ARS: An Adaptive Retransmission Scheme for Contention-Based MAC Protocols in Underwater Acoustic Sensor Networks

Thi-Tham Nguyen and Seokhoon Yoon

Department of Electrical and Computer Engineering, University of Ulsan, Ulsan 680-749, Republic of Korea

Correspondence should be addressed to Seokhoon Yoon; seokhoonyoon@ulsan.ac.kr

Received 11 August 2014; Accepted 13 January 2015

Academic Editor: Nianbo Liu

Copyright © 2015 T.-T. Nguyen and S. Yoon. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the limited capacity and high propagation delay of underwater communication channels, contention-based media access control (MAC) protocols suffer from a low packet delivery ratio (PDR) and a high end-to-end (E2E) delay in underwater acoustic sensor networks due to the reliance on packet retransmission for reliable data delivery. In order to address the problem of low performance, we propose a novel adaptive retransmission scheme, named ARS, which dynamically selects an optimal value of the maximum number of retransmissions, such that the successful delivery probability of a packet is maximized for a given network load. ARS can be used for various contention-based protocols and hybrid MAC protocols that have contention periods. In this paper, ARS is applied to well-known contention-based protocols, Aloha and CSMA. Simulation results show that ARS can achieve significant performance improvement in terms of PDR and E2E delay over original MAC protocols.

1. Introduction

Underwater acoustic sensor networks (UASNs) have received growing interest due to their potential application to oceanographic data collection, environment monitoring, undersea exploration, disaster prevention, assisted navigation, and tactical surveillance [1, 2].

Unfortunately, establishing an effective UASN brings about new challenges due to unique characteristics of the underwater acoustic communication channel. First, the underwater acoustic communication channel has a high propagation delay due to the low speed of acoustic signals, which is approximately 1500 m/s, five orders of magnitude slower than radio waves. Second, the available bandwidth for an acoustic channel is limited, which leads to a low data rate, typically only tens of kilobits per second [1, 3, 4]. Third, the high bit error rate is another challenge on an underwater acoustic communication channel [1].

Media access control (MAC) protocols for UASNs have been extensively studied to mitigate the limitations of underwater communication channels. Among a lot of MAC protocols that have been studied for UASNs, contention-based MAC protocols, most of which are based on Aloha [5–8] and CSMA [9–14], have particularly received a great deal of attention due to their low complexity and high applicability in UASNs [5–17]. It has also been shown that a simple contention-based MAC protocol can achieve acceptable throughput and low latency with a low network load without requiring time synchronization [14, 16].

Contention-based MAC protocols for a UASN can be further classified into handshake-based and random access-based protocols. There have been a lot of studies on handshake-based protocols [10, 15, 16, 18, 19] that attempted to address the long propagation delay in UASNs. However, the exchange of control packets causes a long packet delay, and control packets also have a long preamble, which leads to degraded network performance [20]. As a result, those protocols are not appropriate for applications that require a low delay.

There have also been a considerable number of studies on random access-based MAC protocols in UASNs [5, 6, 8, 9, 12]. A drawback to random access-based MAC protocols comes from their reliance on packet retransmission. More specifically, they depend on retransmission for reliable data delivery, which is suitable for terrestrial wireless networks. However, in a UASN, packet retransmission can quickly
saturate the network due to the limited channel capacity, which results in a high level of packet collisions and the consequent low PDR.

Moreover, due to the high propagation delay of the underwater acoustic communication channel, the MAC protocol requires a long slot duration, which leads to a long back-off interval and end-to-end delay. In other words, the unique characteristics of the underwater acoustic communication channel make existing packet retransmission strategies proposed for terrestrial wireless networks unsuitable for UASNs.

Therefore, in a communication environment with a limited channel capacity, the decision on retransmission should be carefully made so as not to impose a high network load that can inadvertently result in very low performance in terms of PDR and E2E delay.

In order to address this issue, we propose an adaptive retransmission based MAC scheme, named ARS, which selects an optimal value of maximum number of retransmissions that is adapted to the network load such that successful packet delivery probability (PDP) is maximized.

ARS periodically calculates a PDP value using the current maximum number of retransmissions (or maximum retransmissions) and then compares it with the estimated PDP values that are calculated by increasing and decreasing the maximum number of retransmissions. Then, ARS chooses a new value for the maximum retransmissions with which a higher PDP value can be achieved. Simulation results show that ARS can achieve higher performance in terms of PDR and E2E delay compared to the existing schemes. In particular, when the network load changes, ARS also shows higher performance than the existing algorithms. Note that sensors in a sensor network may increase the sensor data transmission rate when specific events occur or some conditions are met.

It is also worthwhile to note that ARS can be applied not only to pure contention-based MAC protocols (including Aloha, Aloha-CS, and CSMA) but also to hybrid MAC protocols that employ contention periods (i.e., by using ARS the performance of data transmission in contention periods can be improved).

The rest of this paper is organized as follows. Sections 2 and 3 present the related work and the system model, respectively. Then, we elaborate on the proposed ARS scheme in detail in Section 4. The simulation results are presented and analyzed in Section 5. Finally, Section 6 concludes the paper and suggests future work.

2. Related Work

MAC protocols for a UASN can be divided into contention-free and contention-based protocols. The contention-free protocols consist of frequency division multiple access (FDMA), time division multiple access (TDMA), and code division multiple access (CDMA), in which they assign different frequency bands, time slots, or spreading codes to different users to avoid collisions among transmissions. In the contention-based protocols, on the other hand, the nodes need to compete to access the shared channel.

It is already known that FDMA is not suitable for UASNs due to the limited available bandwidth of underwater acoustic channels. TDMA requires a large guard time and strict synchronization, which limits its efficiency [4]. Also, it is known that CDMA-based protocols require a high-complexity design for UASNs. In particular, it is necessary to design access codes with high autocorrelation and low cross-correlation properties to achieve minimum interference among users in CDMA-based protocols [1].

In contrast, contention-based MAC protocols, most of which are based on Aloha [5–8] and CSMA [9–13], have recently received significant attention for UASNs due to their simplicity and acceptable throughput [5–7, 9–11, 14–17]. Contention-based MAC protocols for UASNs can be further classified into handshake-based protocols and random access-based protocols.

