STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol

Kurtis Kredo II
Electrical and Computer Engineering Dept.
University of California, Davis
kbkredo@ucdavis.edu

Petar Djukic
Computer Science Dept.
University of California, Davis
pdjukic@ucdavis.edu

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

***Abstract*—In this paper, we propose the Staggered TDMA Underwater MAC Protocol (STUMP), a scheduled, collision free TDMA-based MAC protocol that increases channel utilization by leveraging node position diversity and the low propagation speed of the underwater channel. STUMP uses propagation delay information to overlap node communication and increase channel utilization. STUMP also provides an upper bound on the performance of many ad hoc MAC protocols previously proposed for underwater networks.**

Our work yields several important conclusions. First, leveraging node position diversity through scheduling yields large improvements in channel utilization. Second, STUMP does not require tight node synchronization to achieve high channel utilization, allowing nodes to use simple or more energy efficient synchronization protocols. Third, CDMA, a technique commonly proposed for underwater networks, provides no benefit to scheduled MAC protocols when using realistic spreading values. CDMA, however, still provides other benefits in underwater networks, so our work does not attempt to argue against using CDMA. Finally, we briefly present and evaluate distributed and centralized algorithms that derive STUMP schedules.

I. Introduction

Underwater wireless networks enable many applications, such as biological and environmental monitoring, resource maintenance and monitoring, and military defense. However, wireless technologies for underwater networks have only recently begun rapid development when compared to terrestrial networks. Advances in semiconductor, sensing, and other technologies have made underwater wireless communication feasible for many applications, but much research remains before underwater networks reach the level of sophistication and deployment seen in terrestrial wireless networks.

Wireless communication in underwater networks overlaps with terrestrial communication on several topics, but many unique characteristics in the underwater channel force protocol adaptation to provide good performance. One of the main differences between underwater and terrestrial networks comes from using acoustic communication underwater [1]. Using acoustics enables devices to communicate over long distances (tens of kilometers) at reasonable power levels, but introduces several channel characteristics not seen in terrestrial wireless networks. Extreme multipath [2], long propagation delay, and a channel capacity dependent on distance [3] differentiate communications in underwater and terrestrial networks. The slow sound propagation speed in water, at roughly 1500m/s, particularly affects protocol design. Low sound speed combined with the large distances common in underwater networks makes propagation delays on the order of several seconds common. Long propagation delays invalidate or decrease the usefulness of many techniques used in terrestrial networks, such as carrier sensing and control packet exchanges. Additionally, changing water conditions contribute a time varying aspect to these characteristics, limit underwater devices to low data rates (kilobits per second), and complicate time synchronization protocols.

Underwater MAC protocols attempt to limit or prevent packet collisions since they waste the limited energy and communication resources available to nodes. Communication consumes the majority of a node's energy, so retransmitting packets multiple times quickly reduces a node's lifetime. In addition to the time wasted transmitting during a collision, long propagation delays ensure nodes receive feedback about a collision after significant delay, which decreases performance.

Previous work in underwater network MAC protocols has focused on overcoming the challenges of the acoustic channel, but we exploit these characteristics through scheduling to improve network performance. Our Staggered TDMA Underwater MAC Protocol (STUMP) uses node propagation delay estimates to schedule overlapping transmissions without conflicts. Underwater nodes do not need to reserve the channel for long periods to prevent collisions as node position diversity might cause packets from two different nodes to arrive successfully, even if the packets were transmitted at the same time [4]. Nodes only need ensure that packets arrive during different times at the intended receiver.

We show that: STUMP, which also provides an upper bound on previously proposed underwater ad hoc MAC protocols, performs better than Aloha and optimized TDMA in underwater networks; STUMP handles significant synchronization error; CDMA provides no benefit to the protocols studied with realistic spreading values; and distributed and centralized algorithms exist for implementing STUMP in underwater networks.

We present the models and assumptions used in this paper and introduce STUMP in Section II. Section III details how we formulate the scheduling problem, while Section IV introduces distributed and centralized algorithms that find STUMP schedules. We evaluate the performance of STUMP and other protocols in Section V. Finally, Section VI provides our conclusions and directions for future work.

A. Related Work

Several channel access methods exist for underwater nodes. However, many methods used in terrestrial wireless networks perform poorly in underwater networks due to the unique device and channel characteristics. We introduce several classes of channel access methods proposed in the literature along with our proposed method, which leverages the unique conditions of the underwater channel.

Carrier sense multiple access (CSMA) performs poorly in underwater networks because long propagation delays prevent nodes from obtaining the current channel state. A node may sense the channel as busy long after another node finished transmitting. Likewise, a node may sense the channel as idle even if another node currently transmits. CSMA with collision avoidance (CSMA/CA) also suffers degradation in underwater environments as nodes reserve the channel for several propagation delays, which results in a large overhead per packet. Additionally, very large CSMA/CA control packets [5] waste energy.

Several researchers have attempted to modify CSMA/CA for use in underwater networks. To reduce the overhead associated with control packet propagation, some proposals [6], [7] adjust the timing between control packet exchanges, thus allowing neighbors to learn of impending communications, but with smaller delays. Other work [8], [9] attempts to increase the channel utilization by overlapping control and data packets from several transactions through ad hoc scheduling with control packets.

Aloha has many advantages for use in underwater networks, such as no control packet overhead, no synchronization requirements, and no sensitivity to channel variations. Syed et al [4] examined the effect of node location and transmit time uncertainty on Aloha's performance and found that underwater nodes must use slotted Aloha with an increased time slot size in order to equal the performance of Aloha in terrestrial networks. Other researchers [10] propose using small control packets to weakly schedule the reception of data packets and remove some of the transmit time uncertainty. Aloha's disadvantages lie in the large energy waste from packet collisions and the low achievable throughput.

