optimization (PSO), they presented a new MAC protocol that enables a swarm of sensor nodes to independently find the maximum conceivable rate to generate and send packets. This PSO scheme is built on top of the CSMA protocol. The PSO MAC protocol is therefore a contention-based protocol. For their studies, they compared the new protocol to the CSMA protocol and found that the PSO scheme performed better for larger networks.

The control engineering theory aspect of their scheme uses feedback from earlier failed or successful transmissions of packets to delay the next permitted transmission for a node in order to regulate traffic from the sensor nodes in the network. Although this control technique has already been integrated in existing MAC protocols (in the form of back-off times that are either increased or decreased systematically for nodes in the network to avoid recurring collisions), their proposed control based approach is different. Mickus, Clarke, & Mitchell's (2015) control based approach uses previous information to adjust the back-off time while the regular back-off algorithm resets the time to a different value after successful transmission without necessarily considering old values.

With respect to the PSO scheme proposed by Mickus, Clarke, & Mitchell (2015), each node in the network is considered as a particle searching for the best transmission delay to increase the "fitness" of the network (which was taken to be "the average probability of success taken over a set of packet transmission outcomes"). With every packet and acknowledgement sent to neighbouring nodes by a node, each node shares the information it has about its transmission delay or back off time and the nodes in the network try to adjust their transmission delay to achieve the best probability of a successful transmission while considering the information it has from the other neighbours. The update formula used in computing by how much a particular back-off time should be adjusted is given below:

$$v = w \cdot v_{t-1} + \varphi 1 \cdot (pbest - x_{t-1}) + \varphi 2 \cdot (lbest - x_{t-1}) + taxis \qquad \text{(Equation 1)}$$

The actual back-off time is calculated as:

$$x = x_{t-1} + v, \text{ where,} \qquad \text{(Equation 2)}$$

- $v$ - is a variable determining by how much the back-off time of a node should be adjusted
- $x$ - is the back-off time
- $x_{t-1}$ - is the previous back-off time
- *pbest* - is the back-off time associated with best fitness from a node's personal experience
- *lbest* - is the back-off time associated with the best performing neighbour
- $\varphi 1$ and $\varphi 2$ - random weights in range [0,1]

- $w$, was defined as the inertia weight, a constant to limit the maximum possible value a back-off time can result in.

The variable *taxis* in equation 1 above is defined as:

$$taxis = \begin{cases} +v_c \,.\, (1 - p_s), & \text{if acknowledgement was a success} \\ -v_c \,.\, (1 - p_s).r_c, & \text{if acknowledgement was a failure} \end{cases} \qquad \text{(Equation 3)}$$

- $v_c$ – is a constant that defines by how much the nodes need to react to the feedback
- $p_s$ – is a probability of successful transmission based on previous values
- $r_c$ – is a constant called the control ratio to give the algorithm more control over the value the back-off time can assume.

Mickus, Clarke, & Mitchell (2015) also implemented a pure ALOHA protocol with a binary exponential back-off time to compare to the new protocol. The results from their study showed that throughput for the PSO scheme was better when compared to a CSMA protocol with a limited sensing range but was outperformed by a CSMA protocol with full sensing capability. With respect to delay performance, the PSO scheme performs better than both the CSMA and pure ALOHA protocols. The fairness level for all protocols analysed were similar and there were about 10 trial transmissions per packet or less for the PSO scheme which was better than the other protocols analysed with between 12 to 16 transmissions per packet.

The gap identified in this research is that the new proposed PSO protocol was compared to only contention-based protocols without necessarily evaluating how the genetic protocol performs against contention free or limited contention based protocols. Also, no analyses were made for the protocol for a low load network. It is with this motivation that the PSO scheme by Mickus, Clarke, & Mitchell is implemented in this paper. The PSO algorithm is compared to the round-robin, a contention free MAC protocol, and 0.01 persistent CSMA, a contention-based protocol, to evaluate their performances. No justifications were given for the value of the inertia weight (the constant value, w) used to limit the maximum possible back-off time Mickus, Clarke, & Mitchell (2015) used in the update formula in the PSO scheme. This paper experiments with differing inertia values to find out the effect it has on the performance of the PSO scheme. These MAC protocols will be evaluated using the metrics mentioned in section 1.2 of this paper. A p-persistent PSO scheme which merges the concept of probabilistic hold-off from p-persistent CSMA and the use of historical back-off times from PSO is also implemented and evaluated to analyse the effect the base CSMA protocol used in the PSO scheme has on the scheme.

# Chapter 3: Methodology and Approach

This section describes the implementation process of the round-robin protocol, the 0.01 persistent CSMA protocol and both versions of the PSO scheme used in this paper. It also introduces all software and tools used in the analyses and evaluation of the performance of the protocols and justifies the methodological approach used in gathering the data for the study.

Considering that the aim of this project was to evaluate the performance of genetic algorithms in MAC protocols for wireless sensor networks, the approach used in gathering data for the study was through simulation. This approach involved implementing the protocols and running them in a software that can simulate network environments. The simulation tool used in this study was OMNET++.

## 3.1 OMNET++

OMNET++, sometimes referred to as OMNETpp by other scholars, is a discrete event simulator that can simulate a wireless network interface. It is open-source and free to use for the purposes of academic research. It provides the functionality for implementing and testing code for MAC protocols that represent firmware running on sensor node devices. Networks in OMNET++ are built following a bottom-up architecture. This means that components or modules that make up the network are built as individual units which are then assembled into larger components to eventually constitute the network. Components making up a network simulation are programmed in C++. Each component has a Network Description (NED) file that describes its structure. Parameters associated with the component are defined in the NED file. If the component in question is a simple one that will eventually be used as a unit component in a network, it is accompanied by C++ header and source files that define its behaviour in the network.

### 3.2 Implementation of Round-Robin Protocol

The round robin MAC protocol implemented for this study is simple. According to the algorithm, nodes in the network take turns in sending the data they have to the base. There is a timer variable in the protocol that determines at which time slot each node in the network should send the packet. When the time-slot reserved for any node to send data to the sink is up, an event is sent to that node. A high-level implementation of the round-robin algorithm specific to this study is outlined below:

```
for each node in the network {
      initialize parameters
}

while (current simulation time < time limit for simulation){
      for each node{
            wait for event to send;
            if node has data to send
                  send
            else
                  get next transmission time reserved especially for
                  node to send
       }
}
```

The algorithm dedicates a time-slot interval of 0.1 seconds for any node in the network to send its packet, then moves on to the next node in the network for it to send its data in the next 0.1 second time slot.

