

Article

## Maximization of the Supportable Number of Sensors in QoS-Aware Cluster-Based Underwater Acoustic Sensor Networks

Thi-Tham Nguyen, Duc Van Le and Seokhoon Yoon \*

Department of Electrical and Computer Engineering, University of Ulsan, Ulsan 680-749, South Korea; E-Mails: nttham0611@gmail.com (T.-T.N.); anhduc.mta@gmail.com (D.V.L.)

\* Author to whom correspondence should be addressed; E-Mail: seokhoonyoon@ulsan.ac.kr; Tel.: +82-52-259-1403; Fax: +82-52-259-1687.

Received: 15 January 2014; in revised form: 27 February 2014 / Accepted: 4 March 2014 / Published: 7 March 2014

---

**Abstract:** This paper proposes a practical low-complexity MAC (medium access control) scheme for quality of service (QoS)-aware and cluster-based underwater acoustic sensor networks (UASN), in which the provision of differentiated QoS is required. In such a network, underwater sensors (U-sensor) in a cluster are divided into several classes, each of which has a different QoS requirement. The major problem considered in this paper is the maximization of the number of nodes that a cluster can accommodate while still providing the required QoS for each class in terms of the PDR (packet delivery ratio). In order to address the problem, we first estimate the packet delivery probability (PDP) and use it to formulate an optimization problem to determine the optimal value of the maximum packet retransmissions for each QoS class. The custom greedy and interior-point algorithms are used to find the optimal solutions, which are verified by extensive simulations. The simulation results show that, by solving the proposed optimization problem, the supportable number of underwater sensor nodes can be maximized while satisfying the QoS requirements for each class.

**Keywords:** supportable number of nodes; QoS; optimization; underwater acoustic sensor network

---

## 1. Introduction

As an emerging technique, underwater acoustic sensor networks (UASN) have a wide range of applications, such as oceanographic data collection, environment monitoring, undersea exploration, disaster prevention, assisted navigation and tactical surveillance [1–5]. In order to implement these applications, underwater nodes communicate with each other via acoustic channels that have unique characteristics, including the limited available bandwidth and a high and variable propagation delay [6–9].

In this paper, we consider a UASN that has a cluster-based network topology, in which each cluster is governed by a clusterhead (or gateway node), since it makes the network scalable and can readily provide network connectivity in a harsh communication environment [5,10–13]. In addition, the considered UASN consists of different types of underwater sensor nodes, some of which generate more important data than others, *i.e.*, the sensing data from some sensors may need to be delivered to the clusterhead with a higher PDR (packet delivery ratio). Therefore, the network needs to provide the sensor nodes with differentiated QoS (quality of service) in terms of PDR based on the QoS class to which the sensor nodes belong.

In such a network, an important problem is to maximize the number of nodes that the network can accommodate while still providing the required QoS for each class. In addition, as a related problem, when the operators deploy a UASN, they would want to know the achievable PDR value given the number of sensor nodes in the network. Intuitively, if the number of nodes in a UASN increases beyond a specific amount, the network may not be able to provide the demanded QoS, due to a high level of network traffic.

In order to address the problem of maximizing the supportable number of nodes, we focus on the MAC (medium access control) layer, since it plays a key role for providing QoS and dominates the overall performance of the network [14]. In particular, contention-based MAC protocols have received a lot of attention, due to the simplicity and applicability in UASNs [15–29]. Among various contention-based MAC protocols, Aloha-CS (Aloha with carrier sensing) is a potential low-complexity protocol for UASNs, since it offers a high throughput and low latency in a low network load without requiring time synchronization or a handshaking mechanism [18–20].

In this paper, we design a practical low-complexity QoS-aware MAC scheme and an optimization formulation for maximizing the supportable number of sensors in UASNs. We first estimate the packet delivery probability (PDP) in the MAC layer. Then, based on the PDP estimation, an optimization problem is formulated for maximizing the supportable number of sensors in a specific QoS priority class. The main idea of the formulation is to find optimal values of the maximum packet retransmissions for each QoS class, such that the number of nodes in a specific QoS class is maximized and every node can achieve the required QoS.

The custom greedy and interior-point algorithms are used to find the solutions to the optimization problem. Furthermore, extensive simulations are performed to verify the solutions. The simulation results show that our optimization formulation can maximize the supportable number of underwater sensor nodes, while satisfying the QoS requirement for each class.

The rest of this paper is organized as follows. Section 2 presents the related studies and compares them with the proposed scheme. The system model and problem definition are described in Section 3. Section 4 first discusses the packet delivery probability approximation, then describes the optimization problem formulation. We also discuss the approximation of the background traffic. The performance analysis using various scenarios is presented in Section 5, in which we also discuss solutions and the simulation setup. Finally, Section 6 concludes the paper.

## 2. Related Work

MAC protocols for UASN can be categorized into contention-free and contention-based protocols. The contention-free protocols include time division multiple access (TDMA), frequency division multiple access (FDMA) and code division multiple access (CDMA), in which different time slots, frequency bands or codes are assigned to different users to avoid collisions among transmissions.

FDMA divides the available frequency band into several sub-bands and assigns each sub-band to a node. Due to the limited available bandwidth of underwater channels, FDMA is not suitable for UASNs that consist of a large number of underwater sensors.

In TDMA, in order to avoid the collision of packets from adjacent time slots, guard times are added to the time slot. The high propagation delay in underwater acoustic communication channels requires long guard times, which limit the efficiency of TDMA [30]. Moreover, TDMA systems require precise synchronization for proper utilization of the time slots.

It is also known that CDMA-based protocols require a high complexity design for UASN. In addition, it is a challenging problem to assign pseudo-random codes to a large number of sensor nodes [2].

On the other hand, contention-based protocols have received significant attention for UASN, due to their simplicity, acceptable throughput and energy efficiency [15–23]. For example, the authors of [15] studied the performance of Aloha-based protocols in underwater networks and proposed two enhanced schemes that take advantage of the long propagation delay in the underwater acoustic channel and do not require handshaking or time synchronization.

It was also shown that, under the high and varying propagation delay in underwater acoustic channels, the performance of slotted Aloha becomes similar to that of pure Aloha [23]. The study in [16] proposed a propagation delay-tolerant Aloha protocol, where the authors address the space-time uncertainty by adding guard times to slotted Aloha.

Another simple yet practical Aloha-based protocol, Aloha-CS (Aloha with carrier sensing), was also studied and evaluated in [15,18–20]. In Aloha-CS, a node senses the carrier on the channel before it transmits data. The intended receiver sends an acknowledgment (ACK) packet to the source node to announce the successful reception. For unsuccessful transmissions, the retransmission mechanism with an exponential backoff can be also applied, *i.e.*, the data packet can be retransmitted up to a maximum limit of retries unless an ACK packet is received at the source node. According to the results presented in these studies, Aloha-CS (Aloha with carrier sensing) [18–20] can achieve high throughput and low latency without requiring time synchronization or handshaking.

The authors of [21] proposed an extension of the FAMA protocol [31] for UASN, namely slotted FAMA. Slotted FAMA is also based on carrier sensing and handshaking prior to data transmission. The

new idea of slotted FAMA is that it uses time slotting to eliminate the requirement for excessively long control packets. The study in [22] proposed a reservation-based MAC protocol, T-Lohi, where a node sends a short tone to count the number of contenders. If it does not receive any other tones, it starts data transmission. Otherwise, it goes to the backoff mode.

Although our work is also based on channel contention, those studies differ from ours since they do not consider the provision of QoS or optimality.

There are few MAC protocols that address QoS provision in UASNs. However, there have been several MAC protocols that considered QoS provision for wireless sensor networks [32–39].

In particular, the authors of [37] proposed I-MAC, a hybrid TDMA/CSMA-based MAC protocol for wireless sensor networks. The I-MAC protocol is composed of two phases: the setup and transmission phases. During the setup phase, neighbor node discovery, slot assignment, local framing and global synchronization operations are successively performed. If a node owns assigned slots, it transmits data using those slots. If a node does not own any slot, it uses CSMA to access the channel. By using a different value of the CW (contention window), some groups of nodes can have a higher priority for accessing the channel.

As another example, the study in [38] proposed a MAC protocol that supports QoS in wireless sensor networks. It also uses a hybrid scheduling technique where dedicated time slots are assigned for data packet transmissions, and CSMA/CA-based random access periods are used for control packet transmissions. The MAC protocol consists of four phases: time synchronization, request for time slots, reception of slot schedules and data transfer.

However, the studies in [37,38] do not consider satisfying a given QoS requirement. In addition, they require tight time synchronization and overheads for slot requests and assignments. In contrast, the objective of our work is to design a low-complexity MAC scheme that supports differentiated QoS without requiring time synchronization or scheduling overheads.

There also have been attempts to design a QoS-aware MAC protocol based on channel contention for a wireless sensor network. For example, the study in [39] considered a transmitter-only network and proposed a MAC protocol to provide QoS using an optimal number of transmissions. That work also differs from ours, since it considered a network of nodes without an RFreceiver or packet queuing and a fixed number of transmissions of each packet in a given time interval. Moreover, the objective is different from that in our paper.