Similar to CSMA/CA and Aloha, our work overlaps communications to improve performance, but we focus on periodic, high data rate applications. CSMA/CA and Aloha are more suited for low rate, random traffic patters. Additionally, we provide mechanisms for users to bound packet delays, a feature unavailable in unscheduled protocols. Since STUMP does not require control packet exchanges, but still overlaps communication similar to previous work, it yields an upper bound on the performance of previous underwater ad hoc MAC protocols [6], [7], [8], [9]. The tradeoff for the improved performance is the overhead of schedule generation and, in some cases, node synchronization.

Researchers have proposed code division multiple access (CDMA) techniques in underwater networks to reduce collisions and combat the unique conditions of the underwater channel [11], [12]. Research efforts with CDMA include: using CDMA with multiple user reception to prevent control packet collisions [13], [14], using CDMA to reduce the probability of collisions [15], and exploring the role of CDMA in clustered topologies [16], [17]. While CDMA provides benefits to underwater nodes in many cases, it decreases the already low data rate available to applications.

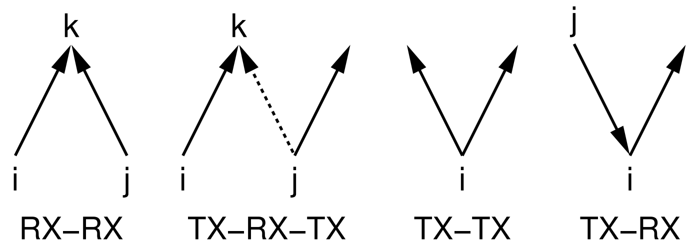

Fig. 1. Scheduling Conflicts

Our work explores using CDMA as a physical layer access mechanism to determine the impact at the MAC protocol level. Exploring the impact of CDMA allows users to better evaluate whether or not to use CDMA in an actual underwater network.

Time division multiple access (TDMA) yields good performance in terrestrial networks under heavy load. However, when used in underwater networks, TDMA performance declines due to the large time slots required to prevent collisions. Each time slot must be long enough for the transmission of a packet plus a guard time that enables the signal to propagate beyond the interference range of the sender. Additionally, varying channel conditions and long propagation delays complicate synchronization protocols required by TDMA, leading to larger time slot overhead. The largest benefit of TDMA is the deterministic performance and energy savings from the absence of collisions.

Our work modifies TDMA to decrease the overhead associated with preventing collisions. We do this by using the long propagation delays to overlap transmissions of nearby nodes, while TDMA only allows nodes to transmit simultaneously if they are out of interference range of each other.

II. Staggered TDMA Underwater MAC Protocol and Network Model

To evaluate STUMP, we model a typical underwater network designed to gather data and forward it to a remote user through a single gateway node, called the sink. Nodes have a single, half-duplex radio capable of single packet reception. Node interference follows the protocol model [18], where each node has an interference range twice its communication range. Any two packets arriving at a node at the same time cause a collision and prevent the node from gaining any information about either of the two packets. These assumptions yield four possible conflicts in the network [19], as illustrated in Fig. 1.

Single packet reception requires packets destined for a node arrive without overlap or a RX-RX conflict arises, causing a collision. TX-RX-TX conflicts are similar, but the result of one node interfering with the transmission of another pair of nodes. In Fig. 1 node j causes a collision with the packet from node

i to node k in the TX-RX-TX conflict. TX-TX conflicts arise from nodes having only one radio, which prevents them from transmitting to two different destination nodes at the same time. Finally, half-duplex radios require nodes to transmit or receive at the same time, but not both, as that would cause a TX-RX conflict.

It is important to remember that, unlike terrestrial networks, collisions in underwater networks involve not only the conflicts, but also the temporal relationship between node transmissions. For example, consider a TDMA protocol and the RX-RX conflict in Fig. 1. In a terrestrial network, node i and node j only cause a collision if they transmit in the same time slot. However, nodes i and j may transmit at the same time without causing a collision in an underwater network if their propagation delays to node k are sufficiently different. Similarly, nodes i and j cause a collision, even with different transmit times, if their propagation delays result in the packets arriving at the same time at node k.

CDMA allows nodes to distinguish different transmissions using spreading codes. Assuming each node uses a unique code for transmission, CDMA removes the possibility of TX-RX-TX conflicts. In this work we assume that using CDMA is subject to all scheduling constraints except for TX-RX-TX conflicts, but increases transmission time by the spreading factor, SF, due to the fixed bandwidth of the underwater channel. Note that we treat CDMA ideally and that its performance depends on many other factors [20].

We assume routing paths remain stable for long periods of time and nodes generate constant traffic at moderate to large volumes. Nodes communicate over multi-hop paths to a central sink, which is the destination of the bulk of network traffic. Each node generates traffic for the sink node, called the uplink traffic, and the sink generates traffic for each node individually, called the downlink traffic. However, our work allows other traffic patterns as well. Applications provide the routing protocol with traffic load requirements. After determining routes, the routing protocol provides the MAC layer with the number of slots to schedule for the next hops. Nodes share transmission durations within a two-hop neighborhood.

In addition to the next hop schedule requirements, STUMP requires propagation delay estimates from its neighbors. This information can be found during network setup and periodically updated during operation. Based on our model, nodes only need to know the propagation delay estimates to their one-hop neighbors and the delay estimates between one-hop and two-hop neighbors. Nodes remain stationary, so the delay estimates vary only slightly as nodes drift on their tethers and ocean conditions change. We model this variance and show its impact on protocol performance.

The MAC protocols organize transmissions into slots using a repeating frame of m slots. Each node transmits in contiguous slots as required by the traffic and routing conditions. Note that this differs from the traditional TDMA structure where each node transmits entirely in one time slot. Depending on the schedule constraints and network conditions, several time slot may be schedule between transmissions to ensure collisions do not occur.