A lot of handshake-based protocols have been studied [10, 15, 16, 18, 19]. For example, Guo et al. [15] introduced the propagation delay tolerant collision avoidance protocol (PCAP). In PCAP, in order to take advantage of a long propagation delay, while the sender is waiting for the clear to send (CTS) packet, it is allowed to transmit another data packet or perform a handshake for the next queued data packet. PCAP requires clock synchronization between neighboring nodes. Another handshake-based protocol, called distance-aware collision avoidance protocol (DACAP), was proposed by Peleato and Stojanovic [16]. Under DACAP, after receiving CTS, the sender waits for a specific time before transmitting the data packet in order to ensure the sender can receive any warning from the intended receiver to avoid the collisions. The length of the waiting period depends on the distance between sender and receiver.

Note that those handshake-based protocols can cause a long packet delay due to the exchange of control packets prior to actual data transmission. Moreover, those control packets also have a long preamble in a practical underwater communication environment, which results in low network performance [20].

Another approach to channel contention resolution is to use tone signals. For example, Syed et al. [17] proposed a tone-based protocol called T-Lohi. In T-Lohi, prior to data transmission, a node transmits a short tone to inform its neighbors about the transmission and receives tone signals from other nodes (which may arrive at different time instances due to different propagation delays) to detect the number of channel contenders. If the node does not receive any tones, it starts data transmission. Otherwise, it performs a backoff with a back-off interval calculated using the number of tones received. However, T-Lohi nodes need special hardware for a wake-up tone receiver to detect tones using low energy consumption.

There have also been a lot of studies on random access-based MAC protocols in UASNs [5, 6, 8, 9, 12]. In particular, Chirdchoo et al. [5] proposed two enhancements to Aloha: Aloha with collision avoidance (Aloha-CA) and Aloha with advance notification (Aloha-AN). These two schemes utilize information obtained from overheard packets plus information about propagation delays between every node pair in the network to calculate other nodes’ busy durations, which are
maintained in the local database table of each node. When a node has a packet to transmit, in Aloha-CA, the node checks the busy durations of other nodes in its database table to determine whether its transmission would cause a collision. In the event of a possible collision, the node defers transmission for a random time. In Aloha-AN, a sender also performs a collision check using its database table. If no collision is foreseen, it transmits a small notification packet to inform other nodes about its pending data transmission.

Another extension of the Aloha protocol is Aloha-CS [14, 16, 21]. According to Petrioli et al. [14] and Peleato and Stojanovic [16], an Aloha-based protocol can be a potential protocol for UASNs because it offers high throughput and low latency and does not require time synchronization or a handshake mechanism. Ahn et al. [6] proposed another Aloha-based protocol, called propagation delay tolerant Aloha (PDT-Aloha), where the authors try to handle the space-time uncertainty in underwater acoustic channels. Nodes transmit only at the start of globally synchronized slots. The spatial uncertainty is handled by adding a guard time, which is proportional to the propagation delay.

A major disadvantage to these random access-based MAC protocols is that they need to rely on a retransmission mechanism for reliable data delivery. Since packet retransmissions can increase network traffic significantly, the decision on packet retransmission should be carefully made so as not to degrade network performance. In order to address this issue, the goal of our work is to design a MAC scheme that can determine an optimal value of the maximum number of retransmissions based on network load so that the packet delivery ratio is maximized with a low end-to-end delay and without requiring time synchronization and special hardware.

Note that some protocols take a hybrid approach that uses features of both TDMA or CDMA and random-access protocols [22, 23]. In particular, Hsu and Hong [22] proposed a hybrid of scheduling and a random-access protocol for UASNs. They divided the channel into several superframes, which contain broadcast, gathering, and event report periods. During the broadcast and gathering periods, each sensor broadcasts and gathers data in a predetermined time slot, where it can transmit data while avoiding collisions. On the other hand, during the event report period, sensor nodes use a random-access protocol to report the sensed events that can not be transmitted using prescheduled time slots.

One benefit of a hybrid protocol is that it can provide differentiated services and quality of service (QoS). For example, the superframe in a hybrid protocol can consist of a contention-free period (CFP) and contention period (CP). In the CFP, time slots are assigned to sensor nodes so that the high-priority data (or data that require a low delay) can be transmitted without collisions. In contrast, for low-priority data or non-real-time data, sensor nodes contend for channel access using a random-access protocol (e.g., CSMA and Aloha) during the CP. Note that ARS can be applied to those hybrid protocols to increase network performance during CPs.

It is worthwhile to note that our work is significantly different from the existing studies on retransmission schemes [24–27] for terrestrial wireless networks in terms of system models, assumptions, and algorithms. For example, the study in [24] assumes a slotted and time synchronized channel and also assumes that the transmitting node can detect packet collision during transmission. In [24], it is also assumed that the number of blocked stations is known for optimal retransmission. Those assumptions are not practical in underwater networks due to a high propagation delay. In contrast, our work does not require time synchronization, packet collision detection during transmission, and information on the number of blocked stations.

The authors of [25] assumed that the base station knows which nodes would transmit a packet in advance and the base station monitors whether or not all expected packets are successfully received. Then, it uses a separate control channel to transmit a busy signal to all successful nodes until all collided packets are retransmitted successfully. Our protocol does not use a separate control channel and nodes do not need to wait until all collided packets are retransmitted successfully.

Our work also significantly differs from the studies in [26, 27]. The work in [26, 27] considered a network that consists of transmitter-only nodes, which have only an RF transmitter without an RF receiver. The sending nodes transmit each packet fixed and predetermined times; that is, the number of total transmissions of each packet is predetermined before the network is deployed. Also, the work in [26, 27] assumes that the network status (e.g., the number of nodes and network loads) does not change during the network life time. Since the network status information is known and each node transmits each packet predetermined times, finding a solution that maximizes the packet delivery probability is rather simple and straightforward.

In contrast, we assume that the network status varies over time. Therefore, the algorithm repeatedly compares the PDP (packet delivery probability) value when the value of the maximum number of retransmissions is decremented and incremented. This process continues to find the optimal value of the maximum number of retransmissions. Note that this approach involves another algorithm: approximation of the PDP values with the incremented and decremented values of the maximum number of retransmission. In addition, in [26, 27], the off-line optimization formulation was possible since every node transmits the packet predetermined times and thus the total traffic can be controlled. However, in this work, the total traffic can not be known since the number of packet transmissions are not predetermined.

