Distributed Scheduling and Routing in Underwater Wireless Networks

Kurtis Kredo II
Electrical and Computer Engineering Dept.
California State University, Chico
kkredo@csuchico.edu

Prasant Mohapatra
Computer Science Dept.
University of California, Davis
pmohapatra@ucdavis.edu

***Abstract*—Several underwater network characteristics, including long propagation delays and a bandwidth dependent on distance, provide unique challenges to protocol designers. In this paper we present STUMP-WR, a distributed routing and channel scheduling protocol, designed for heavily loaded underwater networks. STUMP-WR selects and schedules links using a distributed algorithm to overlap communications by leveraging the long propagation delays. Through simulation we show that STUMP-WR outperforms previously proposed channel access protocols for underwater networks by achieving higher throughput and lower energy consumption per bit delivered at nearly all traffic rates. We also determine the effect of using CDMA on protocol operation.**

I. Introduction

Underwater networks offer new and unique research opportunities and promise great potential for underwater exploration, resource monitoring, and military applications. Their defining characteristic is the use of acoustic communication due to the limited range of RF signals in the underwater environment. Acoustic communication results in a slow propagation speed, approximately 1500 m/s, a bandwidth dependent on node distance, and data rates on the order of kilobits per second. Fixed energy resources place further limits on node capabilities. Addressing these characteristics while providing useful performance has driven much recent research.

Proposed MAC protocols for underwater networks have included modified RTS/CTS schemes [1], [2], adapted Aloha [3], [4], and loosely scheduled mechanisms [5], [6]. The disadvantage of these protocols is the overhead caused by long propagation delays and collisions, particularly in heavily loaded networks. CDMA techniques [7] reduce contention and have other benefits for underwater networks, but require code assignment and power control algorithms and reduce the already limited data rates. Scheduled protocols [8], [9], like the work presented here, provide better performance under heavy loads, but require time synchronization and scheduling algorithm overheads.

The unique capabilities of underwater devices have also driven research in network protocols, such as adaptations of AODV [8] and geographic routing [10]. These protocols require unnecessary overhead for the non-mobile networks under consideration here, but their techniques could easily be adapted for use in other network scenarios.

The closest related work comes from ST-MAC [9], which determines link schedules based on a spatio-temporal conflict graph and a centralized, modified graph coloring algorithm. ST-MAC's centralized approach would require significant overhead to collect network information and distribute transmission schedules.

In this work we expand upon our previous Staggered TDMA Underwater MAC Protocol (STUMP) [11] by proposing and evaluating through simulation a combined routing and scheduling protocol, STUMP With Routing (STUMP-WR), which utilizes a simple distributed algorithm to select and schedule links for transmission while preventing collisions. STUMP-WR operates on links, unlike the rings used in STUMP, as this makes it easier to adapt other link selection algorithms to STUMP-WR. Our contributions include: a distributed, combined routing and scheduling algorithm for underwater networks; comparison with previously proposed protocols through simulation showing that STUMP-WR achieves 2–10 times higher throughput and lower energy consumption per delivered bit; and an exploration of protocol operation, providing users with guidance on protocol parameter values, and evaluation of CDMA as a physical layer access technology.

Section II provides an overview of STUMP-WR and Section III describes the underwater network model. A detail of the channel scheduling algorithm is provided in Section IV. Our simulation methodology is detailed in Section V and the results are located in Section VI. Finally, we conclude the paper in Section VII.

II. STUMP-WR Overview

While characteristics of underwater communications negatively affect many terrestrial wireless protocols, STUMP-WR leverages these characteristics to improve network performance over previous underwater protocols. Long propagation delays are used to overlap multiple transmissions and increase throughput while knowledge of those delays allows nodes to prevent collisions. Variable data rates allow nodes to optimize the time they schedule links for transmission and reception, either to save energy by sleeping more often or to increase network throughput. To accomplish these goals, STUMP-WR nodes use a distributed algorithm to select and schedule links using local state information.