Two options exist for determining the MAC frame size. First, a fixed frame size may be used to simplify protocol operation. In this case nodes have a fixed throughput and "better" schedules are those that utilize less of the frame. A fixed frame size, with a portion reserved for constant traffic, would be beneficial if the nodes being scheduled supported mobile nodes during the inactive portion of the frame. Without mobile nodes to support, the user would have to select a frame size that balances the time spent finding a schedule that fits within the frame with the overhead of wasted space in a frame, resulting in a lower than necessary throughput. The other option is to use a variable sized frame that changes according to the current schedule. Changing the frame size rarely, for example, only when it is a certain percent larger or smaller than needed, decreases the frequency of broadcasting the new frame size throughout the network.

Scheduled protocols require nodes maintain time synchronization among themselves to prevent schedules from drifting and causing collisions. To model synchronization error, we define σ as the maximum synchronization error at any node from a global time. Thus, the local time between any two nodes differs by at most 2σ. While synchronization in underwater networks is more complicated than terrestrial networks, protocols exist for underwater networks [21]. Additionally, variances in the water characteristics and node position result in varying propagation delays between any pair of nodes. We define π as the maximum error experienced in estimating the one way propagation delay between any two nodes. Hence, a packet sent at time t could arrive anywhere during the interval $(t-\pi, t+\pi)$. Note that propagation delay variances also affect synchronization protocols, but we model those effects within the parameter σ.

A. Staggered TDMA Underwater MAC Protocol

STUMP uses node position diversity, through propagation delay estimates, to overlap communications and improve channel utilization. With propagation delay estimates, STUMP can reduce schedule overhead and ensure packets do not collide at the receiver. This allows nodes to transmit at the same time as long as their packets arrive during different times at the intended destinations. STUMP uses the propagation delay information to develop conflict-free schedules that have a higher utilization than terrestrial protocols.

To divide the channel and allow finer scheduling, STUMP utilizes logical rings surrounding each node. Fig. 2 shows how STUMP segments the area surrounding a node into concentric rings. Rings are numbered from the center outward, starting at zero. With this scheme, nodes develop a schedule to transmit to each ring within each frame. Nodes only "reserve" the channel long enough to transmit to a particular ring and the scheduling algorithms ensure that packets do not collide by separating the transmission times by sufficient time slots.

Contrast this with TDMA, which schedules each node to transmit once per frame and adds guard bands at the end of

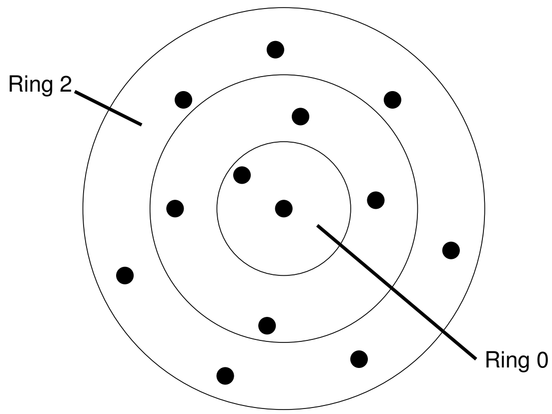

Fig. 2. STUMP Rings

each transmission; TDMA effectively reserves all the rings surrounding a node for each transmission. Note that, while each node transmits to multiple rings, it does so contiguously and only transmits once per frame.

Ring width is determined by the the length of each time slot. If each time slot has a length T and the speed of sound in water equals c, then each ring is cT wide in space. Note that the time slot and ring widths do not have to obey this relationship in general, but the assumption simplifies the schedule constraints and is used for clarity. Nodes do not need location information, only the propagation delay estimate to assign a node to a particular ring.

III. Conflict-Free Scheduling

We now define the characteristics of a valid schedule and define what that means for STUMP and TDMA. In both cases, a valid schedule is a set of time slots assigned to each node for transmission, $\mathbf{S} = \{s_i\}$, that prevents all the conflicts in Fig. 1 and satisfies the demand, $\mathbf{\Delta} = \{\Delta_i\}$, of all nodes. We allow multiple packets to arrive at a node if none of the packets were destined for that node. In each frame, a node i is assigned Δ_i continuous slots, starting at slot s_i.

A valid schedule must ensure all conflicts are resolved to prevent collisions. To accomplish this, we define constraints, $\mathcal{C}$, on the schedules of each protocol that ensure node transmission times are sufficiently separated in the frame. Each conflict in the network requires a schedule constraint to resolve it. TDMA constraints, which simply prevent collisions through large guard periods, are the same for each node, but STUMP constraints, which utilize propagation delay information, optimize the constraints for each possible conflict. Nodes develop STUMP schedule constraints based on local topology, traffic patterns, and propagation delays among their neighbors.

Within each frame there exists an ordering between event pairs, such as transmissions and receptions. When formulating the schedule constraints, we use binary ordering variables, $\mathcal{O} = \{o_{ij}\}$, to determine in what order events occur within a frame. Each constraint contains an ordering variable used to resolve the associated conflict. For TX-RX conflicts, $o_{ij} = 1$ when node i transmits in the frame before receiving the the packet from node j. $o_{ij} = 1$ in a RX-RX conflict when the destination receives its packet from node i before it receives its packet from node j. Finally, TX-RX-TX constraints have $o_{ij} = 1$ when the packet from node i arrives at the destination before the interference from node j arrives. Fig. 1 illustrates the conflict types with the nodes labeled appropriately.

Solving for a valid schedule involves finding sets $\mathcal{S}, \mathcal{O}$ that satisfy the schedule constraints $\mathcal{C}$ and node demands $\cdot$. We now develop constraints for TDMA and STUMP and discuss how to solve for valid schedules in Section IV.

A. STUMP Schedule Constraints

Since nodes transmit to multiple rings per frame, each node is assigned a starting slot for rings where $\Delta > 0$. Node i transmitting to ring α transmits for $\Delta_{i\alpha}$ time slots starting in slot $s_{i\alpha}$. When comparing two nodes, define r_{ij} as the ring node j occupies relative to node i. r_{ij} may not equal r_{ji} due to different propagation paths between the two nodes.

We develop the schedule constraint for the RX-RX conflict in detail, but introduce the remaining constraints more briefly since they are developed very similarly.