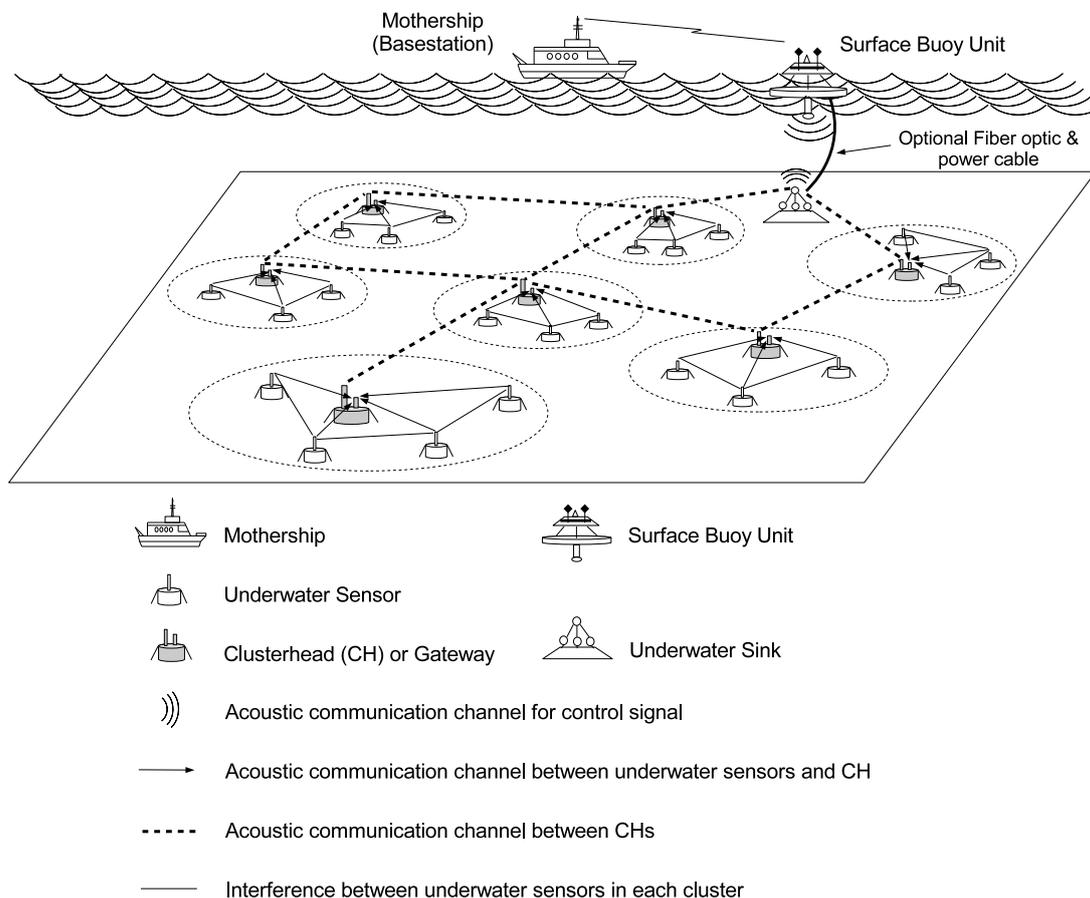
### 3. System Model and Problem Definition

In this paper, we consider a cluster-based UASN, where each cluster is governed by a clusterhead (or gateway node). As shown in Figure 1, each underwater sensor node (or, simply, U-sensor or node) belongs to one cluster. The clusterhead collects sensing data from U-sensors, performs data aggregation/fusion and then forwards the data to the underwater sink node. Clusterheads are equipped with two communication interfaces, so that they can use different channels for communicating with U-sensors and other clusterheads, respectively.

It is assumed that communications in a cluster do not interfere with communications in other clusters, due to the use of different carriers, and U-sensors transmit sensed data to the clusterhead using a direct

acoustic channel [40]. Assigning channels to adjacent clusters or nodes has been considered in several studies [41–45].

**Figure 1.** Cluster-based underwater acoustic sensor network.



U-sensors in a cluster are classified into several QoS classes, each of which has a required packet delivery ratio (PDR). In this paper, required PDR values are used to determine QoS classes. Every node generates a data packet at a predetermined rate and transmits them to the clusterhead. U-sensors in each QoS class are allowed to retransmit each data packet up to the maximum number of retransmissions, unless they receive the corresponding ACK packet from the clusterhead within the ACK timeout interval. Before a U-sensor transmits data, it first performs carrier sensing to assure that the channel is idle. It also performs exponential back-offs when collisions occur.

The considered optimization problem is the maximization of the number of nodes in a specific QoS class, which will be selected by the operators of the network, while providing the QoS for every node in each class.

In order to facilitate discussion, suppose that a set of  $N$  nodes in a cluster is divided into  $m$  QoS classes,  $(Q_1, Q_2, \dots, Q_m)$ , where class  $Q_i$  contains  $n_i$  nodes ( $1 \leq i \leq m$ ). Nodes in each QoS class have a packet size,  $s_i$ , and the corresponding packet transmission delay,  $t_d^i$ . Each node in class  $Q_i$  is allowed to retransmit each data packet up to  $x_i$  times and requires a minimum PDR of  $p_i$ , where  $x_i$  denotes the maximum number of retransmissions. Suppose also that class  $Q_k$  is selected to maximize the number of nodes in the class, where  $1 \leq k \leq m$ .

Therefore, in order to achieve the objective, while providing differentiated QoS to nodes, the core problem is to determine an optimal value of  $x_i$  for each class,  $Q_i$ , such that  $n_k$  is maximized and every node in each class can achieve a PDR of at least  $p_i$ .

#### 4. Maximization of the Supportable Number of Sensors

In this section, we first describe the approximation of the packet delivery probability. Then, we present the formulation of the optimization problem. In addition, we discuss algorithms for finding solutions.

##### 4.1. PDP Approximation

We first define the packet delivery probability (PDP) as the probability that a packet is successfully delivered at the clusterhead when it can be retransmitted up to  $x$  times.

In a UASN, the packet generation rate is usually low, due to the limited bandwidth. In such a network, very few packet losses result from the buffer overflow, since available space is likely when a new packet is generated. Consequently, PDP values can approximate PDR values in a UASN. Therefore, PDP is used in the optimization formulation for PDR.

Now, we discuss the approximation of the PDP of nodes in each class,  $Q_i$ , where a node can retransmit a packet up to  $x_i$  times. In order to approximate the PDP value, we first assume that the packet arrival in a UASN follows a Poisson process, which will also be verified in the following discussion. Then, the probability of  $k$  packet arrivals during an interval of time  $t$  is given by:

$$P[n = k] = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \quad (1)$$

where  $\lambda$  represents the arrival rate of background traffic in a time interval of  $t$  [46].

A U-sensor node in each class,  $Q_i$ , transmits to the clusterhead a data packet in every interval,  $T$ . Suppose that a data packet arrives at the clusterhead at time  $t_0$  with the transmission delay of  $t_d^i$ . In order to avoid collisions for a packet that is transmitted from a node in class  $Q_i$ , no packets from the other  $N - 1$  nodes should arrive at the clusterhead during the interval  $[t_0 - t_d^i, t_0 + t_d^i]$ , i.e., there should be no packet arrival during the interval of  $2t_d^i$ .

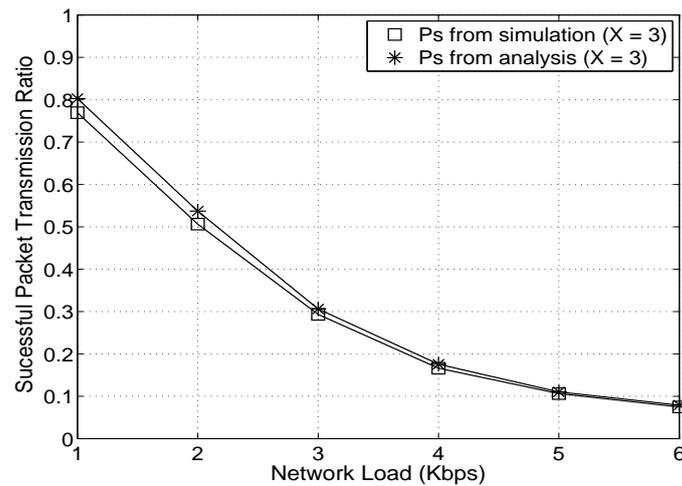
Let  $P_s^i$  and  $P_f^i$  denote the probabilities of the successful and failed packet transmissions of a node in class  $Q_i$  at the clusterhead, respectively, where  $P_f^i = 1 - P_s^i$ . Furthermore, let  $\lambda_b$  denote the arrival rate of the background traffic for a node in an arbitrary class. Then, the probability that a data packet, which is transmitted from a node in class  $Q_i$ , is successfully delivered at the clusterhead is given by:

$$P_s^i = e^{-2\lambda_b t_d^i} \quad (2)$$

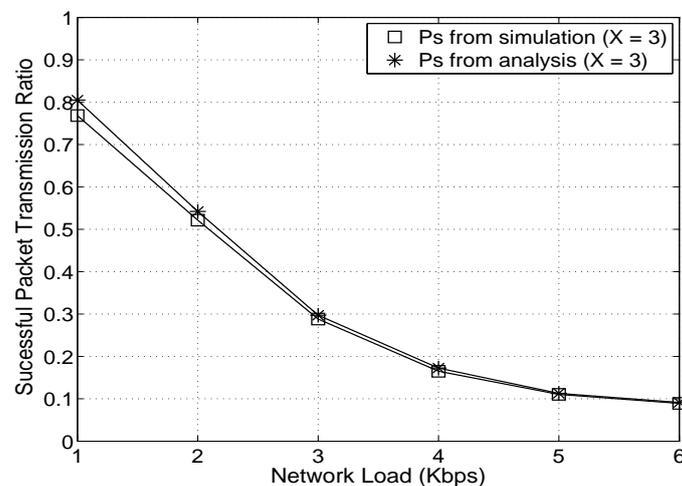
In order to verify the assumption of Poisson distribution of the packet arrival in a UASN, where a node performs carrier sensing and exponential back-offs, we conduct a simple simulation using Aloha and Aloha-CS protocols. The considered cluster in the network consists of 50 U-sensors and one clusterhead that are randomly deployed over an area of  $1,555 \text{ m} \times 1,555 \text{ m}$ . In this example, for simplicity, we assume that there is only one QoS class,  $Q_1$ . Each U-sensor node is equipped with a half-duplex acoustic transceiver that has a data rate of 14 Kbps. Every U-sensor periodically generates a data packet of

160 bytes and sends it to the clusterhead. Each node is allowed to retransmit one data packet up to three times, unless it receives the corresponding ACK packet from the clusterhead. We calculate the probability of successful packet transmission in class  $Q_1$ ,  $P_s^1$ , according to Equation (2), and determine the actual successful individual packet transmission ratio from simulation. Then, we compare the value of  $P_s^1$  from analysis and that from simulation.