### 3.3 Implementation of p-Persistent CSMA

The p-Persistent CSMA algorithm as implemented in this study is briefly explained in this section. A node that is ready to transmit data senses the channel. If the channel is idle, it sends the data immediately. If the channel is not idle, it continuously senses the channel until it becomes idle. The node then sends the data with a probability p. The p value used in this study was 0.01. If the node does not transmit, it tries again at the next transmission time available to it with the same probability to transmit. The process of waiting to send with a probability repeats until the channel is found to be busy or when the node finally transmits the data it holds. The algorithm begins again after either of the last two mentioned scenarios happens.

An exponential back-off algorithm was implemented to resolve collisions that occur with this CSMA protocol. With this algorithm, a random number representing the transmission delay is selected between 0 and $2^n - 1$ for every n collisions that any particular node has encountered.

### 3.4 Implementation of Particle Swarm Optimization Scheme

The algorithm provided in section 2.1 of this paper of the Particle Swarm Optimization MAC scheme was followed. With respect to the values of the constants in the fitness formula, the original values used in the work of Mickus, Clarke, & Mitchell (2015) were used. The values are shown in Table 3.1 below.

Table 3.1: PSO constant values

| Constant | Value |
|---|---|
| w | 0.5 |
| $v_c$ | 0.3 |
| $r_c$ | 0.02 |

Different values of the inertia weight, w, were tested to find out the effect it had on the performance of the PSO protocol.

### 3.5 Implementation of p-Persistent Particle Swarm Optimization Scheme

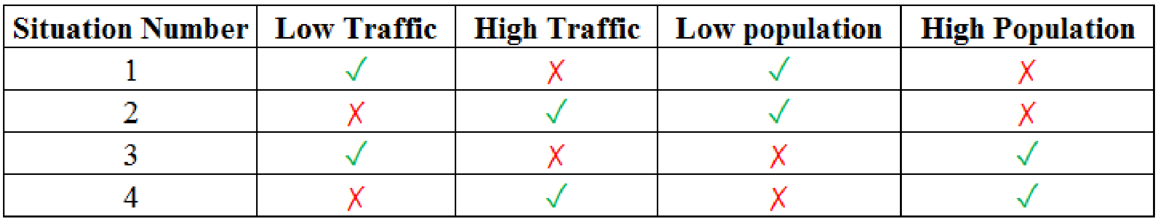
Motivated by the algorithmic concept in the p-Persistent CSMA protocol, a variation of the PSO scheme was implemented where nodes in the network send their data with a probability p after sensing an idle channel. The difference between this algorithm and the p-Persistent CSMA is that the back-off times or transmission delays are calculated with the swarming intelligence algorithm discussed in chapter 2 rather than the exponential back-off algorithm.

### 3.6 The Simulation Environment

To draw valid conclusions on the performance of each protocol implemented, their behaviours were observed in four kinds of situations.

- Situation 1 involved observing the behaviour of the protocol in a low populated network (a network with few nodes) with low traffic. Low traffic implies that each node in the network generated and transmitted packets at a low rate.
- Situation 2 was to observe the protocol in a low populated network with high traffic.
- Situation 3 was to observe the behaviour in a highly populated network with low traffic.
- Situation 4 involved observing the behaviour in a highly populated network with high traffic.

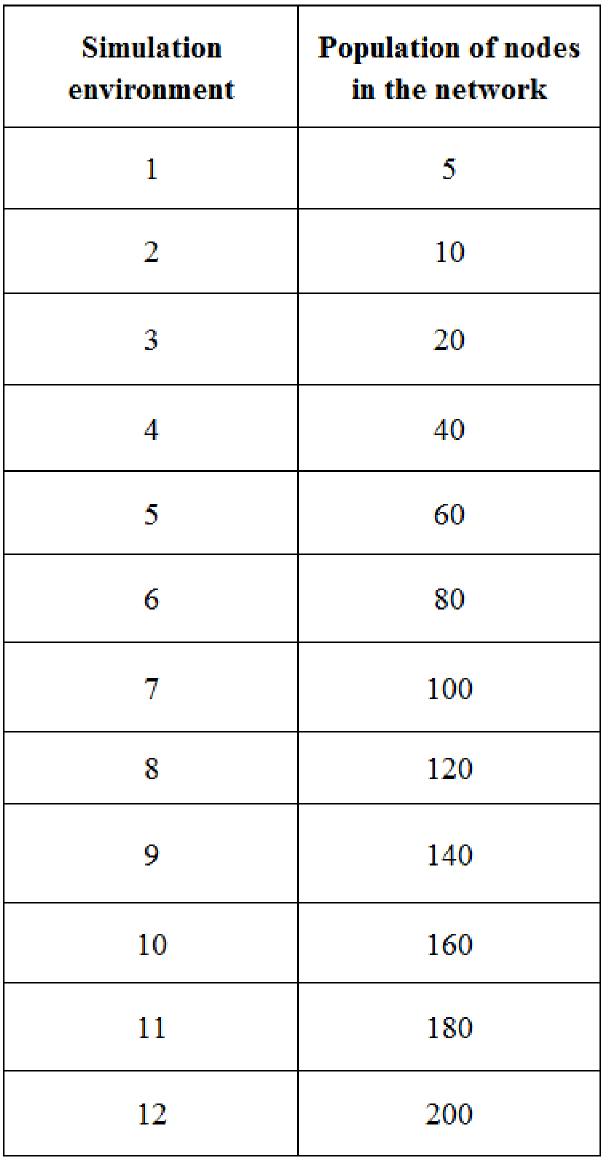
Table 3.2: Summary of target situations for simulation



| Situation Number | Low Traffic | High Traffic | Low population | High Population |
|---|---|---|---|---|
| 1 | ✓ | ✗ | ✓ | ✗ |
| 2 | ✗ | ✓ | ✓ | ✗ |
| 3 | ✓ | ✗ | ✗ | ✓ |
| 4 | ✗ | ✓ | ✗ | ✓ |

To achieve these conditions, twelve simulation environments with varying number of nodes were created. Table 3.3 shows the varying population of sensor nodes used in the simulation.

Table 3.3: The population of nodes used in each of the twelve simulations



| **Simulation environment** | **Population of nodes in the network** |
|---|---|
| 1 | 5 |
| 2 | 10 |
| 3 | 20 |
| 4 | 40 |
| 5 | 60 |
| 6 | 80 |
| 7 | 100 |
| 8 | 120 |
| 9 | 140 |
| 10 | 160 |
| 11 | 180 |
| 12 | 200 |

Figure 3.1 shows a visualization of the simulation environment with 20 nodes and 1 sink node.