3. System Model

The UASN under consideration has a cluster-based network topology where each underwater sensor node belongs to one cluster governed by a clusterhead. It is known that a cluster-based UASN provides suitable network connectivity and scalability in underwater communication environments [28–30].

Each underwater sensor node transmits sensing data using a direct acoustic channel to its clusterhead, which
performs data aggregation and then forwards the data to the sink node. Clusterheads are equipped with two underwater communication interfaces, one for intracluster communications, the other for intercluster communications. It is assumed that communications in one cluster do not interfere with communications in other clusters because they use different carriers or channels [31]. Assigning channels to adjacent clusters or nodes is a well studied area [32–34].

Each sensor node transmits to the clusterhead a data packet of \( L_{\text{data}} \) bytes, including data payload and the header. The data rate is \( R_b \) bps. Thus, the transmission delay of a data packet is \( t_{tr} = L_{\text{data}} / R_b \). Upon receiving the data packet, the clusterhead immediately responds with an acknowledgement (ACK) packet to the source node.

In this paper, to facilitate presentation, we focus on an arbitrary cluster that has \( N \) underwater sensor nodes. Each sensor node can transmit to the clusterhead the same copy of the original packet up to \( x \) times, including both original and retransmitted packets, if it has not received an ACK packet within the ACK timeout interval.

Also, the packet delivery probability represents the successful delivery probability of a packet when the packet can be transmitted up to \( x \) times. Meanwhile, the packet delivery ratio (PDR) refers to the ratio of the number of successfully delivered packets to the number of the packets transmitted, which is usually collected by simulations and experiments.

4. Algorithms

In this section, we describe the detailed algorithm of ARS. ARS selects an optimal value of \( x \) (the maximum number of retransmissions) to maximize packet delivery probability (PDP), which leads to a high PDR and a low end-to-end delay.

First, we discuss the assumption that packet arrivals follow a Poisson process, and we justify that the assumption is acceptable in a UASN where underwater nodes may perform exponential back-off and carrier sensing. Then, we elaborate on how to obtain the PDP value with the maximum number of retransmissions, \( x \). We also discuss the estimation of PDP values with different \( x \) values, which also involves the approximation of network load changes over different \( x \) values. Finally, we describe the selection of an optimal value of \( x \) based on PDP estimation.

4.1. Preliminary. When the packet arrivals follow a Poisson process, the probability of \( k \) packets’ arrival during an interval of time, \( t \), is given by

\[
P[n = k] = e^{-\lambda t} \frac{(\lambda t)^k}{k!}, \quad k = 0, 1, 2, \ldots, \tag{1}
\]

where \( \lambda \) represents the arrival rate of the background traffic in a time interval of \( t \) \((t > 0)\) [35].

In this paper, the arrival rate of the background traffic from the other \( N - 1 \) nodes (except the current node) is assumed to follow a Poisson process and is denoted by \( \lambda_b \).

Now, suppose that a data packet arrives at the clusterhead at time \( t_r \) with a transmission delay of \( t_{tr} \). In order for the data packet not to collide at the clusterhead, none of the packets from the other \( N - 1 \) nodes should arrive at the clusterhead during the interval \([t_0 - t_{tr}, t_0 + t_{tr}]\).

Therefore, the probability that a data packet is successfully delivered without retransmission, \( P_s \), at the clusterhead is given by

\[
P_s = e^{-2\lambda_b t_r}. \tag{2}
\]

4.2. Estimating PDP with the Current Maximum Number of Retransmissions. In this subsection, we first calculate \( P_s \). Then, we extend our discussion to obtain the packet delivery probability with up to \( x \) retransmissions.

As discussed in Section 4.1, in order to calculate \( P_s \), the information needed is the arrival rate of background traffic generated by the other \( N - 1 \) nodes.

In ARS, each node periodically reports to the clusterhead the load it has generated. More specifically, an arbitrary node \( i \) counts the number of original packets transmitted (say \( i_{ori} \)) and the total number of transmitted packets, including those retransmitted (say \( c_{ori} \)), at each time interval \( T_i \). Then, node \( i \) appends the values of \( c_{ori} \) and \( c_{tot} \) to the data packet header and sends it to the clusterhead.

Also, let \( C_{ori} \) and \( C_{tot} \) denote the total number of original packets and the total number of packets transmitted, respectively, by all sensor nodes in the network during \( T_i \). Then, the values of \( C_{ori} \) and \( C_{tot} \) can be approximated by the clusterhead as follows:

\[
C_{ori} = \sum_{i=1}^{N} c_{ori}^i, \tag{3}
\]

\[
C_{tot} = \sum_{i=1}^{N} c_{tot}^i. \tag{4}
\]

The clusterhead then calculates the average number of retransmissions for each packet, \( r_c \) (i.e., \( r_c = C_{tot}/C_{ori} \)). Thus, the arrival rate of background traffic generated by the other \( N - 1 \) nodes during the interval of \( T_i \) can be calculated as follows:

\[
\lambda_b = \frac{C_{tot}}{T_i} \times \frac{N - 1}{N} = r_c \times \frac{C_{ori}}{T_i} \times \frac{N - 1}{N}. \tag{5}
\]

Then, the probability that a single packet transmission is successfully delivered to the clusterhead can be calculated according to (2) using the rate of background traffic calculated by (4). Also, the probability that a single packet transmission fails can be calculated as \( P_f = 1 - P_s \).

Now, we discuss the calculation of the PDP when a packet can be retransmitted up to \( x \) times. In order to facilitate discussion, we define \( P_{s;[j]} \) and \( P_{f;[j]} \) as the probability of the successful and failed delivery of the \( j \)th transmission of a packet, respectively.