**Figure 2.** Approximation of the successful packet transmission ratio. Aloha-CS, Aloha with carrier sensing.



(a) For the case of Aloha



(b) For the case of Aloha-CS

As shown in Figure 2, over different network loads from 1 Kbps to 6 Kbps, the approximation of  $P_s^i$  is fairly similar to the actual successful individual packet transmission ratio. Therefore, in our work, we use the assumption that packet arrivals follow a Poisson process to design the optimization formulation. Now, we define  $P_s^{i,j}$  and  $P_f^{i,j}$  as the probabilities of the successful and failed delivery of the  $j$ -th transmission of a packet of nodes in class  $Q_i$ , respectively. Furthermore, let  $P(x_i)$  denote the PDP that the nodes in class  $Q_i$  can achieve, and recall that one data packet can be retransmitted up to  $x_i$  times. Then,  $P(x_i)$  can be expressed as:

$$P(x_i) = P_s^{i,1} + P_f^{i,1} P_s^{i,2} + \dots + P_f^{i,1} \dots P_f^{i,x_i-1} P_s^{i,x_i} \quad (3)$$

Since each packet transmission can be regarded as an independent event based on the assumption of a Poisson process,  $P_s^{i,j} = P_s^i$  and  $P_f^{i,j} = P_f^i$  for all  $j$  ( $j = 1 \dots x_i$ ). Therefore,  $P(x_i)$  becomes:

$$\begin{aligned} P(x_i) &= P_s^i \frac{1 - (P_f^i)^{x_i}}{1 - P_f^i} \\ &= 1 - (P_f^i)^{x_i} \\ &= 1 - (1 - e^{-2\lambda_b t_d^i})^{x_i} \end{aligned} \quad (4)$$

In the following section, we present an optimization formulation for maximizing the number of sensors in UASN, while satisfying the QoS requirement.

#### 4.2. Optimization Problem Formulation

In this subsection, we describe the proposed optimization problem formulation that is a non-linear optimization problem.

Recall that the nodes in each class,  $Q_i$ , need to guarantee their PDR requirement of at least  $p_i$ . In other words, the approximated PDP of the nodes in each class needs to be at least  $p_i$ . Specifically, the constraint function is expressed as:

$$1 - (1 - e^{-2\lambda_b t_d^i})^{x_i} \geq p_i \quad (5)$$

The actual arrival rate of background traffic for a node in an arbitrary class,  $\lambda_b$ , is the total number of packet arrivals from the other  $N - 1$  nodes in the time interval. It is a challenging problem to calculate the exact value of  $\lambda_b$ , since the actual number of retransmissions for one data packet at a given time depends on the network traffic and status. Therefore, to simplify the problem, we use the maximum arrival rate of background traffic generated by all nodes in the network,  $\lambda_{max}$ . In the following discussion, we prove that the required PDR can be satisfied by using  $\lambda_{max}$ .

In order to calculate the value of  $\lambda_{max}$ , we use the maximum number of retransmissions for each class,  $Q_i$ , which is denoted by  $x_i$ . Then, the maximum arrival rate of background traffic is given by:

$$\lambda_{max} = \sum_{i=1}^m \frac{n_i x_i}{T} \quad (6)$$

Then, the constraint function in which we use the maximum arrival rate,  $\lambda_{max}$ , is given as:

$$1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i} \geq p_i \quad (7)$$

**Lemma 1.** Suppose that we use the maximum arrival rate of background traffic,  $\lambda_{max}$ , to formulate the optimization problem. If we can determine an optimal value of  $x_i$  that satisfies the constraint function in Equation (7), then we can assure that  $x_i$  also satisfies the constraint function in Equation (5).

**Proof.** When we use the actual arrival rate of background traffic for calculating  $P_s$ , then  $P_s^i(\lambda_b) = e^{-2\lambda_b t_d^i}$ . Similarly, when we use the maximum arrival rate of background traffic to calculate  $P_s^i$ , then  $P_s^i(\lambda_{max}) = e^{-2\lambda_{max} t_d^i}$ . From the fact that  $\lambda_{max} \geq \lambda_b$ , we have  $1 - e^{-2\lambda_{max} t_d^i} \geq 1 - e^{-2\lambda_b t_d^i}$ . Note that the value of  $x_i$  is a positive integer. Therefore, we have the following relation:

$$1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i} \leq 1 - (1 - e^{-2\lambda_b t_d^i})^{x_i} \quad (8)$$

According to the constraint function in Equation (7), if we can find a value of  $x_i$  that satisfies Equation (7), then the inequality  $1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i} \geq p_i$  is always true. Combining the relation represented in Equation (8) and the constraint in Equation (7), we can achieve the following relation:

$$p_i \leq 1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i} \leq 1 - (1 - e^{-2\lambda_b t_d^i})^{x_i} \quad (9)$$

As a result, since  $x_i$  satisfies the constraint function in Equation (7) in which the maximum arrival rate of background traffic,  $\lambda_{max}$ , is used, then it also satisfies the constraint function in Equation (5) that uses the actual arrival rate of background traffic,  $\lambda_b$ .  $\square$

Therefore, we have the formulation of  $P(x_i)$  for each class as follows:

$$P(x_i) = 1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i} \quad (10)$$

Now, we describe our optimization problem formulation.

The objective of our optimization problem is to maximize the supportable number of nodes in a class,  $Q_k, n_k$ , where  $k$  is a given integer number from one to  $m$ . Note that  $Q_k$  has the PDR requirement of  $p_k$ . In order to maximize  $n_k$ , we determine the relationships between  $n_k$  and other variables. More specifically, from the fact that  $P(x_k) \geq p_k$ , we have:

$$1 - (1 - e^{-2\lambda_{max} t_d^k})^{x_k} \geq p_k \quad (11)$$

We replace  $\lambda_{max}$  based on Equation (6) to show:

$$1 - (1 - e^{-\frac{2t_d^k}{T}(n_1 x_1 + \dots + n_k x_k + \dots + n_m x_m)})^{x_k} \geq p_k \quad (12)$$

From Equation (12), we obtain the following inequality:

$$n_k \leq -\frac{1}{x_k} \left( \frac{T}{2t_d^k} \ln(1 - (1 - p_k)^{\frac{1}{x_k}}) + \sum_{\substack{i=1 \\ i \neq k}}^m n_i x_i \right) \quad (13)$$

Then, the optimization formulation is that, given  $n_i$  ( $i \neq k$ ) and  $p_1, p_2, \dots, p_m$ , find  $x_1, x_2, \dots, x_m$ , such that:

$$\max \left[ -\frac{1}{x_k} \left( \frac{T}{2t_d^k} \ln(1 - (1 - p_k)^{\frac{1}{x_k}}) + \sum_{\substack{i=1 \\ i \neq k}}^m n_i x_i \right) \right] \quad (14)$$

subject to:

$$(1 - e^{-2\lambda_{max} t_d^i})^{x_i} - (1 - p_i) \leq 0, \quad i = 1 \dots m \quad (15)$$

$$1 \leq x_i \leq l, \quad i = 1 \dots m \quad (16)$$

The constraint in Equation (15) is based on the requirement that the value of  $x_i$  should guarantee  $P(x_i) \geq p_i$ , where  $P(x_i)$  is calculated according to the Equation (10). In addition, the value of  $x_i$  is limited by an upper bound,  $l$ , as impressed in constraint Equation (16).

### 4.3. Finding Solutions

In order to find solutions to the proposed optimization formulation, we use a custom developed greedy algorithm and the interior-point method.

In the greedy algorithm, for each solution vector  $x = (x_1, x_2, \dots, x_m)$ , the maximum value of  $n_k$  is first calculated by using Equation (13). If all PDR constraints are met using the vector and the value of  $n_k$ , it stores those values and checks other vectors. Otherwise,  $n_k$  is decremented until all constraint are satisfied. Among all possible  $n_k$  values, the maximum is selected as  $n_k^{max}$ , and the corresponding vector,  $x$ , is returned as a solution. The detailed algorithm is presented in Algorithm 1. Since there are  $m$  QoS classes, the vector of the optimum variable has  $m$  elements. Each  $x_i$  can be one integer value from one to  $l$  (the upper bound of  $x_i$ ). Then, we have  $l^m$  possible solutions. Furthermore, for each solution, up to  $n_k$  times need to be evaluated. As a result, the worst-case computational complexity becomes  $O(Ul^m)$ , where  $U$  represents the upper bound of the node number in the system.