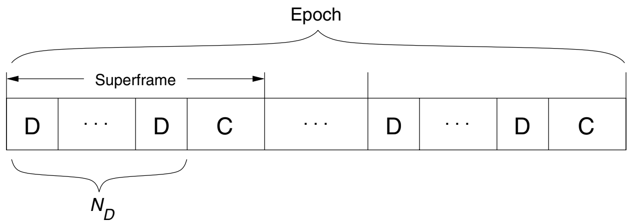

Fig. 1. STUMP-WR Epoch with N_C Superframes

STUMP-WR organizes its operation into time periods called *epochs*, during which nodes use fixed routes and link schedules while computing new values for use in the next epoch. The length of each epoch must be long enough for STUMP-WR to stabilize, while not reducing protocol adaptability to changes in network conditions. In Section VI we evaluate the required epoch length for STUMP-WR.

Each epoch, shown in Fig. 1, contains a set of N_C *superframes* used to transmit data and control packets, where each superframe contains N_D data frames followed by a control frame. For simplicity, both frame types use time slots of length T seconds, but different sized time slots may be used. Data frames are used to carry unicast traffic between nodes and control frames are used for broadcast packets, such as protocol updates, synchronization messages, and acknowledgments. Separate data and control frames are required due to the tight scheduling of unicast links, as detailed in Section IV.

STUMP-WR uses a distributed algorithm to select and schedule links. Nodes select and schedule links in two steps, which can be performed whenever a node receives new information. First, nodes select links to use and assign traffic loads to those links, both for packets generated at the node and packets forwarded from other nodes. Then, nodes update their link schedules based on a set of constraints derived from local interference patterns and the link schedules of neighboring nodes. Nodes broadcast their route and link schedule updates during control frames until the network converges.

A. Link Selection

Nodes select data links to use based on the number of hops to the destination and the energy level of their neighbors, which nodes periodically broadcast. Links with fewer hops to the destination are used first, as paths using these links consume lower energy. When a node has a choice among multiple links with the same hop count, it selects the link whose destination has the most remaining energy to extend the network lifetime. These link selection priorities reduce the total energy consumed during operation and distribute energy consumption throughout the network.

Once a node selects the links to use, it calculates the time slots required to send its data and control packets. Application requirements may dictate how much data traffic each node generates or nodes may adapt these reservations based on feedback from other sources, such as MAC layer queue depth. Nodes also reserve time slots during control frames to broadcast information, such as protocol state and acknowledgments. The amount of control slots each node requires depends on other protocols present in the network and may be fixed or time varying.

B. Link Scheduling

Scheduling links in terrestrial wireless networks requires ensuring nodes transmit at separate times and adding brief guard periods to account for propagation delay. However, this approach results in significant overhead for underwater networks due to the long propagation delays. With knowledge of the link propagation delays, nodes can schedule links to minimize idle time and reduce overhead.

Using the links selected previously, nodes schedule their transmissions to avoid collisions. STUMP-WR uses a distributed scheduling algorithm, based on the Bellman-Ford algorithm, to find a link schedule, where "distances" are functions of link durations, propagation delays, and frame size. The algorithm minimizes the frame size, as detailed in Section IV. Nodes must derive separate data and control schedules, as the different communication patterns require distinct conditions to prevent collisions.

III. Network Model

Underwater networks present unique characteristics for protocol design, which we now elaborate. Additionally, this section describes the device characteristics and communication resources available to the underwater devices under consideration.

Nodes communicate using a single half-duplex acoustic radio capable of single packet reception. The radios transmit acoustic signals using omni-directional transducers and are able to use power control to use the minimum amount of power to achieve a desired SNR at the receiver. Nodes are able to determine which neighbors they may communicate with and which neighbors receive only interference at a given power level.

The underwater acoustic channel provides a variable bandwidth based on node distances [12] due to the frequency dependent attenuation and noise characteristics of the channel. At farther distances, the optimal center frequency and available bandwidth decrease.

The radio and channel characteristics result in four possible link conflicts: *TX-TX* conflicts arise when a node attempts to transmit more than one packet at a time; *TX-RX* conflicts result when a node attempts to transmit and receive at the same time; the arrival of multiple packets at the same destination at the same time result in a *RX-RX* conflict; and *TX-RX-TX* conflicts are created when multiple packets arrive at a destination when only one packet is destined for the device. A valid schedule must avoid all these *network conflicts* to prevent collisions.

