UW-FLASHR: Achieving High Channel Utilization in a Time-Based Acoustic MAC Protocol*

Justin Yackoski Chien-Chung Shen
Department of Computer and Information Sciences
University of Delaware
Newark, DE, USA
yackoski,cshen@cis.udel.edu

ABSTRACT

Time-based medium access control (MAC) has potential advantages over FDMA and CDMA approaches in terms of hardware simplicity, energy efficiency, and delay. Unfortunately, the channel utilization of existing TDMA and CSMA acoustic MAC protocols is generally low due to the long propagation delays of acoustic signals. In this work, we argue that several ideas taken from RF protocols, including exclusive channel access, are either unnecessary in acoustic networks or must be redefined. We present and evaluate UW-FLASHR, a time-based MAC protocol which does not require centralized control, tight clock synchronization, or accurate propagation delay estimation. We demonstrate that UW-FLASHR can achieve significantly higher channel utilization than the maximum utilization possible with existing time-based exclusive access MAC protocols, particularly when the ratio of propagation delay to transmission delay is high or data payloads are small.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Design, Performance

Keywords

Underwater Acoustic Networks, Medium Access Control

1. INTRODUCTION

In RF networks, the maximum propagation delay between a transmitter and any receiver it may interfere with, t_{prop}, is generally several orders of magnitude smaller than the transmission delay, t_{DATA}. As a result, concurrent transmissions by nearby nodes on the same channel nearly always collide. A guard time or inter-frame space can be added between all transmissions that resolves such collisions (with some exceptions) by ensuring a given transmitter and its receiver(s) have *exclusive access* to the channel for a duration equal to $t_{prop} + t_{DATA}$. This exclusive access constraint prohibits use of the channel by other nodes during the time period when the transmitter's signal is still propagating through the medium, under the assumption that a collision will likely result. For RF networks, this solution is simple and, since t_{prop} is tiny compared to t_{DATA}, the maximum channel utilization of such an exclusive access MAC protocol[1] approaches 100% as given by:

$$\frac{t_{DATA}}{t_{DATA} + t_{prop}} \tag{1}$$

Given that only bit rates below approximately 50 kbps are currently achievable (due to absorption, noise, multipath, and other factors) and that the speed of sound in water is a relatively sluggish 1500 m/s, the situation for underwater acoustic networks (UWANs) is much different than for RF networks. Here, t_{prop} can be equal to or even exceed t_{DATA} by an order of magnitude or more, leading to low utilization. In this work we focus on such situations where the ratio of t_{prop} to t_{DATA} is high.

For example, consider the channel utilization of the CSMA protocol shown in Fig. 1. We assume 50 byte RTS and CTS frames, 200 byte DATA frame, no ACK, 40 kbps sending rate, 500 m between the sender and receiver, and a wait of t_{prop} between when the CTS is received and DATA is sent to avoid collisions. If an RTS and CTS is used, only 60 ms out of 1392 ms is spent transmitting (including transmitting the RTS and CTS), giving about 4% channel utilization. If the RTS and CTS is omitted, 40 ms out of 373 ms is spent transmitting, still giving less than 11% utilization. In a TDMA protocol, the utilization still is below 11% due to the use of guard times, equal in size to t_{prop}, between each transmission. While the assumed values may be towards the extreme end for some applications, it is clear from this example that the channel utilization of an exclusive access MAC protocol can be poor in UWANs.

In addition to causing poor channel utilization, exclusive access is actually not necessary for collisions to be avoided in UWANs. Instead, the reception of a transmission by the intended receiver must only be separated in time from the reception of any interfering signals by that receiver. Consider the topology in Fig. 2(a) where t_{prop} is shown next to each link and t_{DATA} is 1 time unit. In such a topology, the transmission schedule shown in Fig. 2(b) results in no collisions. 10 frames are sent and received within a schedule of length 9 (plus 8 very short inter-frame spaces between each time slot), resulting in "channel utilization" above 100%. Since channel utilization can not actually exceed 100%, a new definition of

*This work is supported in part by the National Science Foundation under grant CNS-0721361.

[1]Note that Equation 1 is not a fundamental limitation of the channel, but arises from the exclusive access constraint.

WuWNet'08, September 15, 2008, San Francisco, California, USA.