It is also worthwhile to note that even though the greedy algorithm seems to be expensive in terms of computational complexity, it may be affordable in a practical scenario. For example, when there are 3 QoS classes and  $l = 7$ , in most cases, less than 7,000 iterations are needed in our experiments, which is fairly acceptable, considering the computing power of modern computing systems.

In addition, the interior-point algorithm is used to find the solutions. The interior-point algorithm has been developed to solve linear or non-linear convex optimization problems with inequality constraints in a short amount time. The basic idea of this algorithm is to decompose the problem into a sequence of equality constrained problems and apply Newton's method to each problem [47]. There are a lot of variations of the interior-point method, and many of them have been shown to have a polynomial time complexity [48]. In this paper, we use the MATLAB optimization toolbox for the interior-point method with the assumption that each  $x_i$  is a real number. Then, for simulation, we take the ceiling of  $x_i$  after the solution is obtained, since the  $x_i$  value should be an integer number in the real world. Note that, due to the real number relaxation and non-convexity of the objective function, it is possible that the solutions may not be the global optimal or may not even satisfy the required constrains. However, according to the simulation results, in most cases, the observed solutions are close to the global optimal values.

**Algorithm 1** The Custom Greedy Algorithm**Inputs:** $m$ : number of QoS classes $T$ : packet interval $t_d^i (i = \{1 \dots m\})$ : transmission delay in class  $Q_i$  $n_i (i \neq k, i = \{1 \dots m\})$ : number of nodes in each class except class  $Q_k$  $p_i (i = \{1 \dots m\})$ : PDR requirement in class  $Q_i$  $l$ : maximum number of retransmissions $U$ : MAX\_NODE (upper bound of the node number in the system)**Outputs:**

The maximum supportable number of nodes  $n_k^{max}$  in class  $Q_k$  and the corresponding optimal number of retransmission for each class  $x_i^{opt}$

```

1:  $\lambda_{max} = 0$ ;
2:  $x_i^{opt} = 0; \forall i = \{1 \dots m\}$ 
3:  $n_k = 0$ ;
4:  $n_k^{max} = 0$ ;
5: for each  $(x_i, \dots, x_m) \in \{1 \dots l\}$  do
6:    $n_k = \min \left( \left[ -\frac{1}{x_k} \left( \frac{T}{2^{l_k}} \ln(1 - (1 - p_k)^{\frac{1}{x_k}}) + \sum_{i \neq k}^m n_i x_i \right) \right], U - \sum_{i \neq k}^m n_i \right)$ ;
7:   while  $n_k > n_k^{max}$  do
8:      $\lambda_{max} = \sum_{i=1}^m \frac{n_i x_i}{T}$ ;
9:      $P(x_i) = 1 - (1 - e^{-2\lambda_{max} t_d^i})^{x_i}; \forall i = \{1 \dots m\}$ 
10:    if  $P(x_i) \geq p_i \forall i = \{1 \dots m\}$  then
11:       $n_k^{max} = n_k$ ;
12:       $x_i^{opt} = x_i; \forall i = \{1 \dots m\}$ 
13:      break;
14:    end if
15:     $n_k = n_k - 1$ ;
16:  end while
17: end for
18: return :  $n_k^{max}, x_i^{opt} \forall i = \{1 \dots m\}$ 

```

**5. Performance Study**

In this section, we first describe the simulation setup and then analyze the results of the simulations.

**5.1. Simulation Setup**

In order to evaluate the performance of the proposed protocol, we first consider a cluster with three QoS classes. Then, we extend our discussion to the case of four QoS classes. Finally, we consider a case where each QoS class has a different packet size.

When there are three QoS classes, the nodes in a cluster are partitioned into three QoS (in terms of PDR) classes ( $Q_1, Q_2, Q_3$ ), where  $Q_3$  is the selected QoS class in which we want to maximize the supportable number of sensors (*i.e.*,  $k = 3$ ). It is assumed that the required PDR values for  $Q_1$  and  $Q_2$  are  $p_1 = 0.95$  and  $p_2 = 0.8$ , while the PDR requirement for  $Q_3$  ( $p_3$ ) is a variable parameter. Furthermore, we suppose that the numbers of sensor nodes in classes  $Q_1$  and  $Q_2$  are five and 15, respectively (*i.e.*,  $n_1 = 5, n_2 = 15$ ).

In case of four QoS classes, the nodes in the cluster are divided into four QoS classes, and  $Q_4$  is the selected QoS class (*i.e.*,  $k = 4$ ). The required PDR values for  $Q_1, Q_2$  and  $Q_3$  are  $p_1 = 0.95, p_2 = 0.9$  and  $p_3 = 0.8$ , respectively. The PDR requirement for  $Q_4$  ( $p_4$ ) is a variable parameter. The numbers of sensor nodes in classes  $Q_1, Q_2$  and  $Q_3$  are five, 15 and 20, respectively.

In this paper, for practical simulation, we used the DESERTunderwater simulation framework [20], which incorporates spreading loss and various underwater noises, such as turbulence, shipping, wind and thermal noises. The observed solutions to the optimization formulation in Equations (14)–(16) are used as inputs for the simulations. The value of the maximum number of sensor nodes in the selected class is calculated using the solution. Then, this obtained value is also used for simulations as the number of nodes in the selected class.

Each node is equipped with a half-duplex acoustic transceiver that has a data rate of 14 Kbps and a transmission range of 1,100 m. The speed of the underwater acoustic signal is assumed to be 1,500 m/s. The data generation rate applies to every node in the network. The upper bound of the maximum number of retransmissions is set to seven.

## 5.2. Performance Analysis

In this subsection, we first present simulation results for a case with three QoS classes and discuss the results. Then, in order to show that our approach can support an arbitrary number of QoS classes, we extend our discussion to the case where a cluster has four QoS classes. Finally, we present the simulation results and analysis for a case where each of three QoS classes has a different packet size.

### 5.2.1. Analysis of Results for Three QoS Classes

In this case, we consider a cluster that has three QoS classes. We first discuss the effects of the PDR requirement for a QoS class on PDR and on the maximum number of nodes in that QoS class. Then, we continue our discussion for the effects of the network load on the network performance. We assume that the PDR requirement of  $Q_3$  varies and the PDR requirements of  $Q_1$  and  $Q_2$  are given.

*The effects of PDR requirement for class  $Q_3$ :*

In this case, every node transmits a data packet of 160 bytes to the clusterhead in every interval of  $T = 64$  s, which leads to the transmission rate of 20 bps. The PDR requirement for class  $Q_3$  is varied from 0.7 to 0.86.

Tables 1 and 2 show the effects of the PDR requirement for class  $Q_3$  on the PDR and on the maximum supportable number of nodes in class  $Q_3$ , when the greedy and interior-point algorithms are used, respectively. The tables show required PDR values ( $P^{req}$ ), a solution,  $x$ , calculated PDR values using

optimal solutions ( $P^{anal}$ ), PDR values collected from simulations ( $P^{sim}$ ) and the maximum number of nodes in  $Q_3$  ( $n_3^{max}$ ), calculated using the solution, which is also used for simulations.

As shown in Table 1, when the required PDR for class  $Q_3$  varies from 0.7 to 0.86, the maximum supportable number of sensor nodes in this class decreases from 84 to 50. The results indicate that, when the PDR requirement for the selected class decreases, the considered cluster in the network can accommodate a larger number of nodes, while satisfying the required PDR. For instance, if the PDR requirement for the nodes in class  $Q_3$  is 0.86, the considered cluster in the network can support 50 nodes in this class. However, if class  $Q_3$  is required to provide a PDR of 0.7, the considered cluster can support 84 nodes in class  $Q_3$ . It is intuitive that the number of supportable nodes becomes greater as the required PDR decreases. However, one interesting point is that the supportable number of nodes is very sensitive to the PDR requirement. More specifically, when the PDR requirement of  $Q_3$  is lowered from 0.86 to 0.7 (e.g., 18.6% decrease), the supportable number of nodes in  $Q_3$  increases by approximately 68%.

**Table 1.** The effects of the packet delivery ratio (PDR) requirement for class  $Q_3$  on the PDR achieved from the greedy algorithm and the maximum number of nodes in class  $Q_3$  (with  $n_1 = 5$ ,  $n_2 = 15$ ,  $p_1 = 0.95$ ,  $p_2 = 0.80$ ).