CDMA techniques help alleviate conflicts and are well suited for the underwater environment [13], but come at the cost of more complex hardware and longer transmission times [14]. We consider an ideal DSSS CDMA radio, where code sequences are assigned to nodes for transmission. Nodes using CDMA perfectly reject interference from nodes using a separate spreading code, but require SF times as long to transmit a packet. TX-RX-TX conflicts do not occur with CDMA.

Nodes maintain synchronization using additional timing protocols [15], [16] at the cost of extra energy and communication resources. Synchronization protocols also provide propagation delay estimates through their standard operation.

IV. Conflict-Free Scheduling

Solving for a link schedule involves finding the first transmit slot, s_i, for a link with propagation delay p_i that avoids collisions with all interfering links when the link is used for its required duration, Δ_i. A valid schedule avoids collisions by placing *schedule constraints* on the starting time slot of each link based on the conflicts present in the network. This section presents these constraints, which are used by a modified Bellman-Ford algorithm to find link schedules.

Nodes schedule link transmissions within a frame of m slots. While m may be fixed by the user, we describe how nodes can find the minimum frame size during protocol operation. For the following discussion we do not distinguish data frame size from control frame size to simplify the presentation, but STUMP-WR computes these as two separate values during default operation.

A. Link Schedule Constraints

Each network conflict yields a schedule constraint on the possible transmit times of the conflicting links, which the scheduling algorithm must resolve to avoid collisions. The specific schedule constraint differs for each conflict type, but we only derive TX-RX constraints in detail for brevity. The other constraints are derived in a similar manner. When links have multiple conflicts they will have multiple constraints, which are combined by taking the minimum of the upper bounds and the maximum of the lower bounds. In this way, the scheduling algorithm must only handle one constraint for each link conflict.

1) Example Derivation: Consider a TX-RX conflict between link j and link i, where the destination of j is the source of i. An example frame resolving this conflict is shown in Fig. 2, where the packet transmitted over j arrives before i is scheduled to transmit in that frame. The packet is completely received at time $s_j + \Delta_j + \frac{p_j}{T}$, measured in time slots, where p_j is the propagation delay along j and T is time period of each slot. Link i must transmit after the reception is complete, or $s_i \geq s_j + \Delta_j + \frac{p_j}{T}$. The schedule must also ensure that the transmission on i in the current frame does not end ($s_i + \Delta_i$) after the start of the reception on j in the next frame ($s_j + \frac{p_j}{T} + m$), which yields the inequality $s_j + \frac{p_j}{T} + m \geq s_i + \Delta_i$. So far the constraints are:

$$s_i \geq s_j + \Delta_j + \frac{p_j}{T}$$

$$s_j + \frac{p_j}{T} + m \geq s_i + \Delta_i$$

However, the packet over j can arrive after the transmission for i, which yields a similar pair of inequalities. We use the

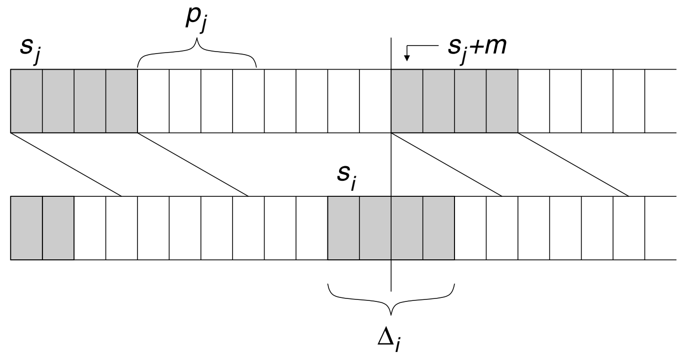

Fig. 2. Schedule Constraint for a TX-RX Conflict

ordering variable o_{ij} to distinguish between these two cases. $o_{ij} = 0$ in the former case, when the reception over j arrives before i is scheduled to transmit, and $o_{ij} = 1$ in the latter case.

Using the ordering variable o_{ij}, we combine these four inequalities into two general TX-RX schedule constraints:

$$s_j + \frac{p_j}{T} + m\,(1 - o_{ij}) \geq s_i + \Delta_i$$

$$s_i + m o_{ij} \geq s_j + \Delta_j + \frac{p_j}{T}$$

Finally, these can be combined and simplified into (1).

2) Constraint Equations: Following similar derivation techniques to the previous subsection, the schedule constraint equations for each conflict type are as follows.