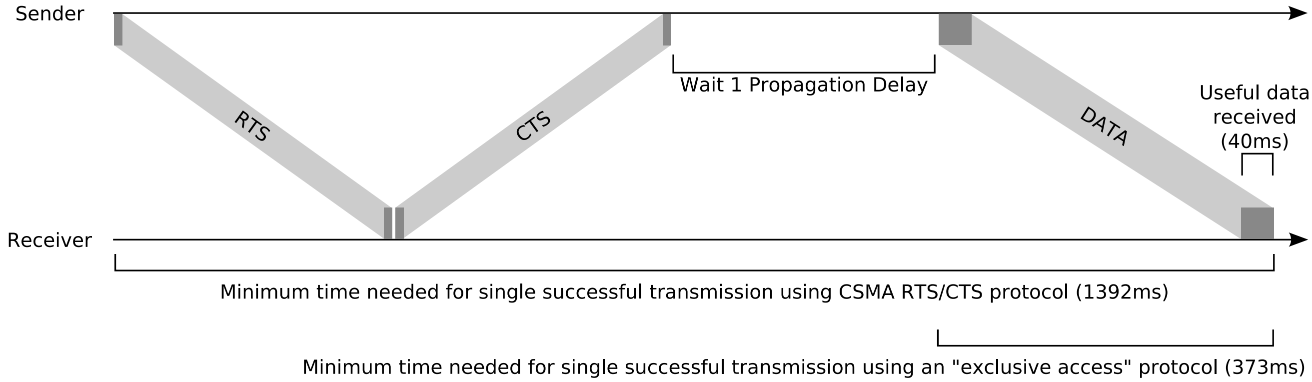
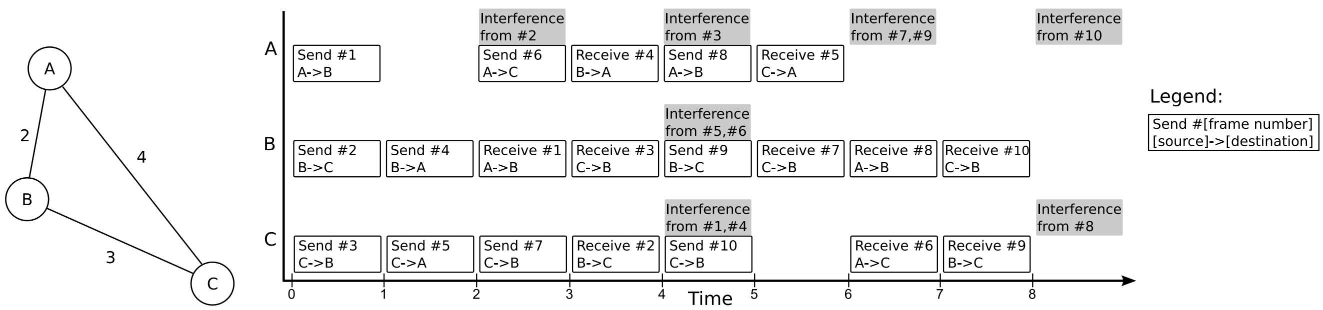


Figure 1: Utilization of existing time-based acoustic protocols. Times are drawn to scale.



(a) (b)

Figure 2: (a) Example topology and propagation delays and (b) one possible collision-free transmission schedule.

channel utilization for UWANs may be needed which considers the fact that interference can be "stacked," for example where node B simultaneously receives interference from frames #5 (sent by node C) and #6 (sent by node A).

Unlike in an RF channel, in an acoustic channel it can sometimes be possible for: (1) two nodes to transmit concurrently (e.g. frames #1 and #2), (2) a node to transmit while receiving interference from another frame intended for another node (e.g. frame #6 is sent while node A receives interference from frame #2), and (3) two nodes to receive two different frames at the same time (e.g. frame #4 received by node A and frame #3 received by node B), *all without any collisions*. Note that such concurrent actions are not possible as a general rule in UWANs, but only in specific situations such as Figure 2 where the propagation delays allows nodes to receive the transmitted signals at non-conflicting times.

A real network will of course neither have such convenient values nor such precise clock synchronization, however this illustrates that a high t_{prop} to t_{DATA} ratio itself does not lead to poor channel utilization. Instead, the true cause is the exclusive access constraint carried over from RF networks, which this example shows to be unnecessarily restrictive. This constraint actually is overly restrictive in both RF and acoustic networks, but in RF networks its negligible cost and simplicity make it useful.