Also, let \( P(x) \) denote the PDP with up to \( x \) retransmissions. Then, \( P(x) \) becomes

\[
P(x) = P_{s;[1]} + P_{f;[1]}P_{s;[2]} + \cdots + P_{f;[j]} \cdots P_{f;[x-1]}P_{s;[x]} . \tag{6}
\]

Since each packet transmission can be regarded as an independent event based on the assumption of a Poisson
process, $P_{(s,j)} = P_s$ and $P_{(f,j)} = P_f$ for all $j$. Therefore, PDP can be expressed as

$$P(x) = P_s \frac{1 - P f^x}{1 - P f} = 1 - P f^x = 1 - \left(1 - e^{-2\lambda f^u} \right)^x.$$  \hspace{1cm} (6)

4.3. Estimating PDP with the Maximum Number of Retransmissions of $x+1$ and $x-1$. Now, we estimate PDP values with two different values of the maximum number of retransmissions, $x+1$ and $x-1$. Note that (6) cannot be directly used to estimate PDP with $x+1$ and $x-1$, since the arrival rate of background traffic, $\lambda_t$, will be altered with a different value of $x$. Instead, we use an observation from (4) that the arrival rate $\lambda_t$ depends on the average number of retransmissions, $r$. More specifically, we first approximate the average number of retransmissions over different values of $x$ to estimate the new arrival rate $\lambda_t$, which will be used to obtain new PDP values.

First, let $r_{x+1}$ and $r_{x-1}$ denote the average actual retransmissions when the maximum retransmissions are $x+1$ and $x-1$.

In addition, we define a random variable, $Y_x$, which represents actual retransmissions when the maximum number of retransmissions is $x$. Without loss of generality, we assume that $x$ is an integer greater than 1. Then, the expected number of transmissions, $E(Y_x)$, can be calculated using the probability that the actual number of retransmissions is exactly $k$, where $k = 1, \ldots, x$. That is,

$$E(Y_x) = \sum_{k=1}^{x} k \times P(Y_x = k)$$

$$= P_s + \sum_{k=2}^{x-1} k \times P_{f}^{k-1} \times P_s + x \times P_{f}^{x-1}.$$ \hspace{1cm} (7)

Note that there is no $P_i$ in the last term in (7), since, whether the $i$th transmission is successful or not, the sensor node will not transmit the packet any more.

Now, we take into account the fact that, for a given integer $n$, $\sum_{k=1}^{n} k r^{k-1} = (1 - r^{n+1})/(1 - r) - (n + 1) r^n/(1 - r)$, where $r \in \mathbb{R}$ and $r \neq 1$. Then, (7) can be expressed as follows:

$$E(Y_x) = P_s + P_{f} \left( \frac{1 - P f^x}{1 - P f} \right)^2 \times \frac{x \times P_{f}^{x-1}}{1 - P f} - 1 \right) + x \times P_{f}^{x-1}$$

$$= P_s + \left( \frac{1 - P f^x}{1 - P f} - x \times P_{f}^{x-1} - P_s \right) + x \times P_{f}^{x-1}$$

$$= \frac{1 - P f^x}{1 - P f} - P_s.$$ \hspace{1cm} (8)

Now, expected retransmissions with maximum retransmissions of $x+1$ and $x-1$, $E(Y_{x+1})$ and $E(Y_{x-1})$ are approximated by replacing $x$ with $x+1$ and $x-1$, respectively. Also, let $\delta_{inc} = E(Y_{x+1}) - E(Y_x)$ and $\delta_{dec} = E(Y_x) - E(Y_{x-1})$.

Then, we approximate values of $r_{x+1}$ and $r_{x-1}$ using $\delta_{inc}$ and $\delta_{dec}$, as follows:

$$r_{x+1} = r_x + \beta \times \delta_{inc}$$

$$r_{x-1} = r_x - \beta \times \delta_{dec},$$ \hspace{1cm} (9)

where $\beta$ is a system parameter that puts a weight on $\delta_{inc}$ and $\delta_{dec}$.

Finally, we estimate the new PDP values using (4) and (6) based on the obtained $r_{x+1}$ and $r_{x-1}$.

Note that the obtained PDP values based on $r_{x+1}$ and $r_{x-1}$ are also approximated values since $r_{x+1}$ and $r_{x-1}$ are calculated based on $P_f$ with the current network load. Our simulation results show that this approximation works well as an indicator to determine whether the value of $x$ should be increased, decreased, or stay the same. In fact, all the information needed is whether the PDP value is increasing or decreasing as $x$ grows, or if the PDP value is around the peak with the current $x$ value.

4.4. Selecting an Optimal Value of the Maximum Number of Retransmissions Using Estimated PDP Values. The main objective of ARS is to keep an optimal value of $x$ that maximizes PDP. Therefore, from among the values of $x, x+1,$ and $x-1$, the clusterhead chooses the one that has the highest corresponding PDP value as the new $x$ value.

Intuitively, when the network load is low, the clusterhead raises the value of $x$ until no higher PDP can be achieved. When the network load is too heavy, on the other hand, the achievable PDP value is low due to network congestion and a high level of packet collisions. In that case, the clusterhead decides to reduce the $x$ value, as long as it can achieve a higher PDP value.

In order to avoid an unnecessary fluctuation of $x$, the clusterhead uses a threshold value for a gain in the PDP value. More specifically, the decision to change the current value of $x$ to other values is made only if the PDP gain is higher than a given threshold value, $\epsilon$. Selection of the $x$ value is repeated at every interval $T_t$ by the clusterhead, which publishes this value to the network. Upon receiving the new $x$ value, each underwater sensor updates the maximum number of transmissions accordingly. Algorithm 1 presents the adaptive selection process.

5. Performance Study

5.1. Simulation Setup. In order to verify that ARS can improve network performance in terms of PDR and E2E delay, we compare the performance of ARS-applied protocols with that of the existing contention-based MAC protocols.

In this paper, we select Aloha and CSMA for performance comparison, since a lot of contention-based MAC protocols are based on Aloha and CSMA. The design, simulate, emulate, and realize test-beds (DESSERT) underwater simulation framework [21] based on NS2 Miracle is used to simulate the protocols in a realistic underwater communication environment.