TX-RX:

$$\Delta_i - \frac{p_j}{T} - m \leq s_j - s_i - m o_{ij} \leq -\Delta_j - \frac{p_j}{T} \quad (1)$$

RX-RX:

$$\Delta_i + \frac{p_i - p_j}{T} - m \leq s_j - s_i - m o_{ij} \leq -\Delta_j + \frac{p_i - p_j}{T} \quad (2)$$

TX-RX-TX:

$$\Delta_i + \frac{p_i - p_{(src_j, dst_i)}}{T} - m \leq s_j - s_i - m o_{ij} \leq -\Delta_j + \frac{p_i - p_{(src_j, dst_i)}}{T} \quad (3)$$

where $p_{(src_j, dst_i)}$ is the propagation delay from the source of j to the destination of i.

TX-TX:

$$\Delta_i - m \leq s_j - s_i - m o_{ij} \leq -\Delta_j \quad (4)$$

The schedule constraints so far have assumed nodes are perfectly synchronized and perfectly determine the propagation delay among nodes. However, errors in synchronization and propagation delay estimates may occur due to clock drift, node movement, and environmental changes. To account for this, the schedule constraints may be adjusted to add in guard periods around schedule events (packet transmissions or receptions). Previous work showed that the scheduling algorithms used by STUMP-WR scale well even with significant errors [11].

B. Scheduling Algorithms

Given the schedule constraints, nodes solve the system of inequalities to derive a valid schedule. However, nodes must first fix the ordering variables so that each schedule constraint contains only two unknown variables, those for the transmit times of each link. A system of inequalities of this form can be solved using the Bellman-Ford algorithm [17].

Several criteria exist for assigning ordering variables, but STUMP-WR assigns ordering variables based on hops to the destination to reduce packet delay and algorithm run time. Links farther from the destination transmit earlier than links closer to the destination, so a node receives a packet before it is scheduled to forward that packet. Links with equal hops are ordered based on link identifiers.

C. Frame Size

A schedule satisfying (1)–(4) prevents collisions between links within the same frame, but the frame size m must be sufficiently large to prevent collisions between transmissions in adjacent frames. However, nodes desire a minimal frame size that provides a high throughput, yet still prevents collisions. From Fig. 2, the idea is to minimize the gap between when i finishes transmitting and the packet arrives on j in the next frame.

A node accomplishes this in two steps: first, by calculating the valid range of slot assignments for each transmitting link at the node; then, by determining the new minimum frame size. Algorithm 1 details how nodes determine the minimum frame size. Nodes compute the bound for each link by solving the general link constraint $\underline{B}_{ij} \leq s_j - s_i \leq \overline{B}_{ij}$ for the transmit time of i and taking the minimum of the upper bounds and the maximum of the lower bounds across all conflicting links, $\mathcal{L}$, with earlier ordering (lines 5–10). Only links with an earlier ordering are considered since later links adjust their schedule based on i. The new frame size, m_{min}, is then calculated as $m - UB + LB$ (line 11). When the current frame is too small to resolve a conflict involving i, $-UB + LB$ will have a positive value, resulting in a new frame with additional slots. Likewise, $-UB + LB$ is negative when the current frame can be reduced without causing a collision.

Since the data schedule allows nodes to transmit across frame boundaries, the final data frame must be extended to prevent collisions between the data and control frames.

D. Control Schedule

One consequence of scheduling links close together is that nodes cannot easily broadcast information to neighboring nodes. To overcome this problem, STUMP-WR provides periodic control frames that allow nodes to broadcast packets to all neighbors. Scheduling broadcast packets requires a larger frame size, so aggregate throughput is decreased compared with data frames.

Algorithm 1 Minimum frame size m_{min} at node n

1: $m_{min} \leftarrow 0$
2: **for all** $i : src_i = n$ **do**
3: $\quad LB \leftarrow 0$
4: $\quad UB \leftarrow m - 1$
5: $\quad$ **for all** $j \in \mathcal{L}_i : i \neq j$ **do**
6: $\quad\quad$ **if** $o_{ij} = 0$ **then**
7: $\quad\quad\quad LB \leftarrow \max\left\{LB, s_j - \overline{B}_{ij}\right\}$
8: $\quad\quad\quad UB \leftarrow \min\left\{UB, s_j - \underline{B}_{ij}\right\}$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad m_{min} \leftarrow \max\{m_{min}, m - UB + LB\}$
12: **end for**
13: **return** m_{min}

STUMP-WR schedules nodes instead of links when deriving control schedules. Node a transmits in time slot s_a for a duration of Δ_a time slots, which is determined by protocol requirements. Control schedule constraint derivation is similar to that for a data links, but there is only one conflict type.