Thus we are motivated to investigate the practicality of a time-based acoustic MAC protocol with the following characteristics:

- **No exclusive access constraint, and thus not subject to Equation 1,**
- **No precise clock synchronization,**
- **No knowledge of propagation delays,**
- **No large guard times between transmissions, and**
- **Completely decentralized operation.**

In the remainder of this paper, we review existing work in Section 2, detail the issues faced by a TDMA-like protocol and describe how the UW-FLASHR protocol overcomes them in Section 3, evaluate the performance of UW-FLASHR in Section 4, and conclude in Section 5.

2. RELATED WORK

Of the recent time-based acoustic MAC protocols, some explicitly enforce the exclusive access constraint via handshaking [4]. Others are implicitly affected by this constraint due to requiring a low duty cycle below the utilization given by Equation 1 [3]. Some enforce the requirement that a transmitter wait one propagation delay before sending a transmission, for example due to a contention round of length equal to one propagation delay [9], or only enforce exclusive access during the DATA frame transmission [2]. A multi-channel protocol such as [10] reduces the probability of collision, but either each channel must be sized such that it is fully utilized by the flow assigned to it (causing high delays and possibly high energy consumption), or the exclusive access constraint must be imposed on a per-channel basis.

[7] and [5] describe more detailed versions of Equation 1 which additionally consider the bit error rate and other factors. The optimal size of a given transmission (adjusted by adapting the size of each packet at higher layers and/or aggregating packets together) is then calculated. This is a valid approach, however its usefulness is limited in scenarios with small packets (e.g. control instructions, status updates, etc.) where the traffic patterns and/or delay constraints prevent such size adjustment. In addition, these schemes

perform this optimization under the assumption that an exclusive access MAC is used. UW-FLASHR can be combined with such schemes, however the cumulative benefit is strongly dependent on the traffic patterns and application constraints. Thus, we focus on the separate problem of improving channel utilization with the assumption that any such optimization has already been performed at higher layers and that further aggregation is not possible.

[1] describes a linear programming formulation of acoustic link scheduling (i.e. TDMA) without the exclusive access constraint. This scheme requires centralized control, precise clock synchronization, and precise knowledge of the propagation delays between all nodes, significant constraints which UW-FLASHR avoids due to the difficulty of meeting them in practice.

We are not aware of any time-based MAC protocols for UWANs which are not subject to the exclusive access constraint in some way *and* which are not significantly impacted by the assumed precision with which clock synchronization and/or propagation delay estimation can be performed. Thus, even for time-based protocols which have adapted to UWANs, Equation 1 is currently an upper bound on channel utilization.

3. A TIME-BASED ACOUSTIC MAC

In this section, we discuss the challenges which face a TDMA-like MAC protocol in UWANs. We show that while it may be difficult or even impossible to achieve precise clock synchronization or precisely estimate propagation delays, it is still possible for nodes to coordinate the timing of their actions using the TDMA-like UnderWater FLASHR (UW-FLASHR) protocol. UW-FLASHR is loosely based on the FLASHR protocol [11] for RF networks, with the mechanisms described in this section added in order to compensate for the propagation delays and other problems faced in UWANs. The major operations of UW-FLASHR are illustrated in Figures 3 and 4, and are detailed further within this section.

Figure 3: Pseudo-code for UW-FLASHR Protocol

```
loop
  if Send invoked for DATA/ACK to node x then
    if Have available time slot for node x then
      Send frame in existing time slot
    else
      Acquire_New_Slot()    /* Fig. 4 */
    end if
  else if Receive DATA or ACK for node x then
    if Δ set indicating new slot request then
      if I am x and no collision will occur then
        Send RESPONSE approving request
      else if Collision will occur then
        Send RESPONSE rejecting request
      end if
    else {DATA or ACK sent in established slot}
      Record time received as bad time for me
      if I am node x then
        Update expiration time for any slots confirmed by
        sender
        Record current offset field to confirm receipt to
        sender
        if DATA frame then
          Invoke Send for ACK to sender
        else {ACK FRAME}
          Confirm receipt of ACK in next DATA sent to
          sender
        end if
      end if
    end if
  else if Receive RESPONSE frame for me then
    if RESPONSE approving request then
      if requestState = unknown then
        requestState ← approved
      end if
    else {RESPONSE rejecting request}
      requestState ← rejected
      if rejecter = x then
        Add bad time(s) provided to the bad time list for
        node x
      else {rejecter = another node}
        Add bad time(s) provided to the global bad time list
      end if
    end if
  else if Receive carrier sense notification then
    If in established portion, record sensed time as bad time
    for me
  end if
end loop
```