1) RX-RX Conflicts: The schedule resolves a RX-RX conflict by preventing the transmissions from node i and node j from colliding at node k. If the transmission from node i arrives at node k first, $o_{ij} = 1$, then the schedule must ensure the packet from node j arrives after node k finishes receiving node i's packet. Node k finishes receiving node i's packet at time $s_{ir_{ik}} + \Delta_{ir_{ik}} + r_{ik} + (0, 1)$. The packet from node j arrives at $s_{jr_{jk}} + r_{jk} + (0, 1)$. The ranges $(0, 1)$ come from the uncertainty of node location within a particular ring. To prevent collisions, consider the worst case, which yields the inequality $s_{ir_{ik}} + \Delta_{ir_{ik}} + r_{ik} + 1 \leq s_{jr_{jk}} + r_{jk}$. A valid schedule must also ensure that node j's transmission does not cause a collision with the transmission of node i in the next frame and the inequality $s_{jr_{jk}} + \Delta_{jr_{jk}} + r_{jk} + 1 \leq s_{ir_{ik}} + r_{ik} + m$ ensures this. A similar pair of inequalities arise when $o_{ij} = 0$. Combining the four inequalities and adding buffer time slots for synchronization and propagation delay estimate errors yields the STUMP constraint for RX-RX conflicts in (1).

$$\begin{aligned}\Delta_{ir_{ik}} + r_{ik} - r_{jk} + \frac{2\sigma + 2\pi}{T} + 1 - m\left(1 - o_{ij}\right) \leq \\ s_{jr_{jk}} - s_{ir_{ik}} \\ \leq -\Delta_{jr_{jk}} + r_{ik} - r_{jk} - \frac{2\sigma + 2\pi}{T} - 1 + m o_{ij} \quad (1)\end{aligned}$$

2) TX-RX-TX Conflicts: In the case of TX-RX-TX conflicts, the schedule must ensure an interference packet does not arrive at a node while it is receiving a valid packet. This condition is nearly identical to the RX-RX conflict. If we simply use node j as the interfering node, we can add conflicts between each pair of nodes i and j when node j could interfere with a transmission from node i. The resulting constraint is identical to (1). Recall that this constraint does not exist in systems that use CDMA under the assumption that each transmitter uses a separate code.

3) TX-TX Conflicts: Each node may only transmit to a single ring at a time, so a valid schedule must include constraints to ensure this. For simplicity, we assume nodes transmit all packets to the outermost ring first, then sequentially to inner rings until they have transmitted all their packets. Thus, $s_{i0} = s_{i1} + \Delta_{i1}, s_{i1} = s_{i2} + \Delta_{i2}, \ldots$. Formally,

$$s_{i\beta} - s_{i\alpha} = -\Delta_{i\beta} \qquad \text{where} \quad \alpha + 1 = \beta \tag{2}$$

In cases where a node does not transmit to a particular ring, then $\Delta = 0$ and the neighboring rings would be "assigned" the same starting slot. Note that this schedule constraint only exists between transmit times of the same node to different rings, never between two different nodes.

4) TX-RX Conflicts: All nodes have half-duplex radios, so the schedule must ensure that a node does not transmit when it is receiving a packet. The schedule allows a node to transmit a packet while the local channel is busy if the node is not the destination of the packet on the channel and the node's transmission will not cause a collision. There may be multiple TX-RX conflicts between a pair of nodes, with one for each transmit ring of the destination. If node i, the receiver, transmits first to some ring n, then $o_{ij} = 1$ and the schedule must ensure $s_{in} + \Delta_{in} \leq s_{jr_{ji}} + r_{ji}$ and $s_{jr_{ji}} + \Delta_{jr_{ji}} + r_{ji} + 1 \leq s_{in} + m$. Simplifying the inequalities, adding buffer for synchronization and propagation delay errors, and combining with the inequalities from $o_{ij} = 0$ yields the general constraint (3).

$$\Delta_{ir_{in}} - r_{ji} + \frac{2\sigma + \pi}{T} - m\left(1 - o_{ij}\right) \leq s_{jr_{ji}} - s_{ir_{in}} \leq -\Delta_{jr_{ji}} - r_{ji} - 1 - \frac{2\sigma + \pi}{T} + mo_{ij} \tag{3}$$

The STUMP scheduling problem, thus, involves finding the set of transmission times, $\mathcal{S}$, that satisfies the schedule constraints (1), (2), and (3) for the conflicts in the network, given the node demand $\cdot$.

B. TDMA Schedule Constraints

We use an "optimal" TDMA in the sense that the guard time slots assigned to a node are minimal to guarantee collision-free operation. Unlike traditional TDMA protocols, which add guard slots long enough to accommodate the full propagation range of a node, the TDMA used here only adds guard slots to reach the furthest node in interference range of a transmitter. Define G_i as the guard slots required after the transmission of node i using TDMA. With a maximum propagation delay of p_i to the furthest neighbor of node i in its interference range, we calculate the guard slots as:

$$G_i = \left\lceil \frac{p_i + 2\sigma + 2\pi}{T} \right\rceil$$

Since the guard slots prevent collisions between nodes that transmit at different times, the schedule only needs to ensure that nodes which cause interference to each other are assigned non-overlapping time slots. If node j transmits after node i ($o_{ij} = 1$) and they cause interference to each other, the schedule must ensure that $s_j \geq s_i + \Delta_i + G_i$. Additionally, the schedule must ensure the transmission from node j does not overlap with the time slots of node i in the next frame, so $s_j + \Delta_j + G_j \leq s_i + m$. Combining these along with the inequalities from the condition of $o_{ij} = 0$ yields the general TDMA constraint

$$\Delta_i + G_i + (1 - o_{ij}) \leq s_j - s_i \leq -\Delta_j - G_j + mo_{ij} \tag{4}$$

A valid TDMA schedule is the set of all transmit times, $\mathcal{S}$, such that (4) holds for each pair of conflicting nodes. Due to the guard slots, this single constraint suffices to prevent collisions from all conflict types. However, when using CDMA with TDMA, nodes that do not transmit to each other or share a common destination are not constrained, as CDMA prevents TX-RX-TX conflicts. Note that since each node transmits only once per frame, TX-TX conflicts are avoided.