Consider two conflicting nodes, a and b. If b transmits after a, then b must wait until its transmission will not cause a collision with the transmission from a. Node b must wait until either the transmission from a reaches it ($p_{(a,b)}$) or until the signal propagates beyond a's interference range (p_a). Thus, b may transmit after $s_a + \Delta_a + \frac{\min\{p_a, p_{(a,b)}\}}{T}$. Similar arguments hold when a waits to transmit after b. Using these values and a similar derivation to that used for data link constraints yields the control schedule constraint in (5).

$$\Delta_a + \frac{\min\left\{p_a, p_{(a,b)}\right\}}{T} - m \leq s_b - s_a - m o_{ab} \leq - \Delta_b - \frac{\min\left\{p_b, p_{(b,a)}\right\}}{T} \quad (5)$$

The control frame size must be long enough to prevent collisions in the following data frame. To accomplish this the control frame size is extended in a similar way to the final data frame size.

V. Simulation Methodology

We evaluate the performance of STUMP-WR and compare it to other protocols through simulation of a network of 13 nodes, separated by 3500m on average, randomly distributed in a two dimensional grid. Simulations, using the OMNeT++ discrete event simulator [18], are run across 30 random node placements and the results show the mean result with 95% confidence intervals.

Each node consumes 80mW while idle, 3000mW while receiving, and negligible energy while asleep. Transmission power varies depending on receiver distance between 10W and 50W and data rates vary between 2kbps at 4000m and up to a maximum of 9.7kbps. Nodes generate traffic at random periods from an exponential distribution for a sink located at the network center. All data packet sizes are fixed at 2448 bits and control packet sizes vary depending on the amount of state information required by the protocols.

Scheduled protocols require node synchronization, so to account for this all scheduled protocols (STUMP-WR, TDMA, and ST-Lohi) periodically include 32 bits of overhead in their broadcast packets.

At network startup nodes perform an initialization procedure to gather local information, such as propagation delays, and determine the initial routes and link schedules. This initialization period is short compared with the total simulation time of two days, so no results from the initialization period are included.

For each protocol we consider several metrics. *Efficiency* is the total number of bits received at the sink divided by the energy consumed by all non-sink nodes. We also consider the *throughput* of the protocols, which equals the total data bits received at the sink divided by the simulation time. Finally, we examine *frame size*, measured in time slots, and *epoch size*, measured in the number of control frames required by the scheduled protocols in each epoch.

We evaluate four protocols, including STUMP-WR, for comparison. Both scheduled and unscheduled protocols are examined.

1) STUMP-WR: Control reservation is fixed for the duration of the simulation at 1408 bits and the epoch size was set at 25 control frames to allow the distributed algorithm to stabilize within the epoch. Each superframe contains 20 data frames (19 regular data frames and one final data frame) and one control frame.

2) Optimized TDMA: The STUMP-WR control schedule yields an optimal TDMA schedule when data and control packets are transmitted together and the frames are continually repeated.

3) T-Lohi: T-Lohi [5] is a random MAC protocol that uses small "tone" packets to contend for the channel and wake neighboring nodes. All versions of T-Lohi are considered (ST, aUT, and cUT). To approximate T-Lohi's wakeup radio, T-Lohi nodes consume zero idle energy. For some results, aUT-Lohi and cUT-Lohi produced similar results, so they are grouped under UT-Lohi for clarity.

4) Aloha: A modified version of Aloha that uses random delays before each transmission to reduce collisions [4].

VI. Results

We now compare the protocols through simulation, using several metrics for comparison.

A. Protocol Performance

As described in Section I, underwater network devices have limited energy and communication resources, so protocols must operate judiciously to provide sufficient data rates and useful lifetimes. Fig. 3 illustrates the efficiency of the evaluated protocols and shows that STUMP-WR delivers more bits per unit of energy. Similarly, STUMP-WR achieves this efficiency without sacrificing throughput, as shown in Fig. 4.

