A Hybrid MAC Protocol with Channel-dependent Optimized Scheduling for Clustered Underwater Acoustic Sensor Networks

Jithin Jagannath, Anu Saji, Hovannes Kulhandjian, Yifan Sun, Emrecan Demiros, and Tommaso Melodia
Department of Electrical Engineering
State University of New York at Buffalo, Buffalo, New York 14260
Email: { jithinja, anusaji, hkk2, ysun24, emrecand, tmelodia}@buffalo.edu

ABSTRACT

We propose a novel optimal time slot allocation scheme for clustered underwater acoustic sensor networks that leverages physical (PHY) layer information to minimize the energy consumption due to unnecessary retransmissions thereby improving network lifetime and throughput. To reduce the overhead and the computational complexity, we employ a two-phase approach where: (i) each member node takes a selfish decision on the number of time slots it needs during the next intra-cluster cycle by solving a Markov decision process (MDP), and (ii) the cluster head optimizes the scheduling decision based on the channel quality and an urgency factor. To conserve energy, we use a hybrid medium access scheme, i.e., time division multiple access (TDMA) for the intra-cluster communication phase and carrier sense multiple access with collision avoidance (CSMA/CA) for the cluster head-sink communication phase. The proposed MAC protocol is implemented and tested on a real underwater acoustic testbed using SM-75 acoustic modems by Teledyne Benthos. Simulations illustrate an improvement in network lifetime. Additionally, simulations demonstrate that the proposed scheduling scheme with urgency factor achieves a throughput increase of 28% and improves the reliability by up to 25% as compared to the scheduling scheme that neither use MDP nor optimization. Furthermore, testbed experiments show an improvement in throughput by up to 10% along with an improvement in reliability.

1. INTRODUCTION

Prolonging the network lifetime is crucial for long-term monitoring missions such as oceanographic data collection, pollution monitoring, offshore exploration, among others. Thus, in this work we propose a medium access control (MAC) protocol for underwater acoustic sensor networks (UW-ASNs) with the objective to maximize the network lifetime and improve the throughput, while maintaining a reliable service.

UW-ASNs present severe challenges to the development of efficient MAC protocols. A number of MAC protocols have been proposed that aim at combating multipath, long propagation delay, and overall energy consumption in UW-ASNs [1, 2, 3, 4]. The energy of the underwater acoustic (UW-A) modem's battery is limited and replacing and/or recharging the battery is in many cases not practical. Therefore, it is crucial to design a MAC protocol that minimizes the overall energy consumption to extend the network lifetime. An important factor to take into consideration is the large transmit power of the UW-A modems, which is in the range of 1.78W-20 W [5], compared to 88.2 mW [6] in terrestrial RF sensor networks. This work aims at minimizing the energy consumption due to unnecessary retransmissions and overhead, which is the main cause of energy depletion in UW-ASNs. The underwater environment is highly noisy due to shipping activities, wind, marine organisms, which results in low signal to noise ratio (SNR), i.e., high bit error rate (BER) in the order of 10^{-4} to 10^{-2} [1, 7], leading to unreliable communication. The severely impaired UW-A channel results in numerous retransmissions due to packet collisions. To overcome this challenge, we take link quality into consideration in our protocol design, with an intention to reduce the packet drop rate and hence, reduce packet retransmissions. We leverage PHY layer information by considering the link quality (i.e., SNR) and residual energy level of the modem's battery with an objective to maximize the network lifetime.

We adopt a clustered topology to further optimize the network energy consumption, owing to its inherent ability to support data gathering [8]. We consider a TDMA scheme for communication within a cluster. The time-slotted structure helps to save energy, by letting the nodes not involved in transmission to go to sleep mode. A contention based CSMA/CA scheme is used for communication between the cluster head and the sink. All of the aforementioned functionalities of our MAC protocol are designed with the objective to extend the network lifetime of UW-ASNs. Thus, in this paper, we present a novel optimized-scheduling scheme, which can be incorporated in any dynamically scheduled MAC protocols for UW-ASNs, with the aim to maximize the energy efficiency and improve the throughput. We have implemented and tested the proposed MAC protocol on a real underwater acoustic testbed using SM-75 acoustic modems by Teledyne Benthos [5]. Simulations and testbed experiments show that significant improvement in throughput and reliability can be achieved, while minimizing the overall energy consumption due to unnecessary retransmissions.

The rest of this paper is organized as follows. In Section 2, we review the related work. We discuss the system architecture in Section 3, while in Section 4, we describe intra-cluster and cluster head-sink communication phase. The time slot allocation using MDP and optimization problem formulation are discussed in detail in Section 5. We present the simulation and testbed evaluation results in Sections 6 and 7 respectively, followed by conclusions in Section 8.

Acknowledgment: This work was partially supported by the National Science Foundation under grants CNS-1055945 and CNS-1126357.

WUWNet'13, November 11 - 13, 2013, Kaohsiung, Taiwan.

http://dx.doi.org/10.1145/2532378.2532382 .

2. RELATED WORK

Several MAC protocols have been proposed to control the medium access of sensor nodes in UW-A channel. In [9], Peleato and Stojanovic adapted the MACA [10] protocol for UW-ASNs, by adjusting the time required for RTS/CTS handshake between sender-receiver pairs by calculating the distance between them. To transmit a data packet, a node initially sends an RTS to the intended receiver. Upon reception of the RTS, the receiver transmits a CTS to the sender and waits for the data packet from the sender. The wait time is calculated based on the round trip time measured during the RTS/CTS exchange. The sender and receiver use the wait time and short warning messages to prevent data packet collisions that may arise from neighboring node transmissions.

In [11], a hybrid MAC for underwater wireless sensor networks (UWSNs) was proposed, which includes scheduled and non-scheduled periods in a time slotted frame. In the scheduled portion of the frame, the sensor nodes communicate in a collision-free manner based on a predefined schedule, while in the non-scheduled portion of the frame, the protocol operates in a contention-based manner by letting nodes adapt to changing traffic conditions. The number of scheduled slots is defined by the network operator or clustering protocol and remains constant once the network operation begins. The non-scheduled slots are temporarily assigned to the nodes based on the distributed state. In [11], the authors assume that the nodes are synchronized.