IV. Scheduling Algorithms

The previous section detailed the TDMA and STUMP scheduling problems, but gave no insight into how to find the schedules. We now present several algorithms, both distributed and centralized, to solve the scheduling problems.

The TDMA and STUMP scheduling problems, specific instances of a Periodic Event Scheduling Problem, are NP-Complete [22], but we simplify the problem by solving it in two steps [23]. First, we determine the ordering variables, o_{ij}, by prioritizing nodes. If node i has a higher priority than node j, then $o_{ij} = 1$, otherwise $o_{ij} = 0$. Only conflicting nodes need to determine relative priority since ordering variables do not exist between nodes without schedule conflicts. With fixed ordering variables, the scheduling constraints become a set of difference equations, which we solve using the Bellman-Ford algorithm [24]. The problem difficulty now lies in finding a good set of ordering variables.

We use two metrics to compare the algorithms and protocols under study: the network throughput and the maximum uplink delay. The network throughput is calculated as the number of slots used by the sink for transmission or reception divided by the number of time slots in the frame, m. For variable sized frames, which change on each schedule calculation, the throughput has a straightforward meaning. However, networks that use a fixed frame size have the same throughput for every schedule. However, the throughput results indicate how much of a frame the scheduled portion requires. A lower throughput means more time slots are used by the scheduled portion and there are fewer time slots available for other activities (such as supporting mobile nodes). Delay is also an important metric in many networks, so we evaluate the uplink delay as the maximum delay experienced by any uplink traffic in the network.

We present two distributed algorithms that determine the ordering variables for the nodes. These algorithms could be combined with other protocols in the network, such as synchronization and routing protocols, to reduce energy, but we leave this for future work. Lastly, we present integer linear programming problems that find the optimal values for throughput and uplink delay, which could be solved centrally.

A. Random Ordering

A simple way to find node priorities is to select them at random, such as from node ids or random numbers generated locally. However, each pair of conflicting nodes must select unique priorities so the ordering variables are well defined. Graph coloring algorithms, such as DRAND [25], satisfy these requirements, but other methods are possible.

Selecting priorities at random requires little effort from the nodes, but it does not guarantee any level of performance. However, we show that random schedules yield characteristics useful in some applications. An alternative way to improve random selection would be to compute several schedules in parallel and use the schedule with the best performance.

B. Uplink Delay Ordering

Another way to select node priorities is to setup the schedule so packets arrive at the sink within a single frame. To do this, a node simply selects a lower priority than any neighbor that relays traffic to it. Leaf nodes would have the highest priority since they must transmit earliest in the frame and the last node on a path before the sink would have the lowest priority.

While this ordering bounds uplink delay to a single frame, it does not guarantee the minimum delay since nodes that forward traffic to a common relay (an RX-RX conflict) may get the same priority. Thus, nodes must have a secondary way to select ordering variables, such as with node id, in cases where priorities are equal. The secondary comparison may choose incorrectly, resulting in non-optimal uplink delay.

C. Linear Program Formulation

To evaluate the distributed algorithms, we define integer linear programming problems that find the minimum uplink delay and minimum frame size for a given routing and network topology. We present the problems in the context of STUMP, but they are equivalently defined for TDMA. Additionally, these problems could be solved centrally at a node with sufficient computational resources if given the network state.

1) Optimal Throughput: The data transmitted in each frame remains constant, so minimizing the frame size yields the optimal throughput. However, attempting to optimize m within a scheduling problem yields a non-linear integer problem, which requires significant resources to solve. Therefore, we approximate the optimal frame size by finding the minimum number of time slots, a, required to schedule all the nodes. Since nodes may be scheduled in fewer time slots than required for a complete frame, we add extra buffer slots. We assume the worst case of 14 time slots, which equals the time to reach the interference range of a node, so the optimal throughput may be slightly larger than we present.

We compute a by solving the following integer linear program

$$\min a$$

such that

$$a \geq s_i + \Delta_i - s_j + 14$$
$$s_i \geq 0$$

for all schedule variable pairs s_i, s_j along with the appropriate TDMA, (4), or STUMP, (1)–(3), constraints.

2) Minimum Uplink Delay: We find the optimal uplink delay by minimizing the maximum delay path in the network. The delay over the hop from node j to node i equals $s_i - s_j$ if node i transmits after receiving the packet from node j. However, if node i transmits before receiving node j's packet it must wait until the next frame to forward the packet, resulting in a delay of $s_i + m - s_j$. Generalizing the result yields the delay along each hop of a routing path as $s_i - s_j + mo_{ij}$. Summing across all hops in a path yields $s_\alpha - s_\beta + m\sum_{(i,j)\in P} o_{i,j}$, where P contains the node pairs along the path, α is the node that transmits to the sink, and β is the source node. Last, we add the duration and propagation delay, p_α, of the last hop to yield the integer linear program

$$\min d$$

such that

$$d \geq s_\alpha + \Delta_\alpha + p_\alpha - s_\beta + m \sum_{(i,j)\in P} o_{i,j}$$
$$s_i \geq 0$$

for each path to the sink in the network along with the appropriate TDMA, (4), or STUMP, (1–3), constraints.

V. Results

We now examine how STUMP compares with the TDMA and Aloha protocols by evaluating their average throughput and delay performance over 100 random topologies. For a consistent comparison, each protocol is evaluated over the same 100 random topologies. We do not simulate Aloha in our networks, but instead compare against Aloha's theoretical optimal throughput [26], [4]. We show that STUMP achieves higher throughput than TDMA and (optimal) Aloha, lower latency than TDMA, and tolerates large synchronization error.