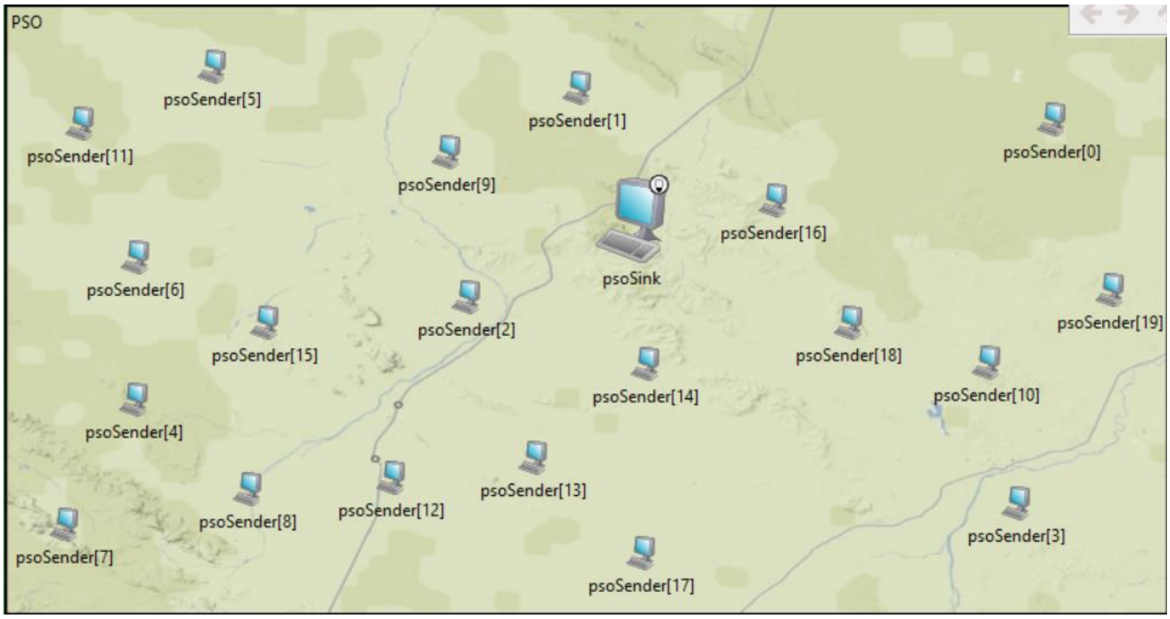


Figure 3.1: Visualization of the simulation environment with 20 nodes and 1 sink node.

The size of each packet generated by a node was 960 bits. The transmission rate for the network was 9.6 kilobits/sec. The time it takes for a packet to travel from the source node to the base station therefore evaluates to 0.1 seconds. To simulate a very high traffic network, each node generated its packet at an inter-arrival rate of 0.1 seconds. This implied that any node in the network could generate packets between the time interval t > 0.0 seconds and t ≤ 0.1 seconds after its last transmission. For a low traffic network, the inter-arrival rate was set to 10 seconds. There is only one sink or base station in the network and the simulation was run for 30 minutes. Table 3.4 and 3.5 summarize the assumptions made for the network simulation.

Table 3.4: Summary of network parameters

| **Parameter** | **Value** |
|---|---|
| Size of packets | 960 bits |
| Transmission rate | 9.6 kilobits/second |
| Duration of packet transmission | 0.1 seconds |
| Simulation runtime | 30 minutes |
| Number of sinks | 1 |

Table 3.5: Inter-arrival time values

| | **High traffic** | **Low traffic** |
|---|---|---|
| **Inter-arrival rate** | 0.1 seconds | 10 seconds |

Each of the twelve simulation environments were run at both high traffic and low traffic. For every environment, 10 trials were recorded. The statistics obtained were averaged to perform the analyses.

# Chapter 4: Analysis of Experimental Results

In this chapter, the results from the simulation are analysed and interpreted. The analysis of the results are done based on the measures mentioned in section 1.3 of this paper. The outline for this chapter is as follows:

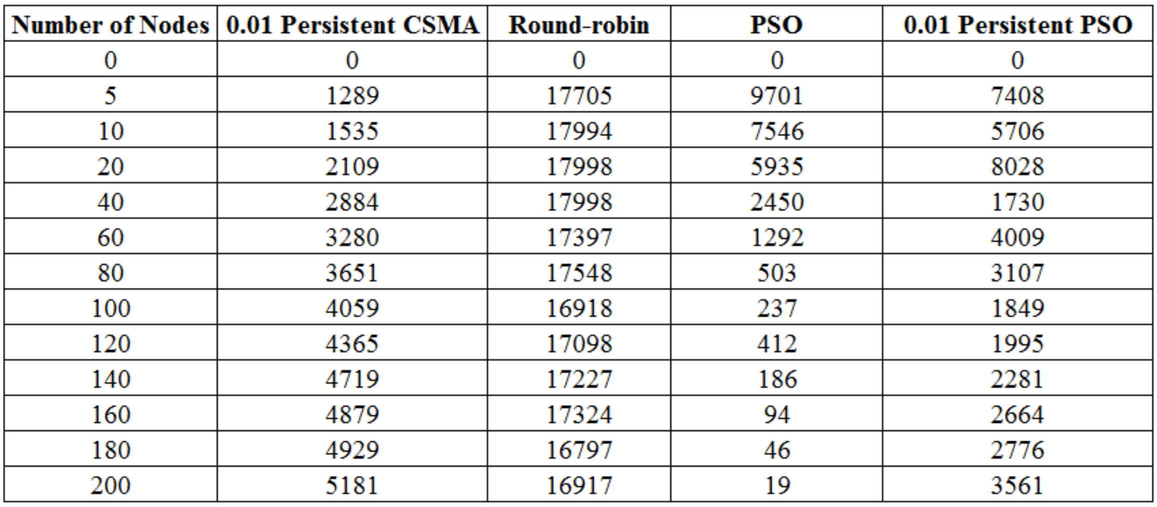
- Section 4.1 presents the throughput analysis of each of the algorithms implemented for both high and low traffic networks
- Section 4.2 discusses the effects of each algorithm on the transmission delay of the nodes in the network for both high and low traffic networks
- Section 4.3 focuses on the PSO scheme and examines the effects of varying the inertia weight, w, in the algorithm

## 4.1 Throughput Analysis

### 4.1.1 Throughput Analysis for a High Traffic Network

Table 4.1 shows the number of packets successfully delivered to the sink node for each of the twelve simulation environments at high traffic for each of the protocols implemented. Figure 4.1 is the corresponding graph to the table.