The cluster considered for the simulation consists of 50 underwater sensor nodes randomly deployed over an area of
Inputs:

\(N\): number of sensor nodes

\(T_l\): time interval

\(x\): maximum number of retransmissions for first interval \(T_l\)

\(t_r\): transmission delay

\(C_{ori}\): total number of original packets in the network after duration of \(T_l\)

\(C_{tot}\): total number of packets transmitted in the network after duration of \(T_l\)

\(r_c\): average number of retransmissions for each packet

\(\beta\): system parameter

\(\epsilon\): threshold value (\(\epsilon > 0\))

Outputs:

\(x_{opt}\): The optimal value of number of retransmissions for next interval \(T_l\)

(1) while (true) do

//Estimate the PDP at current maximum number of retransmissions \(P(x)\):

(2) \(r_c = \frac{C_{tot}}{C_{ori}}\);

(3) \(\lambda_b = r_c \times \frac{C_{ori}}{T_l} \times \frac{(N - 1)}{N}\);

(4) \(P_s = e^{-2\lambda_b t_r}\);

(5) \(P_f = 1 - e^{-2\lambda_b t_r}\);

(6) \(P(x) = 1 - (1 - e^{-2\lambda_b t_r})^x\);

//Estimate the PDP at incremented and decremented value of current maximum retransmissions \(P(x + 1)\), and \(P(x - 1)\):

(7) \(E(Y_x) = \frac{1 - P_f^x}{1 - P_f}\); \(E(Y_{x+1}) = \frac{1 - P_f^{x+1}}{1 - P_f}\); \(E(Y_{x-1}) = \frac{1 - P_f^{x-1}}{1 - P_f}\);

(8) \(\delta_{inc} = E(Y_{x+1}) - E(Y_x); \delta_{dec} = E(Y_x) - E(Y_{x-1})\);

(9) \(r_{x+1} = r_c + \beta \times \delta_{inc}; r_{x-1} = r_c - \beta \times \delta_{dec}\);

(10) \(\lambda_{b(x+1)} = r_{x+1} \times \frac{C_{ori}}{T_l} \times \frac{(N - 1)}{N}\);

(11) \(\lambda_{b(x-1)} = r_{x-1} \times \frac{C_{ori}}{T_l} \times \frac{(N - 1)}{N}\);

(12) \(P(x + 1) = 1 - \left(1 - e^{-2\lambda_{b(x+1)} t_r}\right)^{x+1}\);

(13) \(P(x - 1) = 1 - \left(1 - e^{-2\lambda_{b(x-1)} t_r}\right)^{x-1}\);

//Select the optimal value of maximum number of retransmissions \(x_{opt}\):

(14) if \(P(x + 1) > P(x - 1)\) then

(15) if \(P(x + 1) - P(x) \geq \epsilon\) then

(16) \(x_{opt} = x + 1;\) \(\triangleright\) Increase \(x\)

(17) else

(18) \(x_{opt} = x;\) \(\triangleright\) Remain \(x\)

(19) end if

(20) else

(21) if \(P(x - 1) - P(x) \geq \epsilon\) then

(22) \(x_{opt} = x - 1;\) \(\triangleright\) Decrease \(x\)

(23) else

(24) \(x_{opt} = x;\) \(\triangleright\) Remain \(x\)

(25) end if

(26) end if

(27) Output \(x_{opt}\);

(28) Re-read \(C_{ori}, C_{tot}\);

(29) end while

Algorithm 1: The adaptive selection algorithm.
1555 m × 1555 m. Each underwater sensor node is equipped with a half-duplex acoustic transceiver that has a data rate of 14 kbps at a distance of 1100 m. It is assumed that each underwater sensor node periodically generates a data packet of 160 bytes and sends the data packet to the clusterhead. The speed of underwater acoustic signals is assumed to be 1500 m/s.

The back-off time is calculated as back-off-duration = \(\text{rand}() \times 2^{\text{counter}} \times s\) where counter is the current number of retransmission times (beginning from 0) of the packet and \(s\) is set to 0.1. The initial value of \(x\) for ARS is set to 5. The value of factor \(\beta\) is set to 2.

5.2. Simulation Results. We analyze network performance in terms of PDR and average end-to-end delay. First, we discuss the effects of network load on network performance. The dynamic network load during the simulation is also considered to show that ARS can adaptively find an optimal value of \(x\), adapting to the network load. For example, when network load is low (e.g., 1 kbps), the achieved PDR by CSMA-ARS is similar to CSMA with a large value of the maximum retransmissions (e.g., 5 and 7). On the other hand, when network load is high (e.g., from 4 kbps to 6 kbps), CSMA-ARS can achieve a similar performance to CSMA with the maximum number of retransmissions of 1. CSMA-ARS has a slightly lower PDR than CSMA with \(x = 1\) when the network load is between 4 kbps and 6 kbps. This is because the initial value of \(x\) in CSMA-ARS is 5 and it takes time to reach an optimal point.

Figure 1(a) also indicates that if network load varies over time, CSMA-ARS can achieve over 20% higher PDR than CSMA with any value of \(x\), by adaptively choosing an optimal value of the maximum retransmissions over different network loads.

The end-to-end delays over different network loads are compared in Figure 1(b). As shown in Figure 1(b), CSMA-ARS shows a low delay compared to CSMA with \(x = 3, 5,\) and 7. In particular, CSMA with \(x = 5\) and 7 shows a delay longer than 20 seconds in some cases due to a large number of packet collisions and backoffs. CSMA with \(x = 1\) has the lowest delay by sacrificing PDR at a low network load. Note that the E2E delay reaches a peak and decreases, since most packets are dropped under a very high network load, and those dropped packets are not considered when calculating the delay.

ARS keeps adjusting the value of maximum number of retransmissions, \(x\), such that the estimated PDP value is maximized. From Figures 1(a) and 1(b), the approach that ARS takes can actually achieve a higher PDR value with a low E2E delay compared to the CSMA protocol.

Figure 2 compares the performance of Aloha with ARS and Aloha as network load varies. Similar to the simulation...
under CSMA, as shown in Figure 2, Aloha-ARS can always obtain the PDR value that is close to the maximum PDR value that Aloha can achieve using different $x$ values. More specifically, when network load is low (1 kbps or 2 kbps) Aloha-ARS can achieve similar PDR and E2E delay values to Aloha that uses a high value of $x$ ($x = 5$ or $x = 7$). On the other hand, the achieved PDR and E2E delay values of Aloha-ARS are similar to those of Aloha when Aloha uses a low value of $x$ ($x = 1$) at the higher network load (5 kbps or 6 kbps).