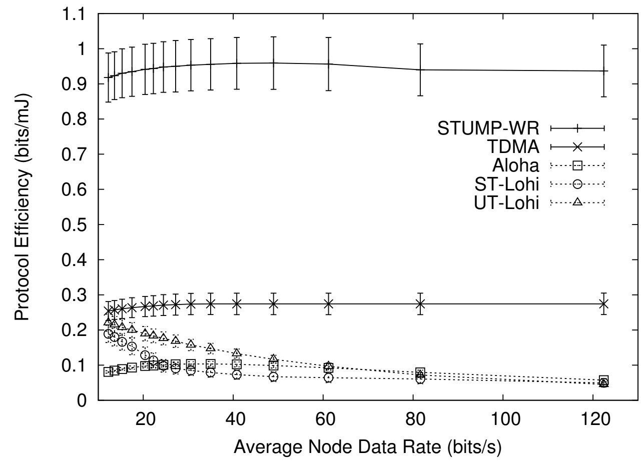

Fig. 3. Protocol Efficiency

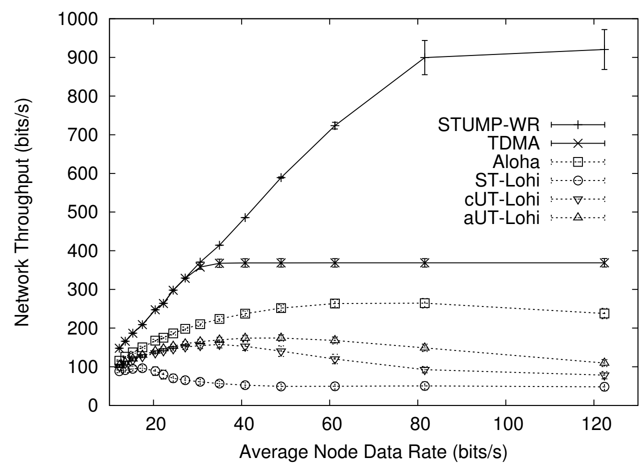

Fig. 4. Protocol Throughput

STUMP-WR outperforms other protocols because it avoids the energy losses of packet collisions. While the T-Lohi protocols reduce collisions through tones, they cannot achieve the throughput of the scheduled protocols due to the long channel access time required in congested networks [19].

B. Epoch and Frame Size

STUMP-WR must balance the benefits of short frames and brief convergence times with the requirement for valid schedules and an adaptability to changing network conditions. Epochs must be long enough for the STUMP-WR algorithm to stabilize before the network adopts new routes and schedules. STUMP-WR used 12.1 control frames on average and a maximum of 16 control frames per epoch, indicating an acceptable convergence time. TABLE I shows the frame sizes for the scheduled protocols. As TDMA does not distinguish between data and control frames, it has only one frame type. STUMP-WR yields a mean data frame size of 151.9 slots, which is 23% larger than the minimum frame size found across all networks, and final frame sizes were found to be about 14 time slots larger. This indicates that STUMP-WR yields a short frame size on average and can provide high throughput.

	Data Frame Size			Control Frame Size		
Protocol	Mean	Min	Max	Mean	Min	Max
STUMP-WR	151.9	123	216	326.6	287	361
TDMA	408.8	363	455	N/A	N/A	N/A

TABLE I
FRAME SIZES

	Data Frame Size			Control Frame Size		
Protocol, SF	Mean	Min	Max	Mean	Min	Max
STUMP-WR, 1	109.6	94	145	117.7	85	155
STUMP-WR, 8	742.0	659	930	197.7	145	265
TDMA, 1	221.4	205	258	N/A	N/A	N/A
TDMA, 8	981.6	873	1234	N/A	N/A	N/A

TABLE II
FRAME SIZES WITH CDMA

Additionally, comparing STUMP-WR and TDMA frame sizes illustrates the benefit of scheduling using propagation delays.

C. CDMA

CDMA allows nodes to send fewer, smaller control packets and limits information distribution to one-hop neighbors at the cost of longer transmission times due to the decreased data rates. TABLE II shows the frame sizes and TABLE III shows the required epoch sizes when using CDMA. The tables list results using CDMA with a spreading factor of 8 and an ideal case where CDMA does not decrease the data rate ($SF = 1$).