Table 4.1: Throughput of the protocols at high traffic (inter-arrival time = 0.1seconds)



| Number of Nodes | 0.01 Persistent CSMA | Round-robin | PSO | 0.01 Persistent PSO |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 5 | 1289 | 17705 | 9701 | 7408 |
| 10 | 1535 | 17994 | 7546 | 5706 |
| 20 | 2109 | 17998 | 5935 | 8028 |
| 40 | 2884 | 17998 | 2450 | 1730 |
| 60 | 3280 | 17397 | 1292 | 4009 |
| 80 | 3651 | 17548 | 503 | 3107 |
| 100 | 4059 | 16918 | 237 | 1849 |
| 120 | 4365 | 17098 | 412 | 1995 |
| 140 | 4719 | 17227 | 186 | 2281 |
| 160 | 4879 | 17324 | 94 | 2664 |
| 180 | 4929 | 16797 | 46 | 2776 |
| 200 | 5181 | 16917 | 19 | 3561 |

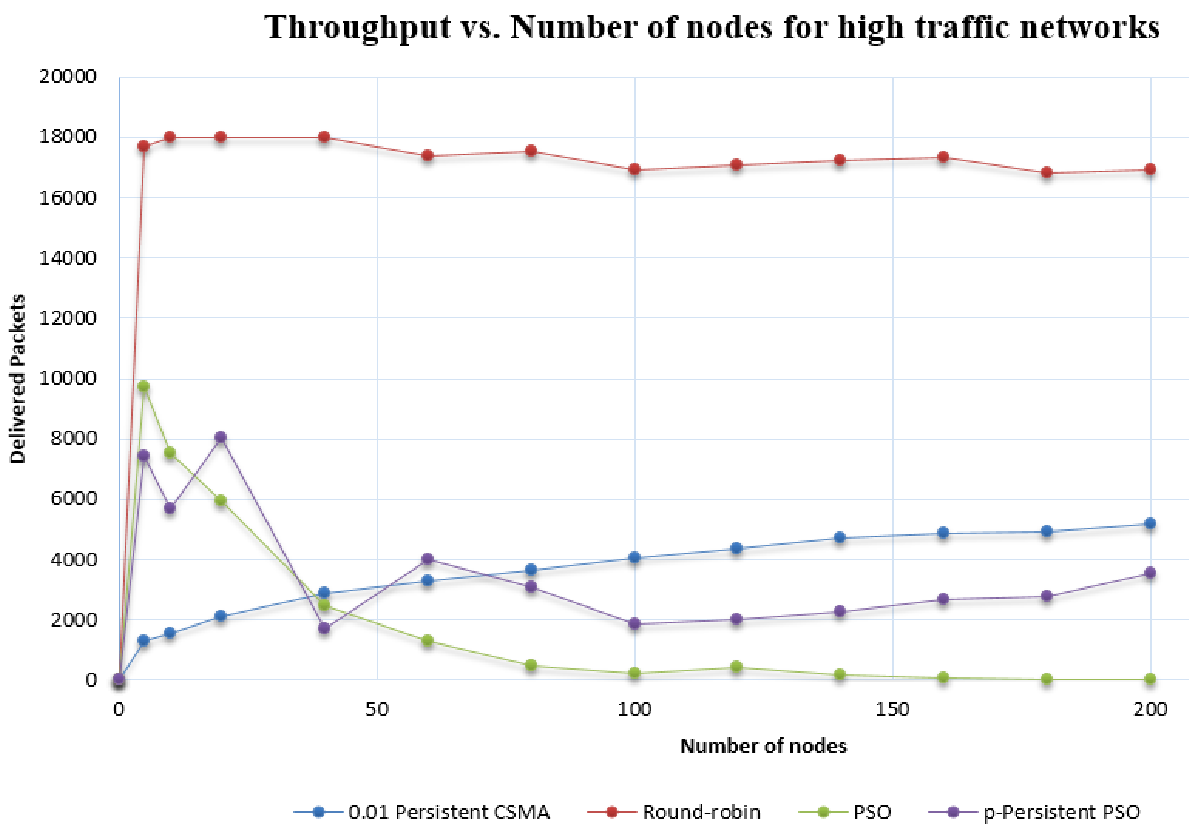


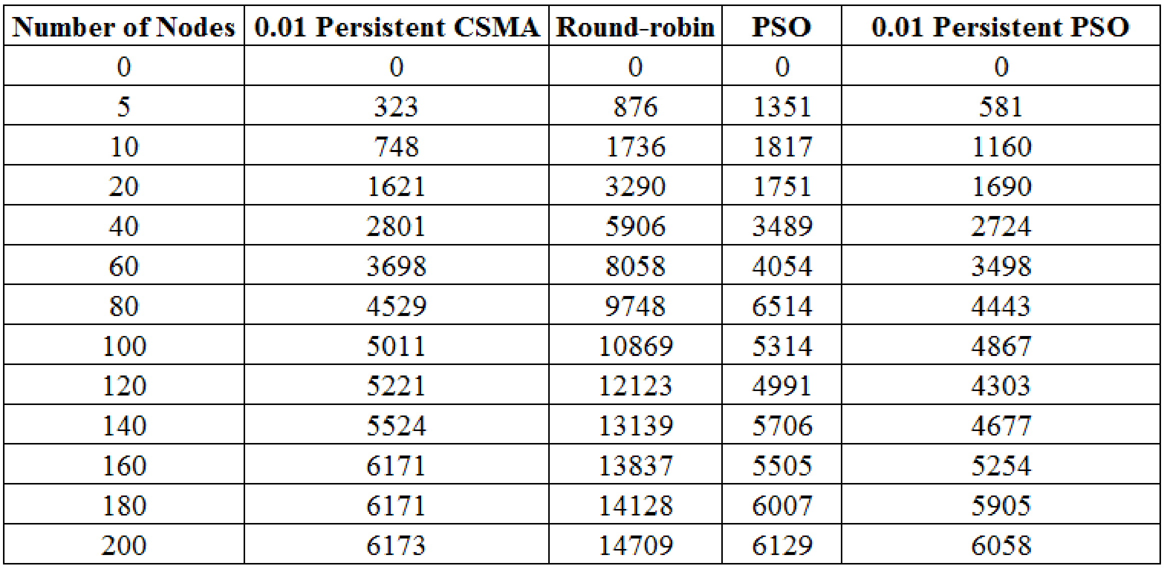
Figure 4.1: Throughput vs. number of nodes for high traffic networks

The results show that round-robin performs better than the other protocols in terms of throughput when the load in the network is very high. This result is consistent with both low and highly populated networks with high traffic. The PSO scheme shows better throughput performance than both 0.01 persistent CSMA and 0.01 persistent PSO when the network population is low. As the number of nodes in the network increases, the throughput performance of the PSO scheme declines considerably. This drop in throughput performance of the PSO can be associated with the implementation of the algorithm. All nodes on the network for the PSO scheme send their packet as soon as the channel is idle. Even with the relative difference in back-off times for the nodes as well as the feedback from previous outcomes of transmissions, the results show that, when the inter-arrival time for packets is high and the population is also very high, the PSO fails to perform adequately. The 0.01 persistent PSO scheme delivers better throughput than the PSO scheme in highly populated networks because it holds off from transmitting even when the back-off time is over for nodes. This probabilistic holds off helps to reduce the number of collisions and increase throughput as shown from the results.