In [12], an energy efficient MAC protocol called UWAN-MAC was proposed, which aims to minimize the energy consumption by determining listen cycles and sleep period to turn off its transceiver circuit to save energy. The protocol proposed in [13], (called T-Lohi), is a contention based MAC protocol where sensor nodes contend by sending a short tone to reserve the channel for data transmission. T-Lohi conserves energy by allowing the sensor nodes to sleep until they hear the wake up tone. The wake up tone is also very short and thus reduces energy consumption due to small overhead.

There are several other works proposed for channel-dependent scheduling of sensor nodes in terrestrial wireless sensor networks (WSN). Among these, [14] is a TDMA-based MAC protocol, which uses link state dependent scheduling (LSDS) and gathers samples of the channel quality, i.e., label slots as bad if transmission/reception fails or good if transmission/reception is successful. All patterns of good and bad states are used as a training set to establish a prediction set that is the shortest repetition of the training set. Using the LSDS approach, for each independent slot, a sensor node only transmits/receives packets when a slot is good and delays transmission/reception when a slot is bad.

An energy efficient transmission strategy for WSNs that operate in a strict energy-constrained environment is discussed in [15]. The binary-decision based transmission in [15] makes a decision on whether to transmit or not based on the current channel conditions. Another component discussed in [15] is a channel-aware backoff adjustment, which favors sensor nodes with better channel quality, when deciding and prioritizing the transmission.

Unlike [11], [12] and [13], our proposed protocol leverages the channel quality information from the PHY layer for scheduling, which helps to reduce retransmissions and contribute to energy saving. The protocol proposed in [14] requires neighbor discovery and schedule exchange between all sensor nodes in the network, which can increase the overhead involved in communication. In [11], the authors assume that each node can overhear its neighbors and share state information to acquire time slots. However, in our approach a member node does not need to share the state information with its neighbors. Instead, it makes a selfish decision on the number of time slots to request from the cluster head and the cluster head optimally allocates time slots to each node using the channel state information. By considering a two-step approach in the intra-cluster scheduling we minimize the overhead involved in communication. Thus, considering channel quality and adopting the two-step approach in the intra-cluster scheduling helps to elevate the energy-saving and throughput by minimizing the overhead and redundant retransmissions.

Unlike [11], we do not assume that the nodes are synchronized. Instead, the cluster head synchronizes the member nodes within a cluster at the start of every cycle, by inserting a time stamp in the header of a broadcasted slot announcement (SA) packet. This way we avoid the use of separate time synchronization packets. Our protocol differs from [12] and [13], in that we consider the residual energy level and buffer occupancy of each node to reduce the energy consumption while maintaining reliability.

The main contribution of this paper is the design and testbed implementation of a novel channel-dependent optimized scheduling scheme within a cluster. The objective of the proposed scheduling scheme is to conserve energy by avoiding unnecessary retransmissions and at the same time improving the throughput, while maintaining reliability, by considering the channel quality of each link and minimizing buffer overflows.

3. SYSTEM ARCHITECTURE

We consider an UW-ASN, in which a set of nodes are deployed on the ocean bed and a sink node is located on the surface. We consider a static clustered topology, where clusters are already formed and we assume there is no inter-cluster interference, which can be obtained by using different carriers or orthogonal spreading codes. The member nodes communicate with the cluster head using lower transmit power levels as they are within the transmission range from the cluster head. Cluster heads monitor and coordinate the transmission of their member nodes. Since the cluster head handles several functionalities, its energy consumption is higher than the rest of the member nodes, and thus it is more likely to fail due to its non-uniform energy consumption. Accordingly, it is necessary to rotate the role of cluster heads based on their residual energy level. Cluster formation and cluster head rotation are not the focus of this work.

The clustered topology, shown in Fig. 1, depicts a sink node S and two clusters with their cluster heads denoted by CH_1 and CH_2, respectively. The member nodes are one-hop away from the cluster head and within the communication range of their respective cluster heads. The cluster heads communicate directly with the sink node. We propose a hybrid scheme for the intra-cluster and cluster head-sink communication, in which TDMA scheduling is used for intra-cluster communication and a CSMA/CA scheme for communication between the cluster head and the sink node. There are two phases, namely, (i) intra-cluster phase and (ii) cluster head-sink communication phase, which are discussed in Section 4. The intra-cluster communication phase is subdivided into several cycles. Each cycle is further divided into time slots, where each member node will transmit data packets to their cluster head in its allocated time slots. The timing diagram, shown in Fig. 2, demonstrates the exchange of control and data packets in one intra-cluster cycle.

The time slot scheduling within the cluster is based on the channel quality, node priority, buffer occupancy and the residual energy level of each node. The cluster head has to collect information about buffer occupancy and residual energy level from its member nodes in order to efficiently allocate time slots for each member node. This may lead to large overhead, due to the exchange of this information

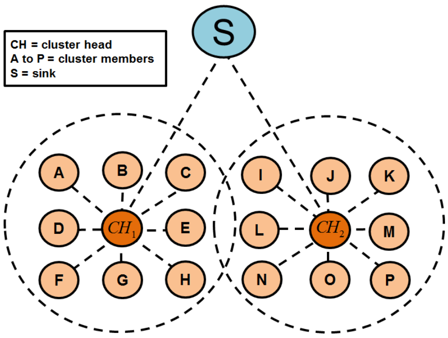

Figure 1: Cluster topology.

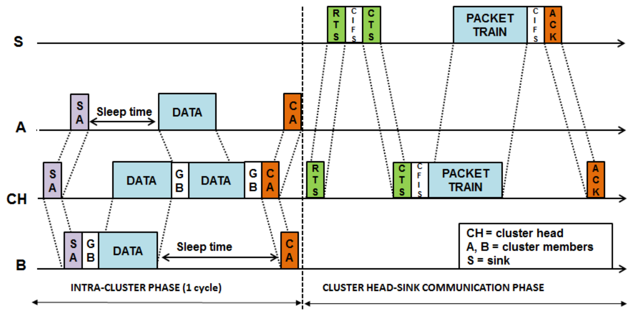

Figure 2: Timing diagram.

within the cluster. Accordingly, we let each member node ask for the required time slots, which are determined based on a selfish decision, by taking into account their own buffer occupancy and residual energy levels. Therefore, in this case the cluster head needs to know the time slots requested by each member node only. The time slots are chosen from an optimal slot policy matrix at the start of the next cycle, which is formulated and obtained using an MDP, details of which are discussed in Section 5. Once the member nodes have informed the cluster head as to the number of time slots they need for the next cycle, the cluster head will allocate the time slots in an optimal way based on the received slot requests from all member nodes and their channel conditions (SNRs). The intra-cluster scheduling functionalities are explained in Section 5.