$P_1^{req}$	$P_2^{req}$	$P_3^{req}$	Opt. Solut.			$P_1^{anal}$	$P_1^{sim}$	$P_2^{anal}$	$P_2^{sim}$	$P_3^{anal}$	$P_3^{sim}$	$n_3^{max}$
			$x_1$	$x_2$	$x_3$							
0.95	0.80	0.70	5	3	2	0.951	0.965	0.836	0.875	0.701	0.729	84
0.95	0.80	0.72	5	3	2	0.959	0.970	0.852	0.896	0.721	0.750	78
0.95	0.80	0.74	5	3	2	0.965	0.977	0.868	0.908	0.741	0.774	72
0.95	0.80	0.76	5	3	2	0.972	0.982	0.883	0.919	0.761	0.803	66
0.95	0.80	0.78	4	2	2	0.960	0.972	0.800	0.850	0.800	0.819	64
0.95	0.80	0.80	4	2	2	0.960	0.972	0.800	0.850	0.800	0.819	64
0.95	0.80	0.82	4	2	2	0.967	0.977	0.820	0.866	0.820	0.847	58
0.95	0.80	0.84	5	3	3	0.953	0.981	0.840	0.913	0.840	0.904	55
0.95	0.80	0.86	5	3	3	0.962	0.986	0.860	0.927	0.860	0.919	50

Now, we discuss the selection of optimal  $x$  values to maximize the  $n_3$  and meet the requirements using examples in Table 1. As shown in Table 1, when  $p_3^{req}$  is 0.76, the achievable  $p_3^{anal}$  is only 0.761 with  $x_3 = 2$ . This indicates that when  $p_3^{req}$  becomes 0.78,  $p_3^{anal}$  cannot meet the requirement any longer with  $x_3 = 2$ , i.e., 0.761 is less than 0.78. From Equation (11), there are two ways to increase  $p_3^{anal}$  (suppose that  $k = 3$ ). One way is to increase the value of  $x_k$  in the left-hand side (lhs) of Equation (11). As  $x_k$  increases, the lhs increases accordingly. Another way is to reduce  $\lambda_{max}$ . It is clear that the lhs decreases as  $\lambda_{max}$  decreases. It can be seen that, from Equation (6), the value of  $\lambda_{max}$  depends on  $x_i$ , where  $1 \leq i \leq m$ .

Note that our greedy algorithm tests all possible cases. In this particular case, it appears that reducing  $\lambda_{max}$  results in a larger  $n_3$ , i.e.,  $x_1$  and  $x_2$  values are decreased to reduce  $\lambda_{max}$ . This can be also regarded as follows. In order to increase  $p_3^{anal}$ ,  $p_1^{anal}$  and  $p_2^{anal}$  are sacrificed by reducing  $x_1$  and  $x_2$ . This also agrees with the results in Table 1 when  $p_3^{req}$  is 0.78,  $p_1^{anal}$  and  $p_2^{anal}$  values have lower values with less  $x_1$  and  $x_2$  values than when  $p_3^{req}$  is 0.76.

It is also possible in some cases that the greedy algorithm selects a higher  $x_3$  value with which a maximal  $n_3$  value can be obtained, while satisfying the requirements. For example, when  $p_3^{req}$  values are varied from 0.82 to 0.84 in Table 1, the algorithm selects an increased value of  $x_3$  to maximize  $n_3$ . In this case,  $x_1$  and  $x_2$  are also increased to meet  $p_1^{req}$  and  $p_2^{req}$ , respectively. Note that when  $x_3$  increases,  $\lambda_{max}$  also increases. However, in this case, the  $p_3^{req}$  gain from raising  $x_3$  is higher than that lost from increasing  $\lambda_{max}$ . Therefore,  $p_3^{req}$  increases when a greater  $x_3$  value is used.

Another point to note is that when  $p_3^{req}$  varies from 0.8 to 0.82,  $n_3^{max}$  also changes from 64 to 58, even with the same  $x_i$  values. This is because  $n_3^{max}$  depends on  $p_3^{req}$ , as shown in Equation (13). Furthermore, note that  $P_1^{sim}$ ,  $P_2^{sim}$  and  $P_3^{sim}$  values increase, since  $n_3^{max}$  has a lower value with the same  $x_i$  values.

The results in Table 1 also show that, in all cases, both  $P^{anal}$  and  $P^{sim}$  are greater than  $P^{req}$ , i.e., the required PDR is always satisfied for all classes. This indicates that, by using the solution to the optimization formulation, the maximum number of nodes in a specific class can be obtained while satisfying the required PDR for all classes.

Table 2 shows the results based on the solution obtained using the interior-point algorithm. The results are close to those in Table 1, except that  $x_i$  and  $n_3^{max}$  are real numbers. Recall that the ceilings of  $x_i$  and the floor of  $n_3^{max}$  values are used for simulations. When the PDR requirement of class  $Q_3$  varies from 0.7 to 0.86, the maximum number of nodes in class  $Q_3$  decreases from 87 to 53.

**Table 2.** The effects of the PDR requirement for class  $Q_3$  on the PDR achieved from the interior-point algorithm, and the maximum number of nodes in class  $Q_3$  (with  $n_1 = 5$ ,  $n_2 = 15$ ,  $p_1 = 0.95$ ,  $p_2 = 0.80$ ).

$P_1^{req}$	$P_2^{req}$	$P_3^{req}$	Opt. Solut.			$P_1^{anal}$	$P_1^{sim}$	$P_2^{anal}$	$P_2^{sim}$	$P_3^{anal}$	$P_3^{sim}$	$n_3^{max}$
			$x_1$	$x_2$	$x_3$							
0.95	0.80	0.70	4.321	2.321	1.737	0.950	0.962	0.800	0.865	0.700	0.716	87.6
0.95	0.80	0.72	4.321	2.321	1.836	0.950	0.967	0.800	0.884	0.720	0.739	82.8
0.95	0.80	0.74	4.321	2.321	1.943	0.950	0.970	0.800	0.896	0.740	0.751	78.3
0.95	0.80	0.76	4.321	2.321	2.058	0.950	0.956	0.800	0.838	0.760	0.821	73.9
0.95	0.80	0.78	4.321	2.321	2.184	0.950	0.956	0.800	0.860	0.780	0.843	69.6
0.95	0.80	0.80	4.322	2.322	2.322	0.950	0.963	0.800	0.882	0.800	0.862	65.5
0.95	0.80	0.82	4.322	2.322	2.474	0.950	0.970	0.800	0.891	0.820	0.881	61.5
0.95	0.80	0.84	4.322	2.321	2.643	0.950	0.974	0.800	0.908	0.840	0.897	57.5
0.95	0.80	0.86	4.322	2.322	2.836	0.950	0.982	0.800	0.917	0.860	0.911	53.6

Furthermore, note that, as shown in Tables 1 and 2, there are differences between  $P^{anal}$  and  $P^{sim}$  values, and in all cases,  $P^{sim}$  values are greater than  $P^{anal}$  values. In particular, we can observe these phenomena more clearly in Table 1, where there is no distortion, due to the ceiling effect. The reason for these phenomena is that the maximum arrival rate of background traffic,  $\lambda_{max}$ , is used to calculate  $P^{anal}$ , which results in a lower value of  $P^{anal}$ . Therefore, this value can be considered as the lower bound of the PDR that can be achieved, and the results also agree with Lemma 1. In terms of  $n_3^{max}$ , the greedy algorithm and interior-point algorithm show similar results, i.e., interior-point algorithm outputs three more nodes on average. Note that the greedy algorithm shows a higher PDR for the highest

priority group, *i.e.*,  $P_1^{sim}$  with the greedy algorithm shows a 0.0091 higher value than with interior point algorithm. Since the greedy algorithm shows a comparable performance in terms of  $n_3^{max}$  and it shows a higher PDR, which is important for guaranteeing QoS, from now on, we focus on the results from the greedy algorithm.

### Effects of Node Load

In this case, the PDR requirement for class  $Q_3$  is fixed to 0.7. Each underwater sensor periodically generates data packet of 160 bytes. Every node in the cluster transmits data at the rate from 20 bps to 50 bps to the clusterhead, *i.e.*, every node transmits data at every interval from  $T = 25.6$  s to 64 s.

Table 3 shows the PDR requirements for each class ( $P_1^{req}$ ,  $P_2^{req}$ ,  $P_3^{req}$ ) and the various node loads in the network. It also shows the solution ( $x_1$ ,  $x_2$ ,  $x_3$ ) and the maximum supportable number of nodes in class  $Q_3$  ( $n_3^{max}$ ) determined from the optimization formulation. The greedy algorithm is used to find the solutions in Table 3.

**Table 3.** The solutions from the greedy algorithm with various node loads ( $n_1 = 5$ ,  $n_2 = 15$ ).

Node load (bps)	$P_1^{req}$	$P_2^{req}$	$P_3^{req}$	Opt. Solut.			$n_3^{max}$
				$x_1$	$x_2$	$x_3$	
20	0.95	0.80	0.70	5	3	2	84
25	0.95	0.80	0.70	5	3	2	60
30	0.95	0.80	0.70	5	3	2	44
35	0.95	0.80	0.70	5	3	2	33
40	0.95	0.80	0.70	5	3	2	24
45	0.95	0.80	0.70	5	3	2	18
50	0.95	0.80	0.70	5	3	2	12

As shown in Table 3, all solutions have the same vector,  $x$ . Note that the node load is controlled by varying the packet transmission interval,  $T$ . Then, from Equation (13), it can be seen that  $n_k$  is inversely proportional to the node load. In other words,  $n_k$  and  $T$  are linearly dependent with given  $p_i$  and  $x_i$  values, which indicates that  $n_k$  depends more on the change of the  $T$  value than on the change of  $x_i$  values.

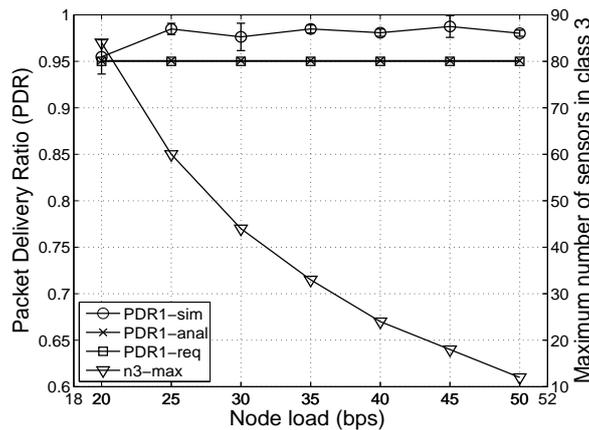
In order to facilitate understanding, we show and compare  $P^{req}$ ,  $P^{anal}$  and  $P^{sim}$  over different node loads in Figure 3. Furthermore, to show the confidence level of simulations, we present the standard deviation of  $P^{sim}$  values along with mean values in Figure 3.