### 4.1.2 Throughput Analysis for a Low Traffic Network

Figure 4.2 and Table 4.2 show the throughput performance of the protocols in a low traffic network. The values in Table 4.2 are the number of data packets successfully delivered to the sink for each network.

Table 4.2: Throughput of the protocols at low traffic (inter-arrival time = 10 seconds)



| Number of Nodes | 0.01 Persistent CSMA | Round-robin | PSO | 0.01 Persistent PSO |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 5 | 323 | 876 | 1351 | 581 |
| 10 | 748 | 1736 | 1817 | 1160 |
| 20 | 1621 | 3290 | 1751 | 1690 |
| 40 | 2801 | 5906 | 3489 | 2724 |
| 60 | 3698 | 8058 | 4054 | 3498 |
| 80 | 4529 | 9748 | 6514 | 4443 |
| 100 | 5011 | 10869 | 5314 | 4867 |
| 120 | 5221 | 12123 | 4991 | 4303 |
| 140 | 5524 | 13139 | 5706 | 4677 |
| 160 | 6171 | 13837 | 5505 | 5254 |
| 180 | 6171 | 14128 | 6007 | 5905 |
| 200 | 6173 | 14709 | 6129 | 6058 |

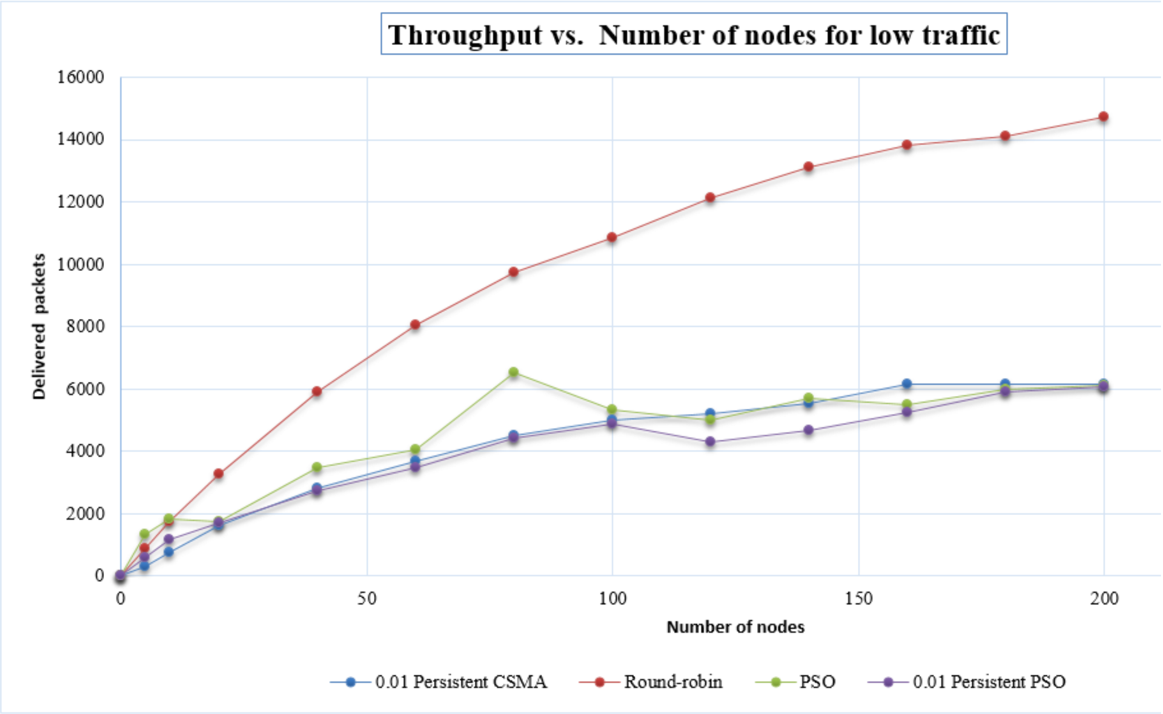


Figure 4.2: Throughput vs. number of nodes for low traffic networks

The results reveal that the PSO algorithm performs better in terms of throughput in a low traffic network than it did in the high traffic network. As the population of nodes in a low traffic network rises, the throughput of the PSO scheme rises as well. The PSO scheme shows better performance than the round-robin protocol when both the traffic and the number of nodes in the network is low. For example, when number of nodes in the network is 5 for a low traffic network, the PSO scheme allows 1351 packets to be successfully delivered to the sink. This throughput value is the greatest among all four algorithms implemented for a 5-node network with low traffic. Round-robin delivers the second highest number of packets at a value of 851. The round-robin algorithm surpasses throughput performance of the PSO scheme as the population in the network rises. Even on a low network, the relative traffic in the network increases when the population increases. The results displayed by highly populated networks with low traffic is therefore very similar to low populated networks with high traffic. The PSO scheme also delivers better throughput than both the 0.01 persistent CSMA and 0.01 persistent PSO protocols when the population in a low traffic network is low. The low performance of the 0.01 persistent CSMA and 0.01 persistent PSO protocols at low load can be associated to the probabilistic hold-off of packets that nodes in the network have. The channel remains idle most of the time because none of the nodes send their packets to the sink. At low loads, there is poor utilization of the channel by the 0.01 persistent CSMA and 0.01 persistent PSO protocols. As the population increases, all three algorithms (PSO, 0.01 persistent CSMA and 0.01 persistent PSO) appear to be at par with one another in terms of throughput performance. The round-robin protocol still shows better throughput performance than all three algorithms when network population is very high. The throughput results shown so far also suggests that when the inter-arrival time becomes larger, both PSO schemes show relatively better throughput performance when compared to lower inter-arrival times on a network with the same number of nodes. On other hand, throughput performance declines as the inter-arrival time decreases on a network with the same number of nodes with respect to the round-robin algorithm.