4. PROTOCOL DESCRIPTION

4.1 Intra-cluster Communication Phase

In the intra-cluster communication phase, a member node transmits the collected data to the cluster head during its allotted time slot, while the other member nodes sleep to save energy. During this phase, the member nodes and the cluster head use their lowest transmit power level for communication within the cluster. The member nodes append the number of time slots they need for the next cycle as a trailer to the data packet sent in the current cycle. The cluster head learns about the time slot requirements of each member node and allocates the available time slots optimally using the proposed optimization algorithm discussed in Section 5. After computing the number of time slots for the member nodes, the allotted time slots are included in the SA packet along with the member node's *IDs*.

From the timestamp, position and allotted time slots in the SA packet each member node will compute the start time for transmission of its data packets. Several cycles of intra-cluster communication phase can take place before transitioning to the cluster head-sink communication phase. The cluster head stores the data packets received from its member nodes over several cycles in the intra-cluster phase to be sent as a long train of packets to the sink node.

Figure 2 shows the communication between the cluster head *CH* and the member nodes *A* and *B* for one intra-cluster cycle followed by the cluster head-sink communication phase. The member nodes (*A* and *B*) join their respective clusters by transmitting a small packet, JOIN request message, to the *CH*. The JOIN request message contains the number of time slots requested from each node. The cluster formation phase occurs prior to the intra-cluster phase.

The cluster head *CH* performs channel dependent optimized intra-cluster scheduling to allocate the time slots to nodes *A* and *B*, and broadcasts the slot information to the member nodes through an SA packet. Upon reception of the SA packet, the member nodes are informed on when to transmit. The guard-band, GB, accounts for additional delays e.g., multipath. In our example, node *B* learns from the SA packet that it has the opportunity to transmit during a particular time slot. While node *B* is transmitting, node *A* sleeps to save energy and wakes up only when needed. Similarly, node *B* goes into the sleep mode when A is transmitting. After a successful reception of the data packets, cluster head *CH* broadcasts a cumulative acknowledgment (CA) packet to acknowledge the reception of data packets. We avoid the use of individual acknowledgement of each data packet to minimize the control overhead by acknowledging the member nodes within a cluster with a single CA packet, as discussed in [3]. The received data packets are then gathered at the *CH*. Packets not received correctly will be retransmitted in the next cycle. We name this sequence of data exchanges between the cluster head and member nodes as the *intra-cluster communication phase*. Since the transmission within each cluster is time-slotted, the member nodes need to be synchronized. We assume that the cluster head knows the location of its member nodes, and hence, can compute the propagation delay with respect to its member nodes, which is used for synchronization. In Section 5, we explain the channel dependent optimized intra-cluster scheduling.

4.2 Cluster Head-Sink Communication Phase

During the cluster head-sink communication phase, the cluster head transmits the packet train to the sink node using CSMA/CA scheme. In this phase, the cluster heads use a higher transmit power level in order to reach the sink. Accordingly, in Fig. 2, the cluster head *CH* transmits an RTS to the sink *S* and waits for a CTS. The CTS packet contains information about the period during which the *CH* intends to keep *S* node occupied. Hence, when the other cluster heads overhear CTS, they will learn that *S* is busy and will defer their transmissions. Upon reception of the CTS, the *CH* will transmit it's packet train to *S*, which will be acknowledged with an ACK packet. The RTS/CTS-packet train and the ACK exchange between the cluster head and sink node is named as the *cluster head-sink communication phase*.

5. INTRA-CLUSTER SCHEDULING

In the intra-cluster scheduling, a TDMA scheme is adopted to minimize the energy consumption by avoiding contention and packet collisions. Additionally, member nodes utilize their state information, such as buffer occupancy and residual energy level of its battery, to compute the number of time slots they will need in the next cycle. One way to address the time slot allocation problem is to transmit the member node's state information to the cluster head and let it compute a global optimal solution. However, by doing so, the packet overhead and optimization problem complexity will increase exponentially, resulting in high computational delay and memory requirements. Hence, we let each member node use the MDP, discussed later in this section, to make a selfish decision on the number of time slots they need during the next cycle. To further improve the reliability of our scheme, the cluster head takes the ultimate time slot allocation decision, which helps to minimize the number of corrupted packets due to poor channel conditions. The intra-cluster scheduling involves the following two steps:

(i) **MDP based time slots request decision at the member node**: The member nodes request from the cluster head the number of time slots it needs during the next cycle. The buffer occupancy and residual energy states change every cycle as shown in Figs. 3 and 4.

The time slots are requested such that the energy consumption is minimized, while at the same time maintaining reliability. Time slots are requested at the start of every cycle in the intra-cluster phase. A stochastic optimizer formulates the problem as an infinite horizon MDP, which considers the buffer occupancy and residual energy level of the member node as the input states. The MDP selects an optimal action (number of time slots required) with the objective of minimizing the energy consumption at each member node, while maintaining reliability by preventing buffer overflow. Every member node performs this operation to obtain the required number of time slots to request from the cluster head.

(ii) **Channel dependent time slot allocation at the cluster head**: Based on the obtained requests and the quality of link to the member node, the cluster head will optimally allocate time slots to its member nodes. We name the process of optimal allocation of time slots based on the received time slot requests and channel quality as *channel-dependent optimized scheduling.*

5.1 Markov Decision Process Formulation

System state space: An optimal policy is a function mapping node state information (residual energy level and buffer occupancy) to the number of time slots required to transmit. Consider packet transmissions from the member node to cluster head over the UW-A channel. Assume that each cycle, in the intra-cluster phase, is divided into equally sized time slots. We assume that all packets transmitted are successfully received at the cluster head. At each cycle, the joint state of the node is defined by the buffer occupancy and residual energy level.