The results show that using CDMA reduces the required control frames (convergence time) approximately by one third, but increases the frame size in proportion to SF. Additionally, the maximum control packet size decreased from 984 bits without CDMA to 400 bits with CDMA, indicating the nodes shared less state. However, due to the increased transmission and reception durations, CDMA decreases protocol efficiency in proportion to SF.

VII. CONCLUSION

The long propagation delays and variable data rate of underwater networks complicate or invalidate many terrestrial wireless protocols. However, leveraging these characteristics can yield performance benefits to underwater networks, as we have shown with the proposed combined routing and scheduling protocol, STUMP-WR. Optimized for heavily loaded networks, STUMP-WR selects and schedules links using a distributed algorithm while purposefully considering the long propagation delays and varying link rates.

We have shown that STUMP-WR outperforms several other protocols proposed for underwater networks on the metrics of bits delivered per unit of energy and throughput. Finally, we showed how CDMA, a popular technique in underwater networks, can reduce state distribution requirements and decrease protocol convergence time.

	Control Frames Used		
Protocol, SF	Mean	Min	Max
STUMP-WR, 1	8.0	4	11
STUMP-WR, 8	7.4	4	10
TDMA, 1	8.3	5	11
TDMA, 8	8.0	3	11

TABLE III
EPOCH SIZE WITH CDMA

REFERENCES

[1] 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, Dec. 2008.

[2] X. Guo, M. R. Frater, and M. J. Ryan, "Design of a propagation-delay-tolerant MAC protocol for underwater acoustic sensor networks," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 170–180, Apr. 2009.

[3] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "Aloha-based MAC protocols with collision avoidance for underwater acoustic networks," in *Proc. of IEEE InfoCom*, May 2007, pp. 2271–2275.

[4] N. Parrish, L. Tracy, S. Roy, P. Arabshahi, and W. L. J. Fox, "System design considerations for undersea networks: Link and multiple access protocols," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1720–1730, Dec. 2008.

[5] A. A. Syed, W. Ye, and J. Heidemann, "Comparison and evaluation of the T-Lohi MAC for underwater acoustic sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1731–1743, Dec. 2008.

[6] M. K. Park and V. Rodoplu, "UWAN-MAC: An energy-efficient mac protocol for underwater acoustic wireless sensor networks," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 3, pp. 710–720, Jul. 2007.

[7] 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, Apr. 2009.

[8] K. Y. Foo, P. R. Atkins, T. Collins, C. Morley, and J. Davies, "A routing and channel-access approach for an ad hoc underwater acoustic network," in *Proc. of the Oceans Conference*, vol. 2, Nov. 2004, pp. 789–795.

[9] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K. C.-J. Lin, "ST-MAC: Spatial-temporal MAC scheduling for underwater sensor networks," in *Proc. of IEEE InfoCom*, Apr. 2009, pp. 1827–1835.

[10] M. Zorzi, P. Casari, N. Baldo, and A. F. Harris III, "Energy-efficient routing schemes for underwater acoustic networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1754–1766, Dec. 2008.

[11] K. Kredo II, P. Djukic, and P. Mohapatra, "STUMP: Exploiting position diversity in the staggered TDMA underwater MAC protocol," in *Proc. of IEEE InfoCom MiniConference*, Apr. 2009, pp. 2961–2965.

[12] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proc. of the Int. Workshop on UnderWater Networks (WUWNet)*, Sep. 2006, pp. 41–47.

[13] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005.

[14] L. Freitag, M. Stojanovic, S. Singh, and M. Johnson, "Analysis of channel effects on direct-sequence and frequency-hopped spread-spectrum acoustic communication," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 586–593, Oct. 2001.

[15] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "MU-Sync: A time synchronization protocol for underwater mobile networks," in *Proc. of the Int. Workshop on UnderWater Networks (WUWNet)*, Sep. 2008, pp. 35–42.

[16] A. A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. of IEEE InfoCom*, Apr. 2006.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2^{nd} ed. MIT Press, 2001.

[18] OMNeT++ discrete event simulator. http://www.omnetpp.org.

[19] F. Guerra, P. Casari, and M. Zorzi, "A performance comparison of MAC protocols for underwater networks using a realistic channel simulator," in *Proc. of the Oceans Conference*, 2009.