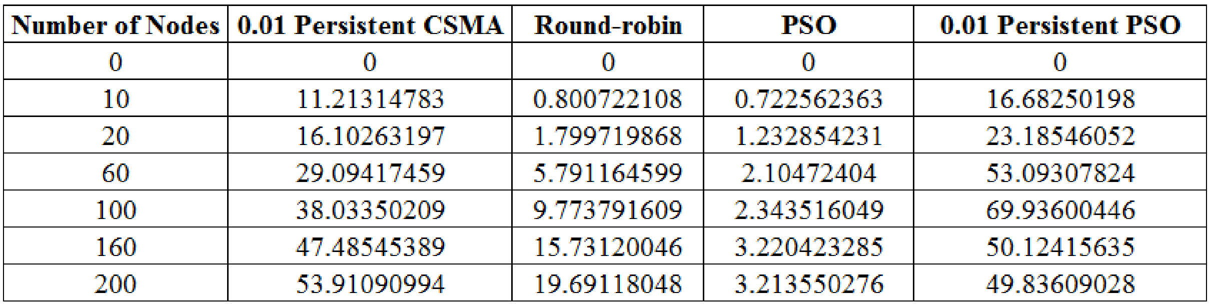
## 4.2 Delay/ Back-off time Analysis

For delay performance, any protocol that shows lower values is preferred and considered to perform better in this respect. Lower values for delay performance implies that sensor nodes in the network do not hold unto data packets they need to send to the sink for too long.

### 4.2.1 Delay/ Back-off time Analysis for a High Traffic Network

For a high traffic network the PSO scheme shows the best delay performance in any number of network population. Table 4.3 shows the average delay of nodes in seconds in networks with varying number of nodes. From this table, it is seen that in a 200-node network, the average time each node waits to send its packet with the PSO scheme is 3.2 seconds. This value indicates that delay in the PSO scheme is very low when compared to 53.9 seconds for 0.01-persistent CSMA, 19.7 seconds for round-robin and 49.8 seconds for 0.01-persistent PSO for the same 200-node network. The round-robin algorithm is still efficient in such an environment because it provides higher throughput performance with considerably low delay in a highly populated network with high traffic as shown in Figure 4.3.

Table 4.3: Average delay of nodes in seconds in a network with high traffic



| Number of Nodes | 0.01 Persistent CSMA | Round-robin | PSO | 0.01 Persistent PSO |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 10 | 11.21314783 | 0.800722108 | 0.722562363 | 16.68250198 |
| 20 | 16.10263197 | 1.799719868 | 1.232854231 | 23.18546052 |
| 60 | 29.09417459 | 5.791164599 | 2.10472404 | 53.09307824 |
| 100 | 38.03350209 | 9.773791609 | 2.343516049 | 69.93600446 |
| 160 | 47.48545389 | 15.73120046 | 3.220423285 | 50.12415635 |
| 200 | 53.91090994 | 19.69118048 | 3.213550276 | 49.83609028 |

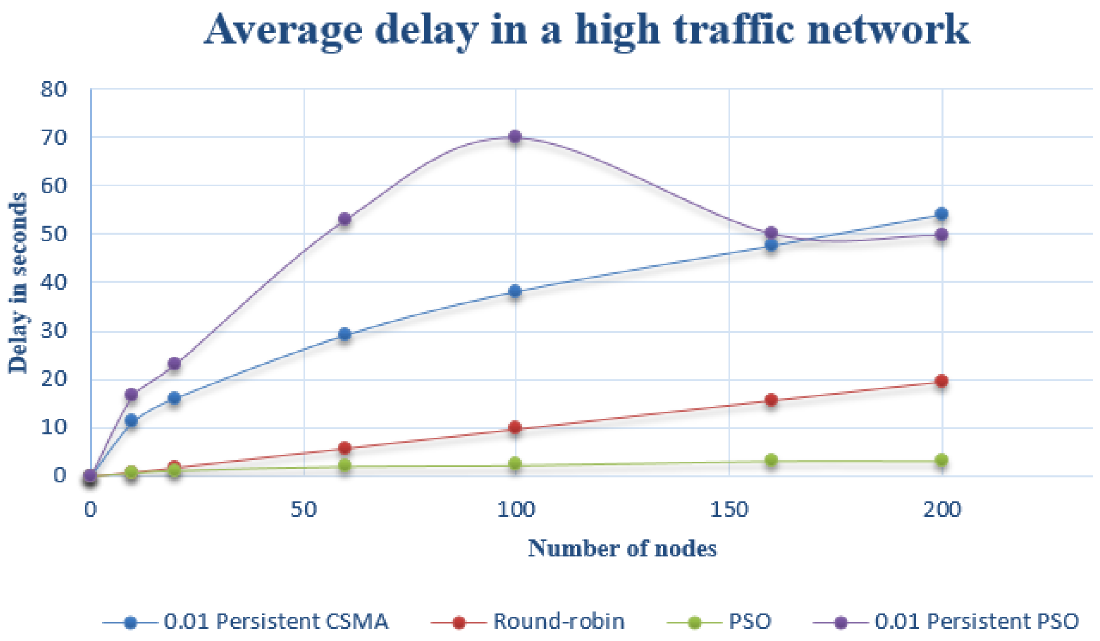


Figure 4.3: Average delay in a high traffic network at low

Both 0.01 persistent CSMA and 0.01 persistent PSO exhibits poor delay performance in a high traffic network. On average, nodes in the network wait longer to send the data packets they have because of the probabilistic hold-off in the algorithm. 0.01 persistent PSO consistently performs the worst in terms of delay in a traffic network as the network population increases. However, 0.01 persistent CSMA exhibits the worst performance in terms of delay in a 200-node network at high traffic.

### 4.2.2 Delay/ Back-off time Analysis for a Low Traffic Network

From Table 4.4 and Figure 4.4, it is evident that the delay for round-robin at low loads is lower than the other three protocols. The 0.01 persistent CSMA shows the worst delay performance in a low populated network with low traffic. For instance, in a 20-node network, each node in the network has an average waiting time of 30 seconds. Comparing this delay value to 1.0 second for round-robin, 20.7 seconds for PSO and 22.4 seconds for 0.01 persistent PSO for the same 20-node network, the 0.01 persistent CSMA has the worst delay.