Another interesting point is that, as shown in Figures 1(a) and 2(a), the PDR of Aloha decreases more sharply than CSMA as the network load grows. This is because more packet collisions can occur in Aloha under a high network load due to the lack of carrier sensing. However, Aloha-ARS and CSMA-ARS show a similar PDR over different network loads, which indicates that ARS can lower the number of packet collisions by adaptively changing the value of $x$.

Figure 3 compares the PDR and delay between CSMA-ARS and Aloha-ARS. As shown in Figure 3, when network load is low, CSMA-ARS and Aloha-ARS achieve a similar PDR, which is close to one. However, when network load is relatively high (around 3 kbps), CSMA-ARS shows a higher PDR since carrier sensing can reduce packet collisions. In case network load is very high, both protocols achieve relatively low PDR values due to the limited channel capacity. In addition, as shown in Figure 3, Aloha-ARS always achieves a lower delay because it does not have latency for carrier sensing.

In order to show the detailed operation of ARS, Figure 4 compares $P(x)$, $P(x - 1)$, and $P(x + 1)$ and the instantaneous PDR values during each round when the network load is 4 kbps, and it also shows how ARS interacts with these values. The instantaneous PDR, which is collected from simulations, is defined as the ratio of the number of packets transmitted to the channel within one round. In Figures 4(a) and 4(b), the $X$-axis represents the round with $x$ value determination. In Figure 4(a), the $Y$-axis represents the selected value of $x$, and, in Figure 4(b), it represents the estimated $P(x)$, $P(x - 1)$, $P(x + 1)$ and the instantaneous PDR.

As shown in Figure 4(a), the value of $x$ is initially 5. During round 1, the clusterhead compares $P(x)$ with $P(x - 1)$ and $P(x + 1)$ based on the traffic reported by sensor nodes. Since $P(x - 1)$ has a higher value, the $x$ decreases to 1 until round 4. This value of $x$ remains at 1 until round 6. Then,
at round 7, the clusterhead decides to increase the value of $x$ to 2, since $P(x + 1)$ is higher than $P(x)$ and keeps this value during the rest of the simulation. The value $x = 2$ is considered as an optimal value of $x$ at a network load of 4 kbps. Figure 4(b) also shows that the estimated $P(x)$ (i.e., PDP) can closely approximate the instantaneous PDR as the network load becomes stable after round 7 (i.e., the network load is relatively unstable until round 7 due to the rapid change of $x$).

Figure 5 indicates the detailed operation of Aloha-ARS when the network load is 4 kbps. Initially, the value of $x$ is 5. Similar to the CSMA-ARS scenario, $x$ decreases to 1 until round 4. There is a fluctuation in $x$ from round 5 to round 8 due to the unstable network traffic as the $x$ value changes.

Figure 4: CSMA: (a) adaptive maximum number of retransmissions and (b) comparison of instantaneous PDR and $P(x)$.

Figure 5: Aloha: (a) adaptive maximum number of retransmissions and (b) comparison of instantaneous PDR and $P(x)$. 
The value of $x$ remains at 2 from round 9 to the end of the simulation since the clusterhead determines the optimal value of $x$ is 2. Figure 5(b) also shows that the estimated $P(x)$ (i.e., PDP) can closely approximate the instantaneous PDR.

5.2.2. Varying Network Load Over Time. In some sensor network applications, sensors may increase the sensor data transmission rate when specific events occur or some conditions are satisfied (e.g., a specific level of vibration or temperature).

In order to see how ARS can adapt to a change in network load over time, every node varies its packet generation rate over a simulation time of 245,000 seconds. More specifically, from the beginning, each node has a rate of 20 bps for 96,000 seconds, which results in a network load of 1 kbps. Then, the generation rate of each node increases to 40 bps in the next round time period of 44,000 seconds (the network load becomes 2 kbps). For the next 30,000 seconds, the rate of each node becomes 80 bps, and then it becomes 120 bps (the network load is 6 kbps) for the next 20,000 seconds. Then, each node decreases its traffic rate to 20 bps for the rest of the simulation.

Figures 6(a) and 6(b) compare the instantaneous PDR (or "inst. PDR" for short) of CSMA-ARS and Aloha-ARS with those of CSMA and Aloha with different values for the maximum number of retransmissions.

From Figure 6(a), we can see that CSMA-ARS adapts well to the change in network load and achieves the highest or near the highest inst. PDR over the entire simulation time. When the network load is low, ARS selects a high $x$ value for maximum retransmissions to obtain a high PDR value. For example, in the interval from 0 to 96,000 seconds, CSMA-ARS shows a PDR (around 0.99) similar to CSMA with $x = 5$ and $x = 7$. On the other hand, when the network load is high, ARS selects a low $x$ value for maximum retransmissions to avoid excessive collisions. For instance, CSMA-ARS obtains an inst. PDR (around 0.5 and 0.38) similar to CSMA with $x = 1$ from 140,000 to 190,000 seconds, when network load is very high.

In contrast, the original CSMA protocol cannot adapt to the network load changes and shows poor performance, depending on network load. For example, CSMA with $x = 5$ shows a PDR value of around 0.99 in the time period between 0 and 96,000 seconds. However, it obtains a PDR value lower than 0.2 between 170,000 and 190,000 seconds, whereas CSMA-ARS achieves a PDR value of around 0.38 in the period. Similarly, CSMA with $x = 1$ only obtains a PDR of less than 0.85, on average, when network load is low (from 0 to 96,000 seconds) whereas CSMA-ARS can achieve a PDR of around 0.99.

The results are also similar when Aloha-ARS is compared with Aloha that uses different $x$ values, as shown in Figure 6(b). Aloha-ARS can adaptively determine an optimal value of $x$ based on changing network load over the simulation time, so it can also achieve the highest or near the highest inst. PDR value.

In fact, ARS shows a higher advantage in this case since Aloha is more sensitive to the network load. For example, Aloha with $x = 5$ only obtains a PDR value lower than 0.1 when the network load is high (from 170,000 and 190,000 seconds), whereas it achieves a PDR value of around 0.99 when the network load is low. In contrast, Aloha-ARS shows a consistently high PDR value compared to the original Aloha. It can also be seen that Aloha-ARS has similar PDR values to CSMA-ARS in most time periods.