From Figure 3, we can see that, when the node load is high, a smaller number of nodes in class  $Q_3$  can be supported. On the contrary, when the node load becomes lower, the considered cluster in the network can accommodate a larger number of nodes in class  $Q_3$ , while meeting the PDR requirements.

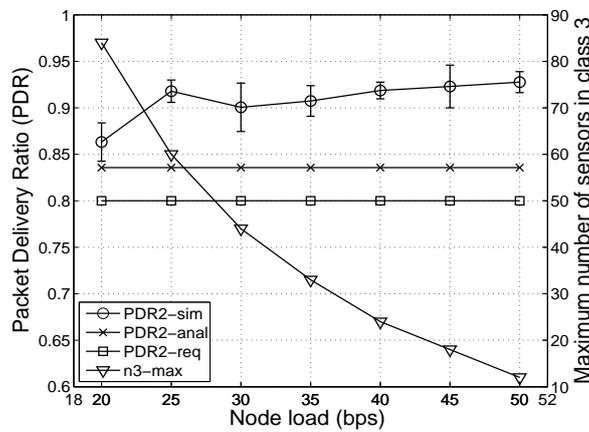
More specifically, Figure 3 shows that the nodes in all classes can satisfy their PDR requirements over various node loads. However, when the node load increases from 20 bps to 50 bps, the maximum supportable number of nodes in class  $Q_3$  shows a sharp decline from 84 to 12. In particular, up to the 30 bps node load, the supportable number of nodes in  $Q_3$  decreases sharply. Figure 3 also indicates that, by using the optimal value of the maximum number of retransmissions, the average PDR value of nodes

in all classes are above their PDR requirements as the node load increases. For example, in Figure 3a, which shows obtained PDR values for class  $Q_1$ , the average PDR values from both the analytical model and simulation are always equal to or greater than the required PDR value, 0.95. This also applies to class  $Q_2$  and class  $Q_3$  in Figure 3b,c, respectively.

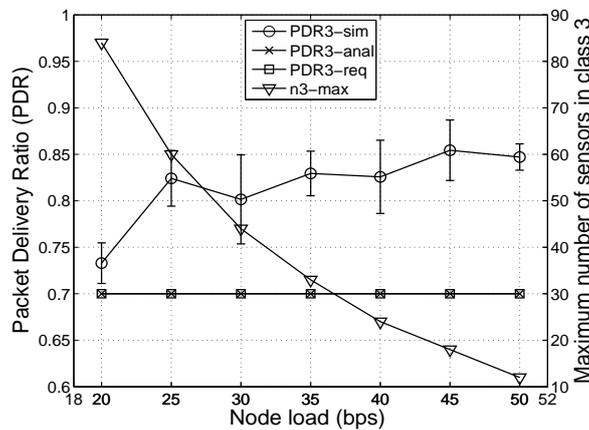
**Figure 3.** Effects of node load on the PDR (mean +/- standard deviation) achieved from the greedy algorithm and from simulations, and the maximum number of nodes in class  $Q_3$  with  $n_1 = 5, n_2 = 15$ . (a) For class  $Q_1$ ; (b) For class  $Q_2$ ; (c) For class  $Q_3$ .



(a)



(b)



(c)

### 5.2.2. Analysis of Results for Four QoS Classes

In this section, we show that the proposed scheme can support four QoS classes. The greedy algorithm is used to find the solution in this experiment.

#### Effects of PDR Requirement for Class $Q_4$

Similarly to the case of three QoS classes, every node transmits a data packet of 160 bytes to the clusterhead in every interval of  $T = 64$  s, which leads to the transmission rate of 20 bps. The PDR requirement for class  $Q_4$  is varied from 0.7 to 0.86.

Table 4 shows the effects of the PDR requirement for class  $Q_4$  on the PDR values of the nodes in each QoS class, which are obtained from the greedy algorithm and simulations. It also shows the obtained maximum supportable number of nodes in class  $Q_4$  using the optimal  $x$  values.

As shown in Table 4, when the required PDR for class  $Q_4$  varies from 0.7 to 0.86, the maximum supportable number of sensor nodes,  $n_4^{max}$ , in this class decreases from 46 to 25 or, equivalently, the maximum supportable number of sensor nodes in the considered cluster decreases from 86 to 65. More specifically, if the required PDR for class  $Q_4$  is 0.7, the considered cluster can support 46 nodes in this class. On the other hand, if the PDR requirement for the nodes in class  $Q_4$  is 0.86, the cluster can support only 25 nodes in class  $Q_4$ .

Note that, in some cases,  $n_4^{max}$  remains the same even when  $P_4^{req}$  increases. For example, when  $P_4^{req}$  varies from 0.74 to 0.80,  $n_4^{max}$  keeps the value of 36. This is because the effect of the  $P_3^{req}$ 's change is not enough for changing the  $n_k$  value in Equation (13). Furthermore,  $P_3^{anal} = 0.83$  is sufficient for  $P_3^{req}$  values from 0.74 to 0.80, which results in the same  $x_i$  values and  $n_k$ . The results also indicate that, among all feasible solutions, the solution  $x = \{4, 3, 2, 2\}$  can achieve the maximum number of nodes in  $Q_4$  in the given  $P_4^{req}$  range.

**Table 4.** The effects of the PDR requirement for class  $Q_4$  on the PDR achieved from the greedy algorithm and the maximum number of nodes in class  $Q_4$  (with  $n_1 = 5$ ,  $n_2 = 15$ ,  $n_3 = 20$ ,  $P_1^{req} = 0.95$ ,  $P_2^{req} = 0.90$ ,  $P_3^{req} = 0.80$ ).

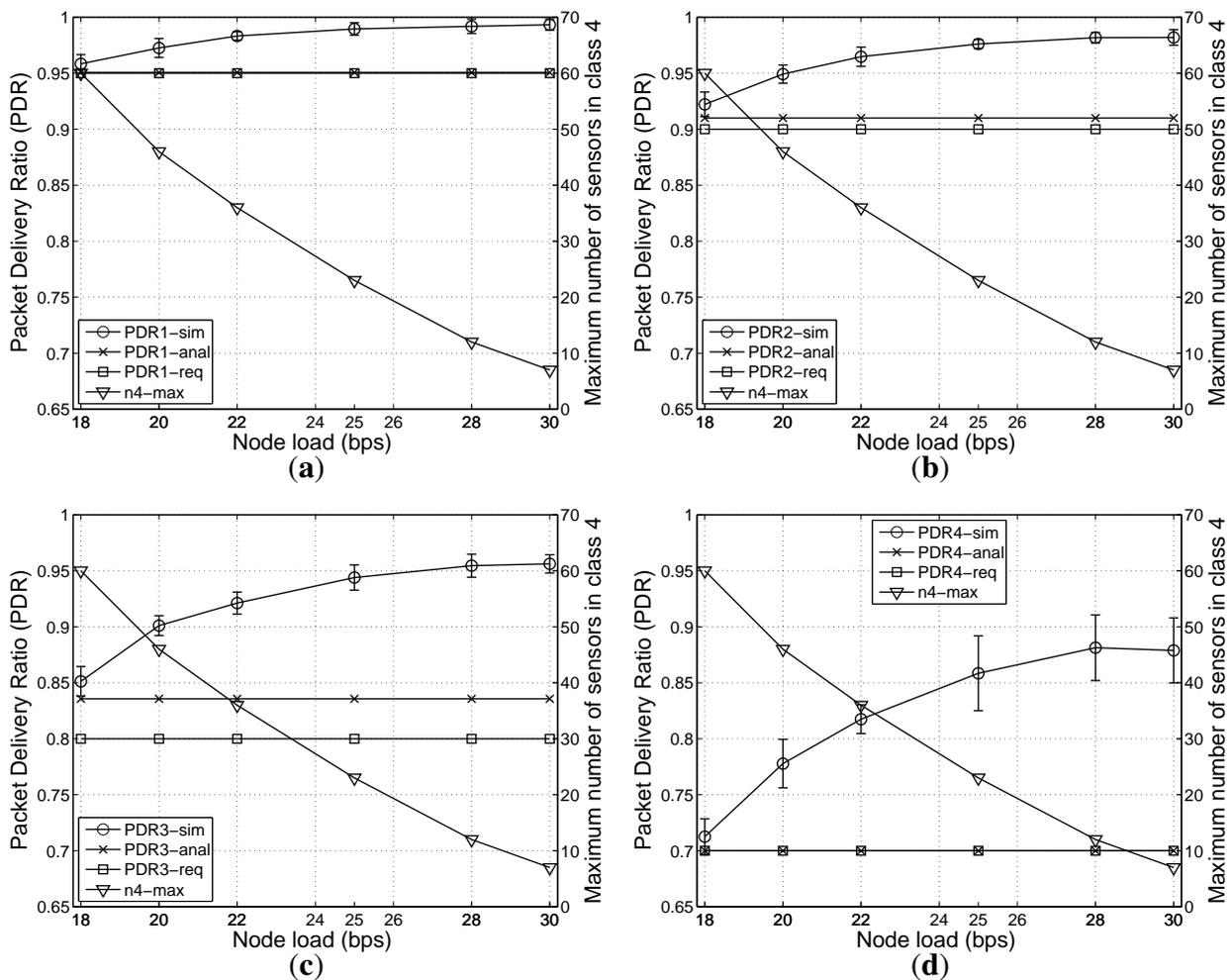
$P_4^{req}$	Opt. Solut.				$P_1^{anal}$	$P_1^{sim}$	$P_2^{anal}$	$P_2^{sim}$	$P_3^{anal}$	$P_3^{sim}$	$P_4^{anal}$	$P_4^{sim}$	$n_4^{max}$
	$x_1$	$x_2$	$x_3$	$x_4$									
0.70	5	4	3	2	0.95	0.97	0.91	0.94	0.83	0.90	0.70	0.77	46
0.72	5	4	3	2	0.96	0.98	0.92	0.96	0.85	0.91	0.72	0.80	40
0.74	4	3	2	2	0.96	0.97	0.91	0.93	0.80	0.84	0.80	0.83	36
0.76	4	3	2	2	0.96	0.97	0.91	0.93	0.80	0.84	0.80	0.83	36
0.78	4	3	2	2	0.96	0.97	0.91	0.93	0.80	0.84	0.80	0.83	36
0.80	4	3	2	2	0.96	0.97	0.91	0.93	0.80	0.84	0.80	0.83	36
0.82	6	4	3	3	0.97	0.99	0.90	0.95	0.82	0.91	0.82	0.91	32
0.84	5	4	3	3	0.95	0.98	0.91	0.96	0.84	0.91	0.84	0.91	30
0.86	5	4	3	3	0.96	0.98	0.92	0.96	0.86	0.93	0.86	0.92	25