Energy is an important metric for underwater acoustic networks, but we do not explicitly measure it in this work for two reasons. First, TDMA and STUMP consume the same energy under identical traffic conditions, so their results would be very similar. Second, random protocols waste significant energy in high load situations through collisions, so their efficiency would be very low. We leave to future work the energy performance comparison of STUMP to random protocols under light to moderate load.

Each topology consists of nodes deployed in a grid pattern made of square cells 3500m on a side. Cells with a center within 7500m of the sink, except for the cell containing the sink, receive a node, resulting in a network of 12 nodes plus the sink. To simulate the effect of currents and ship movement during deployment, we uniformly at randomly select a position from the center third of a cell for each node. Nodes remain stationary

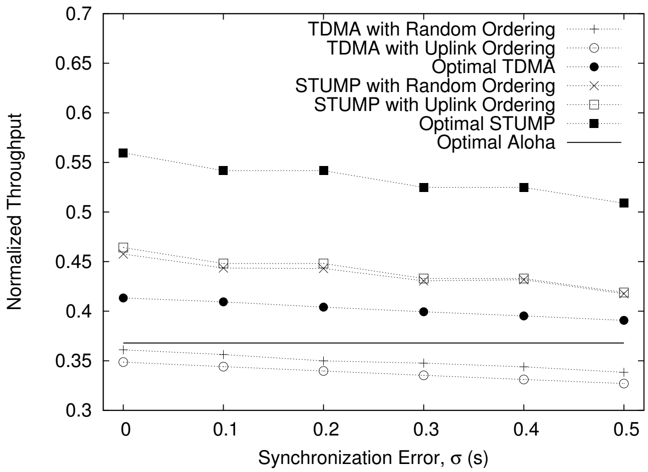

Fig. 3. Normalized Throughput as σ Varies

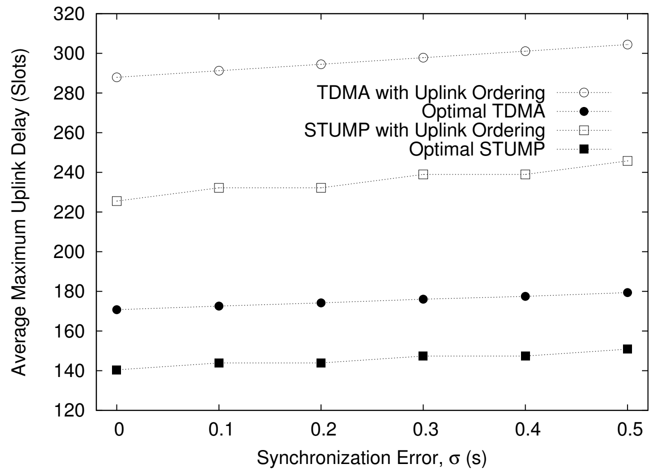

Fig. 4. Average Maximum Uplink Delay as σ Varies

during operation, but small movements due to ocean currents are modeled in the propagation delay error, π. For example, nodes tethered to the ocean floor by anchors move slightly as currents flow through the network. This paper models a two dimensional network, but our work applies just as readily to networks with nodes at different depths.

Nodes have a communication range of 4000m and an interference range of 8000m. Each frame consists of time slots with a duration of 0.4s. Unless stated otherwise, each node generates 10 time slots worth of data destined for the sink and the sink sends one slot worth of data to each node in each frame. Nodes generate routing paths according to a lifetime maximizing protocol [27].

A. Synchronization Error Impact

We first examine how the protocols perform as the synchronization error, σ, varies. Since synchronization and propagation delay estimate errors impact the schedules in a similar way, we only vary σ and leave the propagation delay error, π, equal to zero.

Fig. 3 shows how the throughput of each protocol varies with synchronization error. STUMP achieves a much higher throughput than TDMA and Aloha, indicating scheduling protocols can gain large benefits by using propagation delay information to overlap communications in underwater networks. The benefit of overlapping communication is so large that using STUMP with the non-optimal distributed scheduling algorithms yields better performance than TDMA or Aloha can ever achieve.

Additionally, STUMP adapts to synchronization error with marginal degradation. When $\sigma = 0.5$s, meaning node clocks may differ by up to a second, STUMP decreases its throughput by less than 10%. Thus, nodes may use lower energy, but less precise synchronization protocols without significantly affecting STUMP's performance. Fig. 3 also shows that, due to the long guard slots, TDMA requires near optimal schedules to perform better than Aloha.

Next, compare the maximum uplink delay for the protocols as synchronization error varies, as displayed in Fig. 4. STUMP again performs better than TDMA, achieving a 20% lower latency on average when using the Uplink ordering. However, unlike throughput, STUMP with Uplink ordering does not perform better than TDMA with an optimal ordering due to sub-optimal choices made by the ordering algorithm. Similar to the throughput results, even with significant synchronization error, STUMP's latency increases by less than 9%. Fig. 4 does not show the Random ordering results because those values depend on the frame size selected.

The delay results have shown that Uplink ordering provides a very low maximum uplink delay when compared to Random ordering, but this comes at the cost of downlink delay. Fig. 5 shows both the uplink and downlink delay for both ordering algorithms. Uplink ordering results in very high downlink latency since all the nodes transmit in exactly the wrong order, resulting in an average of nearly two frames of delay. However, the Random ordering performs consistently for both uplink and downlink traffic. This indicates that networks which require fast response times, such as surveillance or monitoring networks, would benefit from Uplink ordering, but general purpose networks may desire Random ordering since it provides balanced and equal performance to all traffic.