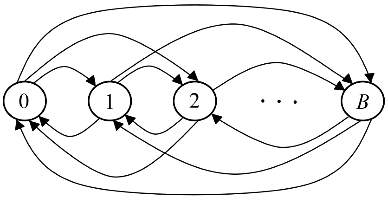

Figure 3: Buffer state of MDP.

The transmission buffer is modeled as a first-in-first-out queue and can hold a maximum of B packets. We define the buffer state to be the buffer occupancy, and denote $b \in \mathcal{B} = \{0, 1, 2, \cdots B\}$ as the buffer state space. The buffer state evolves recursively as follows

$$b^{n+1} = \min(B, b^n + k^n - \max(0, a^n)) , \qquad (1)$$

where b^n is the buffer state in the n^{th} cycle, the initial buffer state is denoted by b^0, and b^{n+1} is the buffer state in the $(n+1)^{th}$cycle, which depends on the action a^n taken in the n^{th} cycle. We define an action as the number of time slots requested by the member node, and represent $a \in \mathcal{A} = \{a_{min}, \cdots, 0, \cdots, a_{max}\}$ as the action state space, where $a_{min} = -x$ is the minimum value of action, which implies that the member node prefers to sleep for x cycles to save energy, and $a_{max} = B$ denotes the maximum value of an action, which is equal to the buffer capacity B.

We assume that during one time slot only one packet can be transmitted. Therefore, the number of time slots requested is equivalent to the number of packets ejected from the buffer. At each cycle, the sensor node injects k^n packets, each of size K bits. The arrival process $\{k^n : n = 0, 1, \cdots\}$ is modelled as a Poisson distribution, $p_k(k) \sim Poi(\lambda)$, where λ is the mean of the number of packet arrivals. Based on the buffer recursion in (1), and the arrival distribution, the sequence of buffer states can be modeled as a controlled Markov chain with transition probability given by $p_b(b'|b, a)$, which is the probability that a node in state b transitions to state b' by taking an action a. We introduce a buffer cost $\beta(b)$, which is expressed in (2) as the expected sum of holding cost and overflow cost to penalize the member node for queuing delays and buffer overflows, thereby avoiding packet losses, due to the buffer overflow,

$$\beta(b) = \sum_k p^k(k) \{\underbrace{(b - \max(0, a))}_{holding\ cost} + \underbrace{+\, \eta \max(0, (b - \max(0, a) + k - B))}_{overflow\ cost}\}. \qquad (2)$$

The holding cost is associated with the number of packets that were not transmitted at the beginning of a cycle and remain in the buffer to be transmitted at the next cycle. The overflow cost represents the number of packets that were dropped from the buffer, due to packets leaving the buffer slower than the arrival rate. Overflow cost imposes a penalty of η on each dropped packet. The selection criterion for η is discussed at the end of this section.

Each node initially starts with a full battery capacity of E, we denote $e \in \mathcal{E} = \{0, 1, 2, \cdots E\}$ as the residual energy state space. At each cycle of the intra-cluster phase, the total energy consumed is given by

$$\sigma(a) = \begin{cases} P_{low}(t_{max} - t_a) + P_{tx}t_a + E_{tr}, & \forall a > 0 \\ P_{low}t_{max}, & \forall a \leq 0 \end{cases} \qquad (3)$$

where P_{low} and P_{tx} are the power consumed in the low power state of the sensor node and the power consumed to transmit one packet, respectively, in units of (Watt), E_{tr} is the energy consumed in transitioning between the low power and active states in units of (Joule), t_{max} and t_a are the duration of one intra-cluster cycle, and the duration of a number of time slots, respectively in seconds. The energy state recursion is given by

$$e^{n+1} = \max\{0, e^n - P_{low}(t_{max} - \max(0, t_a)) + + P_{tx} \max(0, t_a) + E_{tr} \min(1, \max(0, a))\}, \qquad (4)$$

where e^n and e^{n+1} are the current and next residual energy states, respectively with initial residual energy state denoted by e^0. Energy states can be modeled as a Markov chain with transition probability denoted by $p_e(e'|e, a)$, which is the probability that the member node in residual energy state e transitions to e' by taking an action a. To penalize the node that runs out of energy we define a failing-penalty as

$$\tau(e) = 1, \quad for\ \ e = 0, \qquad (5)$$

in which a member node that runs out of energy gets penalized by $\tau(e)$. We also give a sleep reward for a node that prefers to sleep for an a number of cycles expressed by

$$\mu(a) = \min(0, a), \quad \forall\ a < 0, \qquad (6)$$

in which the member node is rewarded by $\mu(a)$.

We define the joint state of the node as a vector comprised of the buffer state b and the residual energy state e, as $s \triangleq (b, e) \in \mathcal{S}$. The action is defined as the number of time slots required to transmit in the next cycle, where $a \in \mathcal{A}$. The

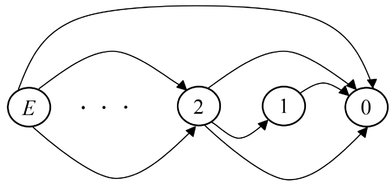

Figure 4: Energy state MDP.