Figure 7 compares the average PDR and end-to-end delay of CSMA-ARS and Aloha-ARS with those of CSMA and
Figure 7: Average PDR when varying the network load over time (a) under CSMA and (b) under Aloha.

5.2.3. Effects of the Number of Data Flows. In this subsection, the effects of the number of data flows on network performance are examined. The number of data flows varies from 10 to 50, whereas the node load remains at 80 bps. CSMA and Aloha are tested with different \( x \) values, from 1 to 7.

As shown in Figures 8 and 9, CSMA-ARS and Aloha-ARS achieve the highest PDR values in most cases, without a significant E2E delay, by adaptively selecting an \( x \) value appropriate to the network status. For example, if CSMA with \( x = 1 \) is used for the network and the number of data flows is 20, CSMA can only obtain a PDR around 0.83, whereas CSMA-ARS can achieve a PDR higher than 0.98. Similarly, if CSMA with \( x = 5 \) is used, and the number of data flows becomes 40, CSMA-ARS achieves a 23% higher PDR and an average E2E delay nine times shorter than CSMA.

The results shown in Figure 9 also indicate that a higher PDR gain can be obtained when the ARS scheme is applied to Aloha. Therefore, we can say that CSMA-ARS and Aloha-ARS can find an optimal value of the maximum retransmissions over different numbers of flows, which results in higher performance than the original CSMA and Aloha.

In Figure 10, the performances of CSMA-ARS and Aloha-ARS are compared when the number of data flows changes. As shown in Figure 10, both protocols show a similar PDR. When the number of flows is 30, CSMA-ARS obtains a slightly higher PDR. With respect to E2E delay, CSMA-ARS always shows a longer delay due to carrier sensing before data transmission.

5.2.4. Effects of the Weighted Factor \( \beta \). In Section 4.3, a system parameter \( \beta \) is used when the values of \( r_{x+1} \) and \( r_{x-1} \) are estimated in (9). Recall that \( r_{x+1} \) and \( r_{x-1} \) represent the average retransmissions when maximum retransmissions are \( x + 1 \) and \( x - 1 \), respectively.

Since the value of \( \beta \) affects the accuracy of the \( r_{x+1} \) and \( r_{x-1} \) value estimations, in this subsection, we examine how network performance will change based on different values of \( \beta \).

As shown in Figure 11, the value of \( \beta \) varies from 1.0 to 2.5. CSMA-ARS with various values of \( \beta \) is also compared to CSMA with \( x = 5 \) in terms of PDR and E2E delay.

We can see that network performance can be improved by selecting an appropriate value for \( \beta \). For example, CSMA-ARS with \( \beta = 1.0 \) obtains a PDR value of around 0.55 when the network load is 3 kbps. However, when the value of \( \beta \) is 2.0, a PDR value of 0.68 can be obtained. As also shown
in Figure 11(a), there is no significant gain in PDR when the value of $\beta$ is greater than 2.0. Therefore, $\beta = 2.0$ is used for simulations in this paper. The value of $\beta$ does not show a remarkable impact on the delay as shown in Figure 11(b). We can see that the performance of CSMA with $x = 5$ is much lower than CSMA-ARS with all values of $\beta$, particularly at high network loads.

6. Conclusions and Future Work

In order to address the problems of a low packet delivery ratio and high end-to-end delay under contention-based MAC protocols in UASNs, we have proposed a novel adaptive retransmission scheme, named ARS, which dynamically selects an optimal value of the maximum number of retransmissions such that the successful delivery probability
of a packet is maximized for a given network load. ARS periodically compares the current PDP against estimated PDPs with incremented and decremented values for the maximum number of retransmissions. Then, ARS selects a new value for the maximum retransmissions to achieve a higher PDP value. In this paper, we have applied ARS to Aloha and CSMA in order to evaluate the performance gain. According to simulation results, ARS can significantly improve network performance in terms of PDR and E2E delay.

For future work, we plan to extend ARS to support different performance requirements in UASNs such that each node can adapt its transmissions to satisfy a specific performance requirement from applications or users.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the 2013 Research Fund of University of Ulsan.
References

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: research challenges,” Ad Hoc Networks, vol. 3, no. 3, pp. 257–279, 2005.

[2] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, “Research challenges and applications for underwater sensor networking,” in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’06), pp. 228–235, Las Vegas, Nev, USA, April 2006.

[3] J. Heidemann, M. Stojanovic, and M. Zorzi, “Underwater sensor networks: applications, advances and challenges,” Philosophical Transactions of the Royal Society A, vol. 376, no. 1958, pp. 158–175, 2012.

[4] J. G. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, “Shallow water acoustic networks,” IEEE Communications Magazine, vol. 39, no. 11, pp. 114–119, 2001.

[5] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “Aloha-based MAC protocols with collision avoidance for underwater acoustic networks,” in Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM ’07), pp. 2271–2275, Anchorage, Alaska, USA, May 2007.

[6] J. Ahn, A. Syed, B. Krishnamachari, and J. Heidemann, “Design and analysis of a propagation delay tolerant ALOHA protocol for underwater networks,” Ad Hoc Networks, vol. 9, no. 5, pp. 752–766, 2011.

[7] P. Mandal, S. De, and S. S. Chakraborty, “A receiver synchronized slotted Aloha for underwater wireless networks with imprecise propagation delay information,” Ad Hoc Networks, vol. 11, no. 4, pp. 1443–1455, 2013.

[8] L. Vieira, J. Kong, U. Lee, and M. Gerla, “Analysis of aloha protocols for underwater acoustic sensor networks,” in Proceedings of the 1st International Workshop on Underwater Networks (WUWNet ’06), Los Angeles, Calif, USA, September 2006.

[9] S. M. Smith, J. C. Park, and A. Neel, “A peer-to-peer communication protocol for underwater acoustic communication,” in Proceedings of the MTS/IEEE Conference (OCEANS ’97), pp. 268–272, IEEE, Halifax, Canada, October 1997.

[10] M. Molins and M. Stojanovic, “Slotted FAMA: a MAC protocol for underwater acoustic networks,” in Proceedings of the OCEANS 2006—Asia Pacific, pp. 1–7, Singapore, May 2007.

[11] D. Fang, Y. Li, H. Huang, and L. Yin, “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 ’10), pp. 1–4, Chengdu, China, September 2010.