B. Performance with CDMA

Many proposals in the underwater network literature use CDMA to reduce collisions and for its ability to negate some negative underwater channel characteristics, so we investigate the impact of using CDMA with STUMP and TDMA. Adding CDMA involves increasing the transmission durations by the spreading factor, SF, for the benefit of eliminating all TX-RX-TX conflicts. These are highly idealized conditions, but we show that CDMA only improves TDMA and STUMP with unrealistic spreading factor values. Fig. 6 shows the normalized throughput for various spreading factors. Notice that when $SF = 1$ all protocols improve dramatically since the TX-RX-TX conflicts are removed without a decrease in the data rate. However, for larger spreading factor values, using CDMA does not yield any improvements in throughput over the protocol

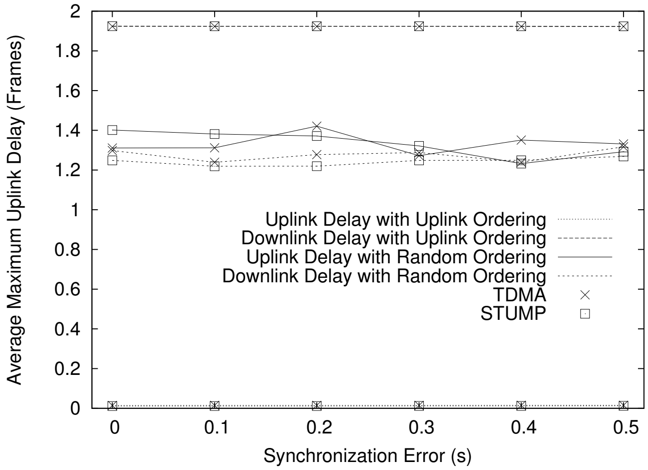

Fig. 5. Average Maximum Delay as σ Varies

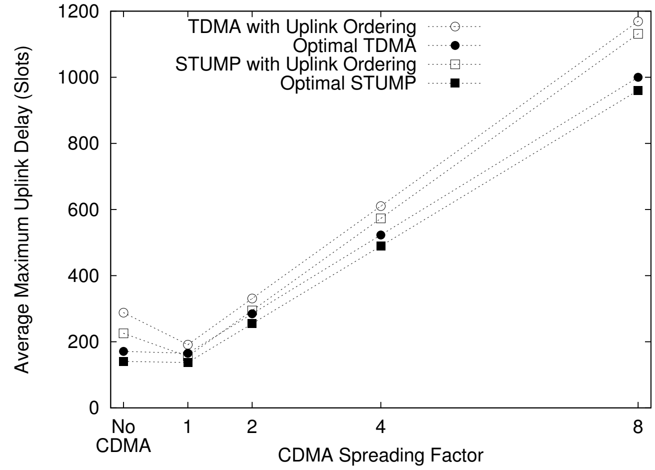

Fig. 7. Average Maximum Uplink Delay as SF Varies

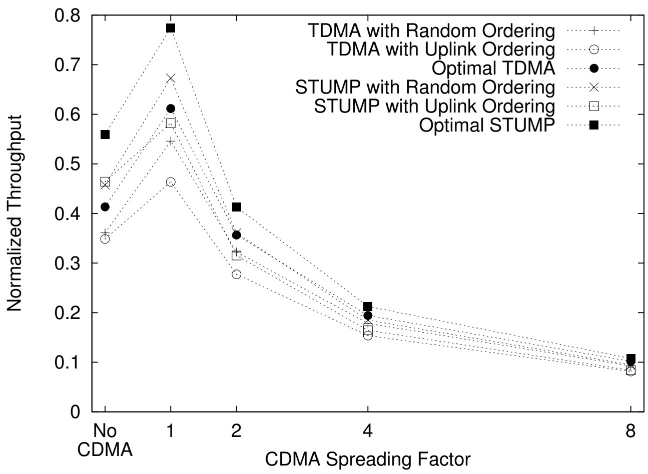

Fig. 6. Normalized Throughput as SF Varies

without CDMA. Thus, for realistic spreading factor values (much larger than 2 [15], [11]), CDMA does not provide any benefit in and of itself at the MAC layer.

CDMA also affects the performance of the scheduling algorithms. Without CDMA, both scheduling algorithms achieve a higher throughput with STUMP than the optimal possible with TDMA. However, as the spreading factor increases, Random ordering results in higher throughput than Uplink ordering. This guides users to use Uplink ordering with STUMP when CDMA is not used as it achieves a similar throughput to Random ordering with a much lower uplink latency, but use Random ordering with CDMA if throughput is important.

Fig. 7 shows the maximum delay averaged over all the networks as the spreading factor varies. Similar to the throughput results, STUMP always achieves lower latency. Also notice that Uplink ordering diverges from the optimal latency as the spreading factor increases much more than it did in the throughput results.

C. Varying Traffic Load

Some applications may not require ten times more uplink traffic than downlink traffic and these parameters have a large impact on protocol performance, so we investigate several different traffic loads. We varied the uplink traffic from each node to 10, 5, or 1 time slots and the downlink traffic as either 1 or 0 time slots to each node. Fig. 8 shows the results for various traffic loads. As before, STUMP outperforms TDMA for all traffic loads. However, at low traffic levels, STUMP can not achieve the throughput possible with Aloha when using the distributed ordering algorithms. As expected, random protocols perform better at low data rates, but scheduled protocols, especially STUMP, perform better at moderate to high data rates. Note that scheduled protocols achieve their throughput without causing collisions, so their energy consumption is potentially lower than Aloha for the low data rates. Similar trends between TDMA and STUMP were found with the uplink delay, but are not included.

Fig. 8 also shows the optimal values for TDMA and STUMP. As indicated there still remains significant room for improvement in ordering algorithms optimized to find schedules with a high throughput. Also note that even at the lowest data rates, STUMP can achieve Aloha's maximum throughput with the proper ordering.