sequence of states $\{s^n : n = 0, 1, 2, \cdots\}$ can be modeled as a controlled Markov chain with transition probabilities that can be determined from the conditionally independent buffer state and energy state as follows

$$p_s\left(s'|s,a\right) = p_b\left(b'|b,a\right) p_e\left(e'|e,a\right), \tag{7}$$

where $p_s\left(s'|s,a\right)$ denotes the probability that a node in joint state s will transition to joint state s' by taking an action a. The objective of the MDP is to find an optimal time slot policy that minimizes the overall cost. Let $\pi : \mathcal{S} \to \mathcal{A}$ denote the time slot policy mapping states to actions such that $a = \pi\left(s\right)$ and $\mathbf{P}\left(b, e\right)$ is the optimal slot policy matrix with buffer state b denoted by the column index and energy state e denoted by the row index, which represents the possible number of time slots a node will need at a given buffer and energy state. The cost function, based on four factors: (i) power cost $\sigma\left(a\right)$, (ii) buffer cost $\beta\left(b\right)$, (iii) failing cost $\tau\left(e\right)$, and (iv) sleep reward $\mu\left(a\right)$, is expressed as

$$C\left(s,a\right) = \sigma\left(a\right) + \lambda_1\beta\left(b\right) + \lambda_2\tau\left(e\right) + \lambda_3\mu\left(a\right), \tag{8}$$

where λ_1, λ_2 and λ_3 are the weights over these factors. The optimal slot policy is obtained by solving the MDP using the value-iteration algorithm, discussed in (9) and (10). The optimal state value function quantifies how good it is to be in the joint state $s = (b, e)$. The optimal state value function and optimal slot policy are expressed as

$$V_\pi^*\left(s\right) = \min_{a\in\mathcal{A}} \left\{ C\left(s,a\right) + \gamma \sum_{s'\in\mathcal{S}} p_s\left(s'|s,a\right) V_\pi^*\left(s'\right) \right\}, \tag{9}$$

$$\pi^*(s) = \arg\min_{a\in\mathcal{A}} \left\{ C\left(s,a\right) + \gamma \sum_{s'\in\mathcal{S}} p_s\left(s'|s,a\right) V_\pi^*\left(s'\right) \right\}, \tag{10}$$

where $\gamma \in (0, 1]$ is a discount factor. The penalty factor η is defined as

$$\eta = \frac{\gamma}{1-\gamma}. \tag{11}$$

The penalty of dropping a packet must be greater or equal the cost of admitting it into the buffer, as discussed in [16]. Performing MDP real-time with dynamically changing network conditions will increase the energy consumption and delay, due to the high computational and/or memory complexity. Thus, MDP is performed offline and the obtained optimal slot policy matrix $\mathbf{P}\left(b, e\right)$, for all possible buffer b^n and energy e^n states, respectively, are stored in a table inside each node. A member node selects the appropriate time slots required for the next cycle from the optimal slot policy matrix $\mathbf{P}\left(b, e\right)$ according to its current buffer and energy state.

5.2 Optimal time slot allocation problem

After the cluster head obtains the number of time slots requested by the member nodes, it allocates the available time slots with an objective to maximize the overall throughput while maintaining a reliable transmission. To maximize the overall throughput it considers the channel quality in terms of the SNR. On the other hand, to maintain a reliable transmission it minimizes the packet loss due to the buffer overflows. The allocated time slots should not exceed the maximum available time slots in one cycle.

The time slot allocation problem can be formulated as an optimization problem as follows

$$\mathcal{P}_1 : \underset{\mathbf{t}_i}{\text{maximize}} \sum_{i=1}^{N} \left(1 - PER_i\right) t_i \tag{12}$$

$$\text{subject to:} \sum_{i=1}^{N} t_i \le T_{max} \tag{13}$$

$$0 \le t_i \le a_i, \quad \forall\, i \in [1, N], \tag{14}$$

where $(1 - PER_i)$ is the probability of successful packet reception from the i^{th} node, where PER_i is the packet error rate of the received packets from the i^{th} node, t_i is number of time slots allocated to the i^{th} node, T_{max} is the maximum number of time slots available per cycle and a_i is the number of time slots requested by the i^{th} node.

For different applications (e.g. tsunami detection, environmental monitoring or undersea explorations) each node might have different priority levels for handling the data to be transmitted. We denote the nodes with higher priority data (eg. tsunami detection messages) as primary nodes while less urgent data transmissions (eg. oceanographic data collection) as secondary.

Accordingly, we reformulate $\mathcal{P}_1$ to take that into consideration by introducing an additional node priority constraint as follows

$$\mathcal{P}_2 : \underset{\mathbf{t}_i, \mathbf{t}_j}{\text{maximize}} \sum_{i=1}^{N} \left(1 - PER_i\right) t_i + \sum_{j=1}^{M} \left(1 - PER_j\right) t_j \tag{15}$$

$$\text{subject to:} \sum_{i=1}^{N} t_i + \sum_{j=1}^{M} t_j \le T_{max} \tag{16}$$

$$0 \le t_i \le a_i, \quad \forall\, i \in [1, N], \tag{17}$$

$$0 \le t_j \le a_j, \quad \forall\, j \in [1, M], \tag{18}$$

$$\sum_{j=1}^{N} t_j = \min\left(\sum_{j=1}^{N} a_j \ , \ T_{max} \right), \tag{19}$$

where a_j is the number of time slots requested by the primary (higher priority) node j and t_j is the number of time slots allotted to the primary node j. In this case, the index i represents the secondary nodes.

In the optimization problem $\mathcal{P}_2$, with the additional priority constraint given by (19), the cluster head takes into consideration the priority of each member node while allocating time slots. The nodes with higher priority are served first and the remaining time slots are distributed among the lower priority member nodes.

By considering the PER_i during scheduling we take into account SNR_i, the SNR of the received packet of node i. The number of time slots allocated for the primary and secondary users is assigned such that it does not exceed the maximum number of time slots available per cycle, T_{max}, which is considered by (16). Constraints (17) and (18) make sure that it does not allocate more than what is asked.

One drawback in $\mathcal{P}_1$ and $\mathcal{P}_2$ formulations is that they consider only the nodes with higher SNR (lower PER) values without considering their urgency. Thus, the nodes with a relatively bad channel condition may never have a chance to transmit, which will eventually cause their buffers to overflow, resulting in packet loss. Accordingly, we modify the optimization problem by introducing an urgency factor θ_i, which takes into account the urgency of the member nodes.