Table 4.4: Average delay in networks with low traffic

| Number of Nodes | 0.01 Persistent CSMA | Round-robin | PSO | 0.01 Persistent PSO |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 10 | 23.98963803 | 0.496024306 | 10.01934519 | 15.94712176 |
| 20 | 30.0483495 | 1.014684886 | 20.7357939 | 22.38744528 |
| 60 | 31.49437856 | 3.270108212 | 25.98385124 | 32.67818517 |
| 100 | 35.96211575 | 5.796148195 | 32.41460398 | 39.29561439 |
| 160 | 44.12117611 | 9.9644297 | 49.17001453 | 47.50411299 |
| 200 | 50.23910459 | 13.07755914 | 53.0141979 | 50.78657219 |

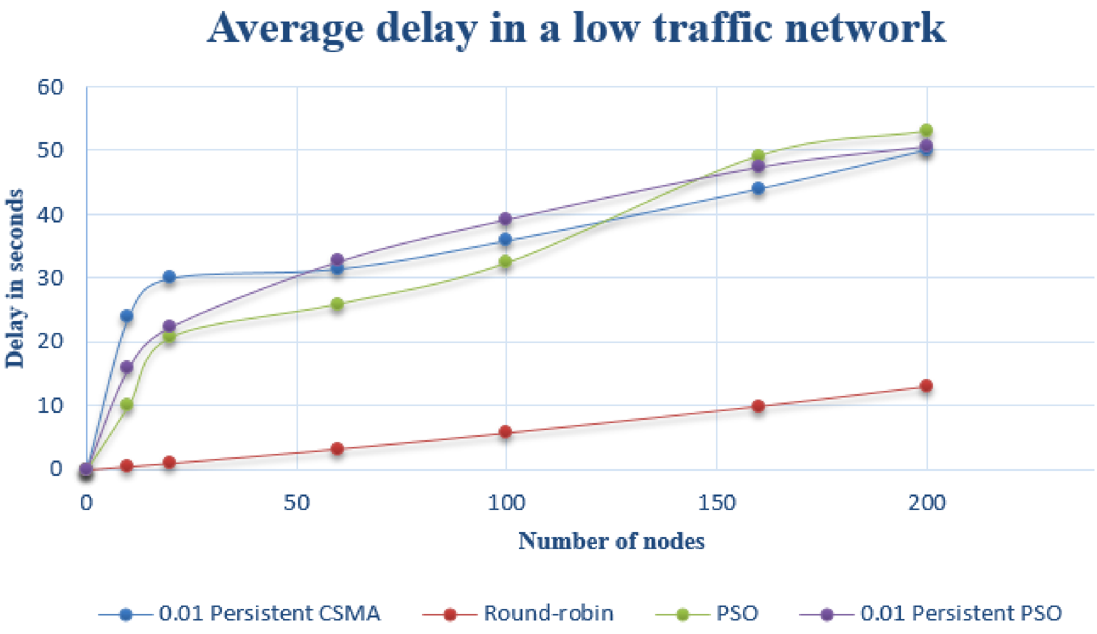
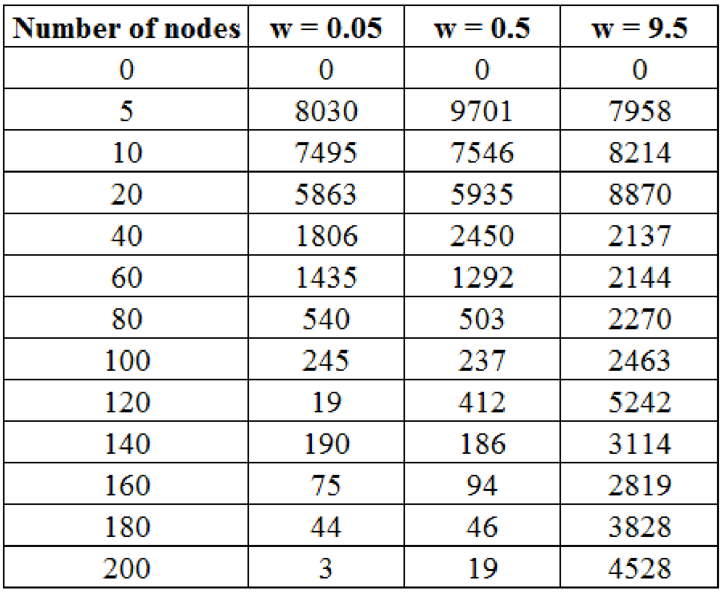


Figure 4.4: Average delay in networks with low traffic

For networks with populations greater than 20, the delay for round-robin at low traffic is still lower than all the other three algorithms. Both versions of the PSO protocol show similar delays for any network population with low traffic with the original PSO protocol showing slightly better delay performance.

### 4.3 Changing Inertia Weight of PSO

The inertia weight (w) used in the update formula of the PSO scheme has a considerable effect on the throughput performance of the protocol. Table 4.5 shows the weights used and the average number of successful packets delivered to the sink that was recorded from 10 runs of the various simulation environments. Figure 4.5 presents the average throughput recorded for a high traffic network with varying inertia weights for the scheme.



| Number of nodes | w = 0.05 | w = 0.5 | w = 9.5 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 5 | 8030 | 9701 | 7958 |
| 10 | 7495 | 7546 | 8214 |
| 20 | 5863 | 5935 | 8870 |
| 40 | 1806 | 2450 | 2137 |
| 60 | 1435 | 1292 | 2144 |
| 80 | 540 | 503 | 2270 |
| 100 | 245 | 237 | 2463 |
| 120 | 19 | 412 | 5242 |
| 140 | 190 | 186 | 3114 |
| 160 | 75 | 94 | 2819 |
| 180 | 44 | 46 | 3828 |
| 200 | 3 | 19 | 4528 |

Table 4.5: The throughput recorded for a high traffic network with varying inertia weights for the PSO scheme