Effects of Node Load

In this case, the PDR requirement for class  $Q_4$  is fixed to 0.7. Every node in the cluster transmits data at the rate from 18 bps to 30 bps to the clusterhead with the data packet size of 160 bytes, *i.e.*, every node transmits data at every interval from  $T = 42.67 s$  to  $71.11 s$ . Figure 4 compares  $P^{req}$ ,  $P^{anal}$  and  $P^{sim}$  over different node loads for all QoS classes.

As shown in Figure 4, the case of four classes shows a similar pattern to the three-class case. When the node load is small, a higher number of nodes in class  $Q_4$  can be supported. When the node load becomes higher, the considered cluster in the network can accommodate a smaller number of sensor nodes in class  $Q_4$ , while satisfying the PDR requirements. Furthermore, the nodes in all classes can meet their PDR requirements over various node loads.

**Figure 4.** The effects of node load on the PDR (mean +/- standard deviation) achieved from the greedy algorithm and from simulations and the maximum number of nodes in class  $Q_4$  with  $n_1 = 5$ ,  $n_2 = 15$ ,  $n_3 = 20$ . (a) For class  $Q_1$ ; (b) For class  $Q_2$ ; (c) For class  $Q_3$ ; (d) For class  $Q_4$ .



### 5.2.3. Analysis of Results for Three QoS Classes with Different Packet Sizes

In this subsection, we consider a case where each QoS class has a different packet size. The sensor nodes in class  $Q_1$ ,  $Q_2$  and  $Q_3$  periodically generate data packets of 300 bytes, 200 bytes and 150 bytes, respectively. The PDR requirement for class  $Q_3$  is varied from 0.7 to 0.9. We discuss the effects of the PDR requirement for class  $Q_3$  on the maximum number of sensor nodes in class  $Q_3$ .

From Table 5, we can see that the nodes in all classes satisfy their PDR requirements, *i.e.*, the  $P^{sim}$  and the  $P^{anal}$  values are greater than  $P^{req}$  values. As the required PDR for class  $Q_3$  increases from 0.7 to 0.9, the maximum supportable number of sensor nodes in this class decreases from 51 to 27 nodes. The results also implicate that, in order to satisfy the required PDR for all classes, the maximum number of nodes in selected class decrease as the required PDR for this class increases.

**Table 5.** The effects of the PDR requirement for class  $Q_3$  on the PDR achieved from the greedy algorithm and the maximum number of nodes in class  $Q_3$  (with  $n_1 = 5$ ,  $n_2 = 15$ ,  $p_1 = 0.95$ ,  $p_2 = 0.80$  and a different packet size for each QoS class).

$P_1^{req}$	$P_2^{req}$	$P_3^{req}$	Opt. Solut.			$P_1^{anal}$	$P_1^{sim}$	$P_2^{anal}$	$P_2^{sim}$	$P_3^{anal}$	$P_3^{sim}$	$n_3^{max}$
			$x_1$	$x_2$	$x_3$							
0.95	0.80	0.70	5	2	1	0.979	0.989	0.879	0.897	0.716	0.719	51
0.95	0.80	0.72	4	2	1	0.970	0.987	0.903	0.924	0.748	0.750	42
0.95	0.80	0.74	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.76	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.78	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.80	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.82	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.84	6	2	2	0.970	0.994	0.813	0.878	0.873	0.893	40
0.95	0.80	0.86	6	2	2	0.971	0.993	0.817	0.886	0.876	0.890	39
0.95	0.80	0.88	6	2	2	0.981	0.996	0.845	0.909	0.895	0.921	32
0.95	0.80	0.90	5	2	2	0.977	0.995	0.873	0.929	0.915	0.938	27

Another point to note in Table 5 is that in many cases,  $n_3^{max}$  keeps the same value of 40. When  $P_3^{req}$  is 0.74, the algorithm selects  $x = \{6, 2, 2\}$  with which the obtained values of  $n_3^{max} = 40$  and  $P_3^{anal} = 0.873$ . When  $P_3^{req}$  becomes 0.76,  $n_k$  is calculated again using Equation (13). In this case, it appears that the effect of the  $P_3^{req}$ 's change is not significant to change the new  $n_k$  value. Moreover,  $P_3^{anal} = 0.873$  is sufficient for new  $P_3^{req} = 0.76$ . Therefore, the  $n_k$  remains at the same value. The phenomenon continues until  $P_3^{req}$  becomes 0.86, which has a sufficient impact on changing the  $n_k$  value in Equation (13). When  $P_3^{req}$  varies from 0.88 to 0.9,  $P_3^{anal} = 0.895$  does not meet the new  $P_3^{req} = 0.9$ . Therefore, as shown in Equation (11),  $x_3$  should be increased or  $\lambda_{max}$  should be reduced. In this case, the algorithm chooses to reduce  $\lambda_{max}$  by decreasing the  $x_1$  value from six to five, since it leads to a larger value of  $n_3^{max}$ .

## 6. Concluding Remarks

In this paper, we have proposed a practical and low-complexity MAC scheme that does not require time synchronization or scheduling overhead, for QoS-aware and cluster-based underwater acoustic

sensor networks (UASN). In particular, we have considered an optimization problem to maximize the supportable number of sensor nodes in UASNs that are required to provide differentiated QoS in terms of PDR. In order to address the problem, the packet delivery probability (PDP) has been estimated, and based on the estimation, an optimization formulation has been designed to determine optimal values of the maximum number of packet retransmissions for each QoS class. The greedy and interior-point algorithms are used to find the solutions, which are verified by simulations. The simulation results have shown that, by solving the proposed optimization formulation, the supportable number of underwater sensor nodes can be maximized, while satisfying the QoS requirements for each class.

### Acknowledgments

This work was supported by the 2012 Research Fund of University of Ulsan.

### Author Contributions

Seokhoon Yoon and Thi-Tham Nguyen developed the network architecture and algorithms. The experimental simulations of this work were conducted by Thi-Tham Nguyen. Seokhoon Yoon, Thi-Tham Nguyen and Duc Van Le performed analysis and wrote the manuscript. All authors have read and approved the final manuscript.

### Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

### References

1. Lloret, J. Underwater sensor nodes and networks. *Sensors* **2013**, *13*, 11782–11796.
2. Akyildiz, I.F.; Pompili, D.; Melodia, T. Underwater acoustic sensor networks: Research challenges. *J. Ad Hoc Netw.* **2005**, *3*, 257–279.
3. Heidemann, J.; Ye, W.; Wills, J.; Syed, A.; Li, Y. Research Challenges and Applications for Underwater Sensor Networking. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'06), Las Vegas, NV, USA, 3–6 April 2006; pp. 228–235.
4. Heidemann, J.; Stojanovic, M.; Zorzi, M. Underwater sensor networks: Applications, advances, and challenges. *Philos. Trans. R. Soc. A* **2012**, *370*, 158–175.
5. Akyildiz, I.F.; Pompili, D.; Melodia, T. State of the art in protocol research for underwater acoustic sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.* **2007**, *11*, 11–22.
6. Sozer, E.M.; Stojanovic, M.; Proakis, J.G. Underwater acoustic networks. *IEEE J. Ocean. Eng.* **2000**, *25*, 72–83.
7. Stojanovic, M. On the Relationship between Capacity and Distance in an Underwater Acoustic Communication Channel. In Proceedings of the 1st ACM International Workshop on Underwater Networks (WUWNet 06), Los Angeles, CA, USA, 25 September 2006.
8. Stojanovic, M.; Preisig, J. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Commun. Mag.* **2009**, *47*, 84–89.