Table 1: Simulation Parameters: Network lifetime

Simulation parameters	Values
Number of member nodes per cluster	9
Maximum buffer size of each node	40 packets
Number of intra-cluster cycles	6500
SNR range	1 dB - 15 dB
Mean packet per node per cycle	1
Transmission energy per packet	1.78 W
Low power listening energy	16.8 mW
Wake up energy	0.2 W

The modified optimization problem is formulated as

$$\mathcal{P}_3\colon \underset{\mathbf{t}_i}{\text{maximize}} \sum_{i=1}^{N} \left[(1-\theta_i)\left(1-PER_i\right)+\theta_i\right] t_i \tag{20}$$

$$\text{subject to}\colon \sum_{i=1}^{N} t_i \leq T_{max} \tag{21}$$

$$0 \leq t_i \leq a_i, \quad \forall\, i \in [1, N], \tag{22}$$

where $\theta_i = \frac{a_i}{a_{max}+\epsilon}$ is the urgency factor. In the case of high traffic, a member node may request up to a_{max} time slots, the size of its buffer, which may cause the weight on the SNR_i term to vanish in the absence of ϵ. Hence, ϵ is introduced and is set to a very small value ≈ 0.0001. A high value of θ_i will give smaller weight to $(1 - PER_i)$ and as a result, the urgency of the node will be taken into account, thereby preventing buffer overflow. Accordingly, the optimization problem $\mathcal{P}_3$ allocates time slots based on the received signal quality (SNR) and urgency factor with the objective to maximize the throughput and at the same time prevent buffer overflows.

6. SIMULATION RESULTS

An intra-cluster scenario comprising of 9 member nodes is simulated in a custom-designed network simulator to analyze the network lifetime. The parameters used for simulation are presented in Table 1. We compare our proposed scheme employing MDP with optimized time slot allocation; (i) with an urgency factor, (ii) without an urgency factor and (iii) a scheme that neither uses MDP nor an optimization at the cluster head. In (iii), each member node informs the cluster head their current buffer occupancy. The cluster head collects the buffer occupancy information from all the member nodes in its cluster and allocates time slots as $t_i = \frac{b_i T_{max}}{\sum_{j=1}^{N} b_j}$, where b_i and b_j correspond to the buffer occupancy of the i^{th} and j^{th} nodes, respectively. Which implies that member nodes with higher buffer occupancy will receive proportionally higher share of the total time slots available during an intra-cluster cycle.

Figure 5 illustrates the network lifetime by measuring the number of failed member nodes in the cluster as the number of intra-cluster cycles increases. In Fig. 5, we can observe that the scheme with optimization and without urgency factor has the highest network lifetime, as the number of intra-cluster cycles elapse before the first member node fails is much higher compared to the other two schemes. This is due to the fact that time slots are allotted to member nodes with a better channel quality, thus reducing the number of packets lost. As a result, it reduces the number of retransmissions, which leads to a reduction in the energy consumption. The scheme that neither uses MDP nor optimization performs worse than the one with optimization and urgency factor. This is because it takes into account the urgency of the slot requirement along with channel quality of the member node. As a result, time slots may be allocated to members with slightly poorer channel quality, but with higher urgency factor.

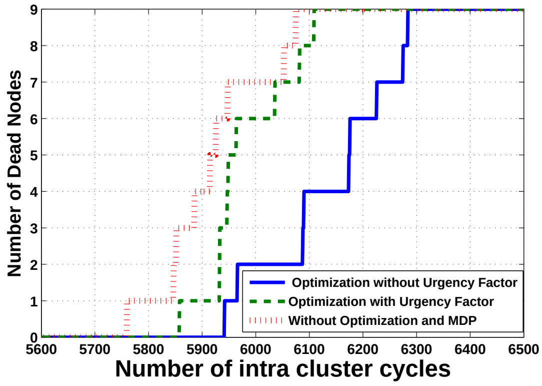

Figure 5: Network lifetime.

Table 2: Simulation Parameters: For varying traffic

Simulation parameters	Values
Number of member nodes per cluster	9
Maximum buffer size of each node	40 packets
Number of intra-cluster cycles	500
SNR range	1 dB - 20 dB 1 dB - 15 dB
Range of mean packet per node per cycle	1 - 5
Transmission energy per packet	1.78 W
Low power listening energy	16.8 mW
Wake up energy	0.2 W

The next set of simulations represent the network behavior in varying traffic conditions. The set of simulation parameters are shown in Table 2. All simulations are carried out with two different channel conditions; relatively good channel condition with SNR range varying uniformly between 1 dB to 20 dB, and relatively poor channel condition with SNR varying uniformly between 1 dB to 15 dB. The traffic is represented as the average number of packets produced by each member node per cycle and is varied from 1 to 5.

Figures 7 and 6 represent the throughput for relatively poor channel conditions and good channel conditions, respectively. It is evident that the schemes with optimization have higher throughput as compared to scheme that neither uses optimization nor MDP. At higher traffic level, as the buffer is almost always likely to be full, all nodes may be requesting a large number of time slots. Thus, the scheme with optimization and urgency factor not only reduces the buffer overflow, but also favors the nodes with better channel condition, due to the factor ϵ, as discussed in Section 5. Thus, we can see from Fig. 7, which describes a relatively poor channel conditions, the scheme with optimization and urgency factor has the highest throughput at higher network traffic.

We define reliability as the total number of packets successfully received at the cluster head divided by the total number of packets generated. Figures 8 and 9 show the reliability of the network for the three different schemes. We can see that the schemes with optimization perform better in terms of reliability compared to the scheme without optimization and MDP, because the packet loss rate of the optimized schemes are lower, as they favor nodes with better channel quality. The optimized scheme with urgency factor reduces the buffer overflow as the traffic increases, as shown in Figs. 10 and 11, which describe the number of packets dropped due to the buffer overflow. Since the number of packets dropped at the buffer is lowest for the optimized scheme with urgency factor it has the highest reliability among the three schemes.

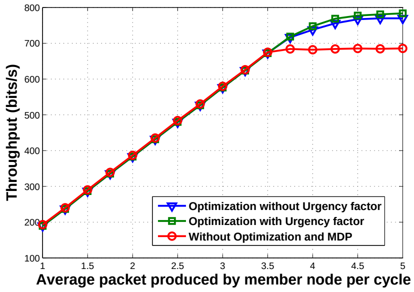

Figure 6: Throughput (SNR range: 1dB-20dB).

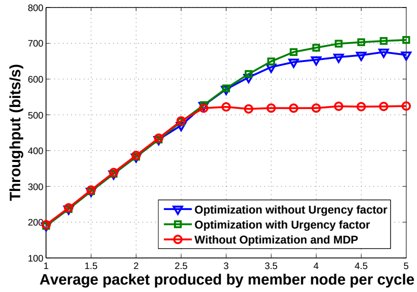

Figure 7: Throughput (SNR range: 1dB-15dB).