[12] D. Wang, X. Hu, F. Xu, H. Chen, and Y. Wu, “Performance analysis of P-CSMA for underwater acoustic sensor networks,” in Proceedings of the OCEANS, pp. 1–6, Yeosu, Republic of Korea, May 2012.

[13] L. Jin and D. Huang, “A slotted CSMA based reinforcement learning approach for extending the lifetime of underwater acoustic wireless sensor networks,” Computer Communications, vol. 36, no. 9, pp. 1094–1099, 2013.

[14] C. Petrioli, R. Petroccia, and M. Stojanovic, “A comparative performance evaluation of MAC protocols for underwater sensor networks,” in Proceedings of the OCEANS 2008, pp. 1–10, Quebec City, Canada, September 2008.

[15] X. Guo, M. R. Frater, and M. J. Ryan, “A propagation-delay-tolerant collision avoidance protocol for underwater acoustic sensor networks,” in Proceedings of the Asia Pacific (OCEANS ’06), pp. 1–6, Singapore, May 2007.

[16] B. Peleato and M. Stojanovic, “Distance aware collision avoidance protocol for ad-hoc underwater acoustic sensor networks,” IEEE Communications Letters, vol. 11, no. 12, pp. 1025–1027, 2007.

[17] A. A. Syed, W. Ye, and J. Heidemann, “T-Lohi: a new class of MAC protocols for underwater acoustic sensor networks,” in Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08), pp. 789–797, Phoenix, Ariz, USA, April 2008.

[18] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “RIPT: a receiver-initiated reservation-based protocol for underwater acoustic networks,” IEEE Journal on Selected Areas in Communications, vol. 26, no. 9, pp. 1744–1753, 2008.

[19] N. Chirdchoo, W.-S. Soh, and K. C. Chua, “MACA-MN: a MACA-based MAC protocol for underwater acoustic networks with packet train for multiple neighbors,” in Proceedings of the IEEE 67th Vehicular Technology Conference-Spring (VTC ’08), pp. 46–50, May 2008.

[20] Y. Zhu, Z. Jiang, Z. Peng, M. Zuba, J.-H. Cui, and H. Chen, “Toward practical MAC design for underwater acoustic networks,” in Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM ’13), pp. 683–691, IEEE, Turin, Italy, April 2013.

[21] R. Masiero, S. Azad, F. Favaroli et al., “DESSERT Underwater: an NS-Miracle-based framework to design, simulate, emulate and realize test-beds for underwater network protocols,” in Proceedings of the MTS/IEEE Yeosu Conference: The Living Ocean and Coast—Diversity of Resources and Sustainable Activities (OCEANS ’12), pp. 1–10, May 2012.

[22] C.-S. Hsu and X.-Z. Hong, “An efficient scheduling and random access hybrid medium access control protocol for underwater sensor networks,” in Proceedings of the 15th International Symposium on Wireless Personal Multimedia Communications (WPNC ’12), pp. 207–211, Taipei, Taiwan, September 2012.

[23] D. Pompili, T. Melodia, and I. F. Akyildiz, “A CDMA-based medium access control for underwater acoustic sensor networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 4, pp. 1899–1909, 2009.

[24] J. S. Meditch and C. T. A. Lea, “Stability and optimization of the csma and csma/cd channels,” IEEE Transactions on Communications, vol. 31, no. 6, pp. 763–774, 1983.

[25] M. Z. Ali and M. Torlak, “Retransmission optimization in CDMA random access networks,” in Proceedings of the Military Communications Conference (MILCOM ’02), pp. 799–803, Anaheim, Calif, USA, October 2002.

[26] R. S. Sudhaakar, S. Yoon, J. Zhao, and C. Qiao, “A novel QoS-aware MAC scheme using optimal retransmission for wireless networks,” IEEE Transactions on Wireless Communications, vol. 8, no. 5, pp. 2230–2235, 2009.

[27] S. Yoon, C. Qiao, R. S. Sudhaakar, J. Li, and T. Talty, “QoMOR: a QoS-aware MAC protocol using optimal retransmission for wireless intra-vehicular sensor networks,” in Proceedings of the Mobile Networking for Vehicular Environments (MOVE ’07), pp. 121–126, IEEE, Anchorage, Alaska, USA, May 2007.

[28] I. F. Akyildiz, D. Pompili, and T. Melodia, “State of the art in protocol research for underwater acoustic sensor networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 11, no. 4, pp. 11–22, 2007.

[29] M. C. Domingo and R. Prior, “A distributed clustering scheme for underwater wireless sensor networks,” in Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC ’07), pp. 1–5, Athens, Ga, USA, September 2007.
[30] S. Saxena, S. Mishra, and M. Singh, “Clustering based on node density in heterogeneous under-water sensor network,” *International Journal of Information Technology and Computer Science*, vol. 5, no. 7, pp. 49–55, 2013.

[31] J. Jagannath, A. Saji, H. Kulhandjian, Y. Sun, E. Demirors, and T. Melodia, “A hybrid MAC protocol with channel-dependent optimized scheduling for clustered underwa ter acoustic sensor networks,” in *Proceedings of the 8th ACM International Conference on Underwater Networks and Systems (WUWNet ’13)*, Kaohsiung, Taiwan, November 2013.

[32] H. Skalli, S. Ghosh, S. K. Das, L. Lenzini, and M. Conti, “Channel assignment strategies for multiradio wireless mesh networks: issues and solutions,” *IEEE Communications Magazine*, vol. 45, no. 11, pp. 86–95, 2007.

[33] A. Naveed and S. S. Kanhere, “Cluster-based channel assignment in multi-radio multi-channel wireless mesh networks,” in *Proceedings of the IEEE 34th Conference on Local Computer Networks (LCN ’09)*, pp. 53–60, Zurich, Switzerland, October 2009.

[34] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, “Interference-aware channel assignment in multi-radio wireless mesh networks,” in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM ’06)*, pp. 1–12, Barcelona, Spain, April 2006.

[35] A. Papoulis and S. U. Pillai, “Random walks and other applications,” in *Probability, Random Variables and Stochastic Processes*, chapter 10, p. 456, McGraw-Hill, 4th edition, 2002.
Submit your manuscripts at http://www.hindawi.com