VI. Conclusions

We have shown that synchronized protocols perform well in underwater environments if propagation delay is used to improve channel scheduling. The Staggered TDMA Underwater MAC Protocol increases the performance of the traditional TDMA protocol by using propagation delay estimates to schedule overlapping transmissions that do not collide at the receiver. We have shown:

- Utilizing propagation information from neighboring nodes allows scheduled protocols to perform well by overlapping communications and reserving smaller portions of the channel. STUMP outperformed "optimal" TDMA in all

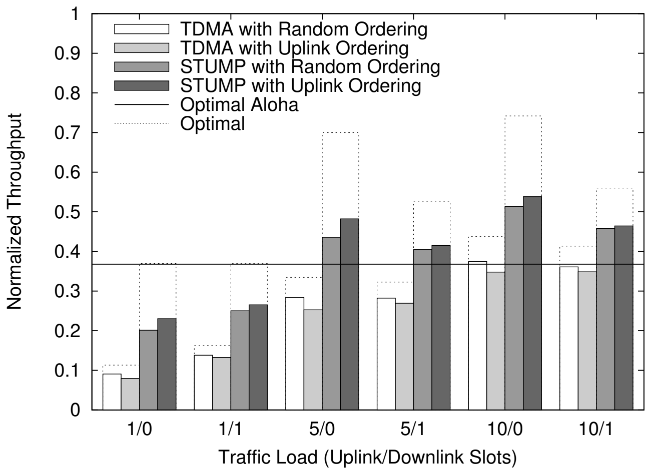

Fig. 8. Normalized Throughput as Traffic Load Varies

cases and performed better than Aloha except for very small traffic loads.

- STUMP handles synchronization error gracefully, losing less than 10% of its performance for synchronization errors up to 0.5s. Tolerating high synchronization error allows the network to use synchronization protocols that consume lower energy or have lower overhead.
- Deriving a schedule for STUMP can be done using distributed or centralized algorithms. We presented two distributed algorithms and evaluated them through simulation, giving users a choice depending on their application requirements. Users are also free to adapt ordering algorithms for their particular need, which can be easily integrated into the work presented here. Additionally, we presented two centralized algorithms that can find optimal schedules.
- CDMA provides no benefit to the scheduled protocols studied here for realistic spreading factor values. Beyond a spreading factor of 2, the increase in transmission time negates any benefit achieved by removing TX-RX-TX conflicts. However, this does not eliminate the usefulness of CDMA at the physical layer, where it can decrease some negative underwater channel characteristics.

Several directions exist for future work with STUMP that we plan to pursue. Combining the priority assignment and schedule generation algorithms with other protocols, such as routing or synchronization protocols, would decrease energy overhead. Exploring the effect of schedule generation and routing frequency on network lifetime would provide users with useful guidelines for operation. Finally, many underwater networks contain mobile nodes, so expanding STUMP to support mobility would increase its application space.

REFERENCES

[1] X. Lurton, *An introduction to underwater acoustics*. Praxis Publishing, 2002.

[2] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," in *Proc. ACM WUWNet Conf.*, Sep. 2006, pp. 1–5.

[3] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," in *Proc. ACM WUWNet Conf.*, Sep. 2006, pp. 41–47.

[4] A. Syed, W. Ye, B. Krishnamachari, and J. Heidemann, "Understanding spatio-temporal uncertainty in medium access with ALOHA protocols," in *Proc. ACM WUWNet Conf.*, Sep. 2007, pp. 41–48.

[5] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC protocol for underwater acoustic networks," in *Proc. IEEE Oceans Conf.*, May 2006.

[6] B. Peleato and M. Stojanovic, "A MAC protocol for ad-hoc underwater acoustic sensor networks," in *Proc. ACM WUWNet Conf.*, Sep. 2006, pp. 113–115.

[7] P. Xie and J.-H. Cui, "R-MAC: an energy-efficient MAC protocol for underwater sensor networks," in *Proceedings of the IEEE International Conference on Wireless Algorithms, Systems and Applications (WASA)*, Aug. 2007, pp. 187–195.

[8] X. Guo, M. R. Frater, and M. J. Ryan, "An adaptive propagation-delay-tolerance MAC protocol for underwater acoustic sensor networks," in *Proc. IEEE Oceans Conf.*, Jun. 2007.

[9] J. Yackoski and C.-C. Shen, "UW-FLASHR: Achieving high channel utilization in a time-based acoustic MAC protocol," in *Proc. ACM WUWNet Conf.*, Sep. 2008.

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

[11] 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.

[12] 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.

[13] H.-X. Tan and W. K. G. Seah, "Distributed CDMA-based MAC protocol for underwater sensor networks," in *Proceedings of the IEEE Conference on Local Computer Networks*, Oct. 2007, pp. 26–33.

[14] A. Kebkal, K. Kebkal, and M. Komar, "Data-link protocol for underwater acoustic networks," in *Proc. IEEE Oceans Conf.*, Jun. 2005, pp. 1174–1180.

[15] D. Pompili, T. Melodia, and I. F. Akyildiz, "A distributed CDMA medium access control for underwater acoustic sensor networks," in *Proc. Med-Hoc-Net*, Jun. 2007.

[16] P. Casari, S. Marella, and M. Zorzi, "A comparison of multiple access techniques in clustered underwater acoustic networks," in *Proc. IEEE Oceans Conf.*, Jun. 2007.

[17] F. Salvá-Garau and M. Stojanovic, "Multi-cluster protocol for ad hoc mobile underwater acoustic networks," in *Proc. IEEE Oceans Conf.*, vol. 1, Sep. 2003, pp. 91–98.

[18] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[19] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, Mar. 1999.

[20] M. Stojanovic, "Recent advances in high-speed underwater acoustic communication," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 2, pp. 125–136, Apr. 1996.

[21] A. A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," in *Proc. IEEE InfoCom Conf.*, Apr. 2006, pp. 892–903.

[22] P. Serafini and W. Ukovich, "A mathematical model for periodic scheduling problems," *SIAM Journal on Discrete Mathematics*, vol. 2, no. 4, pp. 550–581, 1989.

[23] P. Djukic and S. Valaee, "Link scheduling for minimum delay in spatial re-use TDMA," in *Proc. IEEE InfoCom Conf.*, May 2007, pp. 28–36.

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

[25] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proc. ACM MobiHoc Conf.*, May 2006, pp. 190–201.

[26] D. P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1991.

[27] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, Aug. 2004.