9. Stojanovic, M. *Acoustic Underwater Communications*; Proakis, J.G., Ed.; Wiley: New York, NY, USA, 2003; Chapter: *Encyclopedia of Telecommunications*.
10. Chitre, M.; Shahabodeen, S.; Stojanovic, M. Underwater acoustic communications and networking: Recent advances and future challenges. *Mar. Technol. Soc. J.* **2008**, *42*, 103–116.
11. Domingo, M.C.; Prior, R. A Distributed Clustering Scheme for Underwater Wireless Sensor Networks. In Proceedings of the 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Athens, Greece, 3–7 September 2007; pp. 1–5.
12. Sharad, S.; Shailendra, M.; Mayank, S. Clustering based on node density in heterogeneous underwater sensor network. *Int. J. Inf. Technol. Comput. Sci. (IJITCS)* **2013**, *5*, 49–55.
13. Salva-Garau, F.; Stojanovic, M. Multi-Cluster Protocol for Ad Hoc Mobile Underwater Acoustic Networks. In Proceedings of OCEANS 2003, San Diego, CA, USA, 22–26 September 2003; pp. 91–98.
14. Yigitel, M.A.; Incel, O.D.; Ersoy, C. QoS-aware MAC protocols for wireless sensor networks: A survey. *Comput. Netw.* **2011**, *55*, 1982–2004.
15. Chirdchoo, N.; Soh, W.S.; Chua, K.C. Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks. In Proceedings of the 26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6–12 May 2007; pp. 2271–2275.
16. Ahn, J.; Syed, A.; Krishnamachari, B.; Heidemann, J. Design and analysis of a propagation delay tolerant ALOHA protocol for underwater networks. *Ad Hoc Netw.* **2011**, *9*, 752–766.
17. Jin, L.; Huang, D. A slotted CSMA based reinforcement learning approach for extending the lifetime of underwater acoustic wireless sensor networks. *Comput. Commun.* **2013**, *36*, 1094–1099.
18. Petrioli, C.; Petroccia, R.; Stojanovic, M. A Comparative Performance Evaluation of MAC Protocols for Underwater Sensor Networks. In Proceedings of OCEANS 2008, Quebec, QC, Canada, 15–18 September 2008; pp. 1–10.
19. Peleato, B.; Stojanovic, M. Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks. *IEEE Commun. Lett.* **2007**, *11*, 1025–1027.
20. Masiero, R.; Azad, S.; Favaro, F.; Petrani, M.; Toso, G.; Guerra, F.; Casari, P.; Zorzi, M. DESERT Underwater: An NS-Miracle-Based Framework to Design, Simulate, Emulate and Realize Test-Beds for Underwater Network Protocols. In Proceedings of OCEANS 2012 MTS/IEEE Yeosu, Yeosu, Korea, 21–24 May 2012; pp. 1–10.
21. Molins, M.; Stojanovic, M. Slotted FAMA: A MAC Protocol for Underwater Acoustic Networks. In Proceedings of OCEANS 2006—Asia Pacific, Singapore, Singapore, 16–19 May 2007; pp. 1–7.
22. Syed, A.A.; Ye, W.; Heidemann, J. T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks. In Proceedings of the 27th Conference on Computer Communications (INFOCOM 2008), Phoenix, AZ, USA, 13–18 April 2008; pp. 789–797.
23. Vieira, L.F.M.; Kong, J.; Lee, U.; Gerla, M. Analysis of ALOHA Protocols for Underwater Acoustic Sensor Networks. In *Work in Progress poster at the First ACM International Workshop on UnderWater Networks (WUWNet)*; ACM: Los Angeles, CA, USA, 2006.

24. Smith, S.M.; Park, J.C.; Neel, A. A Peer-to-Peer Communication Protocol for Underwater Acoustic Communication. In Proceedings of OCEANS '97, MTS/IEEE, Halifax, Canada, 6–9 October 1997; pp. 268–272.
25. Fang, D.; Li, Y.; Huang, H.; Yin, L. A CSMA/CA-Based MAC Protocol for Underwater Acoustic Networks. In Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010; pp. 1–4.
26. Xie, P.; Cui, J.H. Exploring Random Access and Handshaking Techniques in Large-Scale Underwater Wireless Acoustic Sensor Networks. In Proceedings of OCEAN 2006, Boston, MA, USA, 18–21 September 2006; pp. 1–6.
27. Guo, X.; Frater, M.R.; Ryan, M.J. An Adaptive Propagation-delay-tolerant MAC Protocol for Underwater Acoustic Sensor Networks. In Proceedings of OCEANS 2007—Europe, Aberdeen, Scotland, 18–21 June 2007; pp. 1–5.
28. Guo, X.; Frater, M.R.; Ryan, M.J. A Propagation-Delay-Tolerant Collision Avoidance Protocol for Underwater Acoustic Sensor Networks. In Proceedings of OCEANS 2006—Asia Pacific, Singapore, Singapore, 16–19 May 2006; pp. 1–6.
29. Peng, Z.; Zhu, Y.; Zhou, Z.; Guo, Z.; Cui, J.H. COPE-MAC: A Contention-Based Medium Access Control Protocol with Parallel Reservation for Underwater Acoustic Networks. In Proceedings of OCEANS 2010, Sydney, Australia, 24–27 May 2010; pp. 1–10.
30. Proakis, J.G.; Sozer, E.M.; Rice, J.A.; Stojanovic, M. *Shallow Water Acoustic Networks*; John Wiley and Sons: Boston, MA, USA, 2003; Chapter *Encyclopedia of Telecommunications*.
31. Fullmer, C.L.; Garcia Luna Aceves, J.J. Floor acquisition multiple access (FAMA) for packet-radio networks. *SIGCOMM Comput. Commun. Rev.* **1995**, *25*, 262–273.
32. Yoon, S.; Qiao, C.; Sudhaakar, R.; Li, J.; Talty, T. QoMOR: A QoS-Aware MAC Protocol Using Optimal Retransmission for Wireless Intra-Vehicular Sensor Networks. In Proceedings of Mobile Networking for Vehicular Environments 2007, Anchorage, AK, USA, 11 May 2007; pp. 121–126.
33. Liu, Z.; Elhanany, I. RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks. In Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, USA, 23–25 April 2006; pp. 768–773.
34. Tan, J.; Chan, M.C.; Tan, H.X.; Kong, P.Y.; Tham, C.K. A Medium Access Control Protocol for UWB Sensor Networks with QoS Support. In Proceedings of IEEE Conference on Local Computer Networks, Montreal, Canada, 14–17 October 2008; pp. 289–296.
35. Kim, H.; Min, S.G. Priority-Based QoS MAC Protocol for Wireless Sensor Networks. In Proceedings of IEEE International Symposium on Parallel Distributed Processing, Rome, Italy, 23–29 May 2009; pp. 1–8.
36. Baroudi, U. EQoSA: Energy and QoS Aware MAC for Wireless Sensor Networks. In Proceedings of the 9th International Symposium on Signal Processing and Its Applications, Sharjah, United Arab Emirates, 12–15 February 2007; pp. 1–4.

37. Slama, I.; Shrestha, B.; Jouaber, B.; Zeghlache, D. A Hybrid MAC with Prioritization for Wireless Sensor Networks. In Proceedings of IEEE Conference on Local Computer Networks, Montreal, Canada, 14–17 October 2008; pp. 274–281.
38. Yahya, B.; Ben-Othman, J. Energy efficient and QoS aware medium access control for wireless sensor networks. *Concurr. Comput. Pract. Exper.* **2010**, *22*, 1252–1266.
39. Sudhaakar, R.S.; Yoon, S.; Zhao, J.; Qiao, C. A novel Qos-aware MAC scheme using optimal retransmission for wireless networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 2230–2235.
40. Jagannath, J.; Saji, A.; Kulhandjian, H.; Sun, Y.; Demirors, E.; Melodia, T. A Hybrid MAC Protocol with Channel-dependent Optimized Scheduling for Clustered Underwater Acoustic Sensor Networks. In Proceedings of ACM International Conference on Underwater Networks and Systems (WUWNet), Kaohsiung, Taiwan, 11–13 November 2013.
41. Skalli, H.; Ghosh, S.; Das, S.K.; Lenzini, L.; Conti, M. Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions. *IEEE Commun. Mag.* **2007**, *45*, 86–95.
42. Saurav, G.; Nive, D.; Tanmoy, S. *A Cluster Based Multi-Radio Multi-Channel Assignment Approach in Wireless Mesh Networks*; Springer Berlin Heidelberg: Berlin, Germany, 2011; Chapter: *Computer Networks and Information Technologies—Series: Communications in Computer and Information Science*, pp. 487–491.
43. Naveed, A.; Kanhere, S.S. Cluster-Based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks. In Proceedings of the 34th IEEE Conference on Local Computer Networks, Zurich, Switzerland, 20–23 October 2009; pp. 53–60.
44. Raniwala, A.; Gopalan, K.; Chiueh, T. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.* **2004**, *8*, 50–65.
45. Ramachandran, K.N.; Belding, E.; Almeroth, K.C.; Buddhikot, M.M. Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In Proceedings of the 25th INFOCOM 2006, Barcelona, Spain, 23–29 April 2006; pp. 1–12.
46. Papoulis, A.; Pillai, S.U. *Probability, Random Variables and Stochastic Processes*; McGraw-Hill: New York, NY, USA, 2002; Chapter 10, pp. 456.
47. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, United Kingdom, 2004; Chapter 11, pp. 561–623.
48. Nemirovski, A.S.; Todd, M.J. Underwater sensor nodes and networks. *Acta Numer.* **2008**, *17*, 191–234.