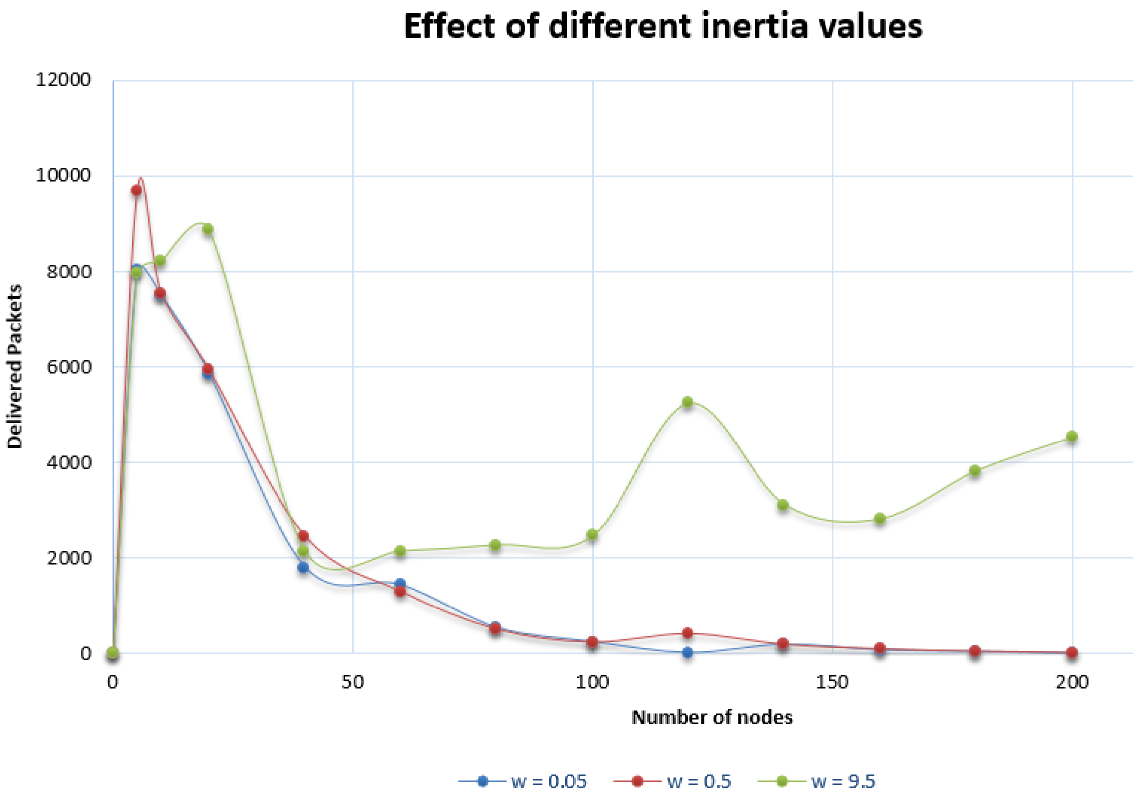


Figure 4.5: Throughput vs. number of nodes for different inertia weights

The experiments revealed that increasing the inertia weight improved the throughput performance of the PSO scheme. Very low inertia weights result in low throughput performance of the PSO scheme because current back-off times are only changed slightly from the previous back-offs. From previous results obtained from the simulation, it can also be inferred that an extremely high inertia weight will eventually reduce the throughput performance of the network. A large inertia weight implies that the transmission delays for each node will be changed greatly by the algorithm with each feedback received. Big changes to the transmission delay suggest that the channel would remain idle most of the time since nodes in the network would wait too long to transmit.

# Chapter 5: Conclusion and Recommendation

## 5.1 Conclusion

In this project, the round-robin protocol and the PSO protocol as proposed by Mickus, Clarke, & Mitchell (2015) were implemented and their performances evaluated. A p-persistent CSMA with a p value of 0.01 and a variation of the PSO scheme that uses the concept of the p-persistent CSMA were also analysed. These four protocols were simulated with OMNET++. It was discovered that the PSO algorithm performed poorly in terms of throughput on a highly populated network with high traffic when compared to the other protocols implemented. Round-robin showed the best throughput performance in a highly populated network with high traffic. On low populated networks with low traffic, the PSO scheme performs the best in terms of throughput. The delay performance of the PSO scheme in a highly populated network with high traffic is better than the other three algorithms. However, round-robin shows the best delay performance for highly populated networks with low traffic. The p-persistent PSO performed better than the original PSO in terms of throughput when the traffic on a densely populated network was high. It was also discovered that a low inertia weight dampened the performance of the PSO in terms throughput. On the other hand, extremely high inertia weight values could dampen the PSO scheme's performance.

From the experimental results in this project, it can be concluded that any form of 0.01 persistent algorithm shows poor delay performance since the probabilistic hold-off of packets makes the channel remain idle most of the time. The performance of the genetic PSO MAC scheme is also highly dependent on the base CSMA protocol used.

## 5.3 Future work

Possible work that could be done to further this study is to simulate the algorithms with the goal of finding the effect the protocols have on energy consumption in sensor nodes.

There exists a field of genetic algorithm based MAC protocols that uses the idea of putting the entire network system into clusters and making the nodes work to optimize performance in the clusters. The Ant Colony Optimization algorithm is also one possible genetic algorithm that can be integrated into MAC protocols with the goal of optimizing its performance. Exploring how the schemes implemented in this work compare to the cluster-based genetic protocols and the Ant Colony Optimization algorithm would further this study.

# References

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems.* New York: Oxford University Press.

Dorigo, M., & Birattari, M. (2010). Ant colony optimization. In *in Encyclopedia of Machine Learning* (pp. 36-39). New York City: Springer.

Egea-López, E., & Vales-Alonso, J. (2007). Performance Evaluation of Non-persistent CSMA as Anti-collision Protocol for Active RFID Tags. *Wired/Wireless Internet Communications, 4517*, 279-289.

Fan, J., & Parish, D. J. (2007). Using a genetic algorithm to optimize the performance of a wireless sensor network. *The 8th Annual Postgraduate Symposium, The Convergence of Telecommunications, Networking and Broadcasting, Liverpool John Moores University, 28th-29th June.* Citeseer.

Kennedy, J. (2010). Particle swarm optimization. In *In Encyclopedia of machine learning* (pp. 760-766). New York: Springer.

Krishnamachari, B. (2005). An introduction to wireless sensor networks. *Second International Conference on Intelligent Sensing and Information Processing (ICISIP).* Chennai.

Libelium Comunicaciones Distribuidas S.L. (2015, May). *50 sensor applications for a smarter world.* Retrieved from Libelium: http://www.libelium.com/top_50_iot_sensor_applications_ranking/

Mahalik, N. P. (2007). *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications.* Berlin: Springer.

Mickus, T., Clarke, T., & Mitchell, P. (2015). Swarming medium access control protocol for large-scale wireless sensor networks. *Next Generation Mobile Applications,* *Services and Technologies 2015 9th International Conference on* (pp. 102-107). IEEE.

Mitchell, M. (1998). *An introduction to genetic algorithms.* Cambridge: Bradford Books.

Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm-a novel tool for complex optimisation problems. *in Proceedings of the 2nd virtual international conference on Intelligent Production Machines and Systems (IPROMS)*, (pp. 454-459).

Qiao, G., Yang, J., Ma, X., Zhang, Y., & Dong, H. (2013). Simulation and experimental verification of MAC protocols for underwater acoustic communication networks. *In Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems*, (pp. 1-5). Kaohsiung, Taiwan.

Stankovic, J. (2006). *Wireless sensor networks.* University of Virginia, Department of Computer Science, Charlottesville.

Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer networks.* Boston: Pearson Education, Inc.

Xu, G., Shen, W., & Wang, X. (2014, September). Applications of wireless sensor networks in marine environment monitoring: A survey. *Sensors, 14*(9), 16932-16954. doi:10.3390/s140916932

Yassein, B., Manaseer, S., & Al-Turani, A. (2009). *A performance comparison of different backoff algorithms under different rebroadcast probabilities for MANET's.* Department of Computer Science, Jordan University of Science and Technology.