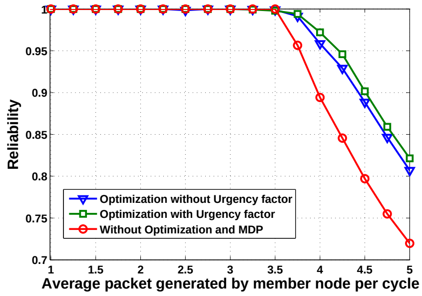

Figure 8: Reliability (SNR range: 1dB-20dB).

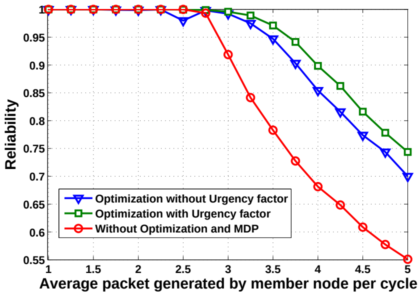

Figure 9: Reliability (SNR range: 1dB-15dB).

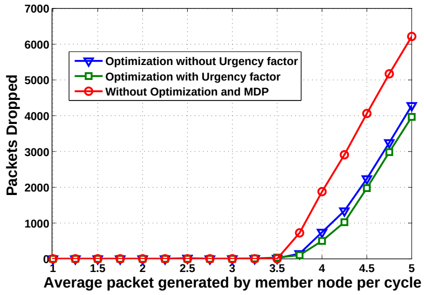

Figure 10: Packets dropped (SNR range: 1dB-20dB).

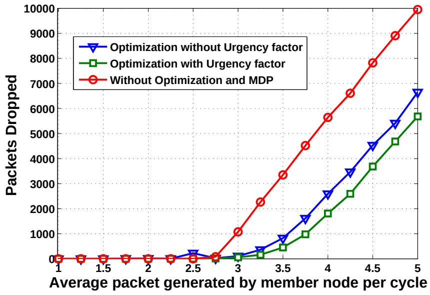

Figure 11: Packets dropped (SNR range: 1dB-15dB).

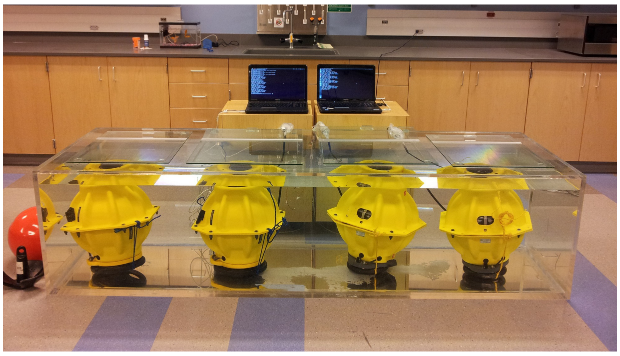

Figure 12: Experimental setup.

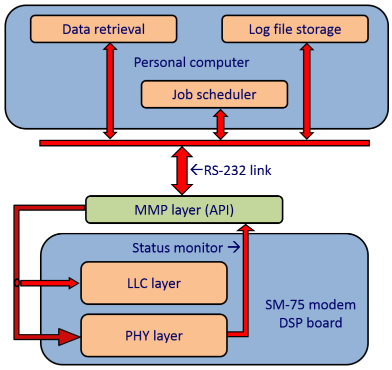

Figure 13: Communication architecture.

7. TESTBED EVALUATION

Experiments were conducted in a water tank of dimensions 8 ft × 2.5 ft × 2 ft as shown in Fig. 12 to evaluate the proposed intra-cluster scheduling scheme. The testbed consists of four SM-75 SMART acoustic modems by Teledyne Benthos [5]. A bit rate of 800 bits/s with multiple frequency shift keying (MFSK) modulation scheme was used to conduct the experiments. The modems were connected through an RS-232 serial port and controlled by a PC with modem management protocol (MMP), as shown in Fig. 13. MMP is a binary interface for machine-based command and control of the modem. Using the MMP, the PC can control the lower layers of the modem's protocol stack. The MMP serves as an application programming interface (API) that interfaces the PC to the modem. MMP layer abstracts all the main PHY and logical link control layer (LLC) functionalities of the modem to the PC [17]. It also offers access to status monitoring signals that carries information about SNR, round-trip time (RTT) among others. The logic in control of the proposed scheduling is implemented in C++ language. The node, second from the left corner of tank shown in Fig. 12, is assigned as the cluster head CH, while the other three nodes are the member nodes. The intra-cluster scheduling scheme with urgency factor was experimented and the obtained results are compared with the scheduling scheme, which neither uses the MDP nor the proposed optimization scheme. The experiments were conducted to analyze; (i) the throughput and (ii) the reliability, under varying traffic conditions.

Table 3: Experimental Parameters

Experimental parameters	Values
Number of member nodes per cluster	3
Maximum buffer size per node	40 packets
Number of intra-cluster cycles	20
Range of mean packet per node per cycle	1 - 5

Initially, all member nodes were allocated one time slot each. A total of 20 cycles of the intra-cluster communication phase were conducted. In every cycle, the sequence of data and control packets exchange took place as described in Fig. 2. The parameters used in the experiment are shown in Table 3. The modems were set to transmit at their lowest transmit power level (1.78 W).

We compare the proposed scheduling scheme employing MDP with optimized time slot allocation; (i) with an urgency factor, and (ii) the scheme that uses neither MDP nor an optimization at the cluster head. In (ii), each mem-

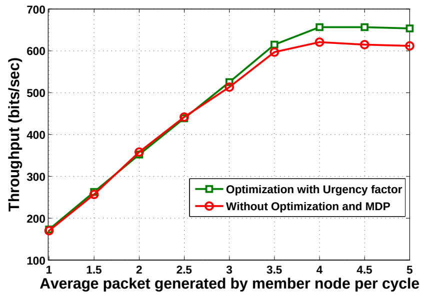

Figure 14: Throughput versus traffic load (testbed evaluation).

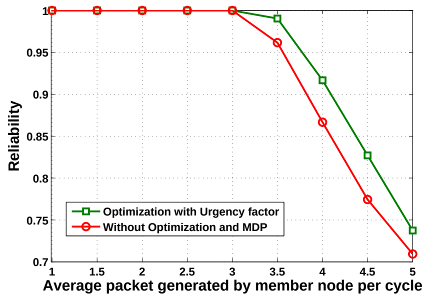

Figure 15: Reliability versus traffic load (testbed evaluation).

ber node informs the cluster head about their current buffer occupancy. The cluster head collects the buffer occupancy information from all the member nodes in it's cluster and allocates time slots proportionally among them as described in Section 6.

Figure 14 illustrates the throughput in [bits/s], which is defined as the number of data packets successfully delivered at the cluster head from the member nodes per unit time. It is evident, from Fig. 14, the scheme with optimization and urgency factor improves the throughput by up to 10%, because it takes into account the urgency of the time slot requirement along with channel quality of the member nodes. At higher traffic level, as the buffer level will almost always be full, all member nodes may be requesting large number of time slots. Thus, the scheme with optimization and urgency factor not only reduces the buffer overflow, but also favors the member nodes with better channel condition, due to the ϵ factor as discussed in Section 5. Thus, we can see from Fig. 14 that the scheme with optimization and urgency factor has the highest throughput at higher traffic load.

We also compute the reliability following the discussions in Section 6. Figure 15 shows the reliability of the network for the two different schemes. It is evident that the scheme with optimization and urgency factor performs better in terms of reliability, compared to the scheme without optimization and MDP, because the loss rate of the optimized scheme is lower, as it favors the member nodes with better channel quality.

8. CONCLUSIONS

In this work, we have proposed a channel-dependent optimized intra-cluster scheduling, which takes into consideration the residual energy level, buffer occupancy, urgency factor of the member nodes and the SNR of each link. From the simulation results we can see that the optimized scheme without urgency factor provides better network lifetime with an increase in packet drop rate at high traffic. The performance of optimized scheme with urgency factor lies in between the other two schemes in terms of network lifetime, however less packets are dropped from the buffer. Optimized scheme with urgency factor performs well at higher traffic loads. The schemes with optimization shows significant improvement in throughput, reliability and network lifetime in contrast to the scheme without optimization and MDP. Simulations demonstrate that the proposed scheduling scheme with urgency factor achieves a 28% improvement in throughput and improves reliability by up to 25% as compared to the scheduling scheme which does not use MDP and optimization. The testbed evaluations substantiate an improvement in throughput by up to 10% and also improves reliability.

9. REFERENCES

[1] T. Melodia, H. Kulhandjian, L. Kuo, and E. Demirors. *Advances in Underwater Acoustic Networking.* In Mobile Ad Hoc Networking: Cutting Edge Directions, Eds. S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, John Wiley and Sons, Inc., Hoboken, NJ, second edition, 2013.

[2] L. Kuo and T. Melodia. Distributed Medium Access Control Strategies for MIMO Underwater Acoustic Networking. *IEEE Trans. Wireless Communications*, 10(8):2523–2533, August 2011.

[3] H. Kulhandjian, T. Melodia, and D. Koutsonikolas. CDMA-based Analog Network Coding through Interference Cancellation for Underwater Acoustic Sensor Networks. In *Proc. of ACM Intl. Conf. on UnderWater Networks and Systems (WUWNet)*, Los Angeles, CA, USA, Nov. 2012.

[4] D. Pompili, T. Melodia, and I. F. Akyildiz. A CDMA-based Medium Access Control Protocol for Underwater Acoustic Sensor Networks. *IEEE Transactions on Wireless Communications*, 8(4):1899–1909, April 2009.

[5] Teledyne-Benthos, Acoustic Modems. [Online]. Available: http://www.benthos.com.

[6] The Sensor Network Museum. [Online]. Available: http://www.snm.ethz.ch/snmwiki/Projects/MicaZ.

[7] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, May 2005.

[8] M. Khan, B. Bhargava, S. Agarwal, and L. Lilien. Self-configuring node clusters, data aggregation, and security in microsensor networks. Technical report, Purdue University, West Lafayette, Indiana, USA, March 2003.

[9] B. Peleato and M. Stojanovic. A mac protocol for ad-hoc underwater acoustic sensor networks. In *Proc. of ACM Intl. Workshop on UnderWater Networks (WUWNet)*, pages 113–115, Los Angeles, CA, USA, 2006.

[10] P. Karn. MACA – a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, September 1990.

[11] K. B. Kredo and P. Mohapatra. A hybrid medium access control protocol for underwater wireless networks. In *Proc. of ACM Intl. Workshop on UnderWater Networks (WUWNet)*, pages 33–40, Montreal, Quebec, Canada, 2007.

[12] M. K. Park and V. Rodoplu. UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks. *IEEE Journal of Oceanic Engineering*, 32(3):710–720, 2007.

[13] 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 IEEE Intl. Conf. on Computer Communications (INFOCOM)*, pages 231–235, Pheonix, AZ, USA, April 2009.

[14] V. Phua, A. Datta, and R.C.Oliver. A TDMA-Based MAC Protocol for Industrial Wireless Sensor Network Applications using Link State Dependent Scheduling. In *Proc. of IEEE Global Telecom. Conf. (GLOBECOM)*, San Francisco, CA, USA, November 2006.

[15] C. V. Phan, Y. Park, H. H. Choi, J. Cho, and J. G. Kim. An energy-efficient transmission strategy for wireless sensor networks. *IEEE Transactions on Consumer Electronics*, 56(2):597–605, May 2010.

[16] N. Mastronarde and M. van der Schaar. Fast reinforcement learning for energy-efficient wireless communication. *IEEE Trans. Sig. Proc.*, 59(12):6262–6266, December 2011.

[17] H. Kulhandjian, L. Kuo, T. Melodia, D. A. Pados, and D. Green. Towards Experimental Evaluation of Software-Defined Underwater Networked Systems. In *Proc. of IEEE UComms*, pages 1–9, Sestri Levante, Italy, September 2012.