3.1 Cycle Boundaries

Since TDMA divides time into fixed-size cycles, it is necessary for adjacent nodes using such a protocol to agree upon the cycle boundaries (i.e., synchronize their clocks). [8] shows that relatively precise clock synchronization can be achieved if the propagation delay is predictable and static for short periods of time. We assume that clock synchronization between neighbors can be achieved to within approximately the sum of the prediction error and short-term variability of the propagation delay between neighboring nodes. We expect this sum to be roughly an order of magnitude smaller than the propagation delay itself.

We have implemented our own simple clock synchronization mechanism as part of UW-FLASHR which uses information piggybacked on transmissions to achieve rough clock synchronization. In our implementation, nodes do not estimate propagation delay and each node stops further attempting to explicitly synchronize with a neighbor if the node believes that its clock and the neighbor's clock are currently synchronized to within 1 s, although much closer synchronization below 50 ms is often achieved. We do not describe this protocol here due to space constraints and because work such as [8] shows the clock synchronization we require is feasible.

In UW-FLASHR, we divide each cycle into two portions, an *experimental portion* and an *established portion* (described in the next sub-section). The primary concern in determining cycle boundaries is the requirement that transmissions made in one portion of the cycle do not interfere with transmissions made in the other portion. Thus, we set the thickness of the cycle boundaries equal to t_{prop}. Since we expect the clock synchronization error to be at least an order of magnitude smaller than t_{prop}, which is itself an overestimate, we assume this thickness alone is sufficient to avoid interference between cycle portions. Note that UW-FLASHR wastes $2t_{prop}$ *per cycle* (where each cycle contains many transmissions) while existing schemes waste t_{prop} *per transmission*.

Apart from cycle boundary agreement, UW-FLASHR does not assume that the propagation delay can be determined or that clocks are synchronized at all when scheduling transmissions. Thus, in cases where clock synchronization and propagation delay estimation to the precision assumed above isn't possible, the only impact is that either the thickness of these boundaries must be increased (at worst to $2t_{prop}$ based on [8]), or transmissions received at times

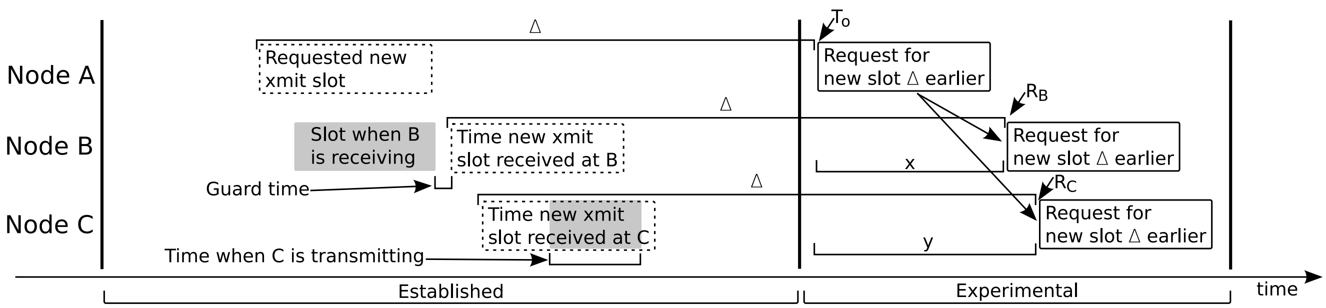


Figure 5: Example where node A transmits a request for a new slot. Times are not drawn to scale.

Figure 4: Pseudo-code for Acquire_New_Slot() function

```
while already attempting a new slot do
    Wait until slot is acquired or rejected
end while
x ← intended recipient of new slot
attempts ← 0
requestState ← unknown
Select candidate slot that does not overlap with entries in global
bad time list or in bad time list for node x
repeat
    At random time in experimental portion, send frame with Δ
    set to indicate requested slot
    Process any RESPONSE frames or Aborts
until (attempts > minAttemptsIfApproved
AND requestState = approved) OR attempts >
maxAttemptsBeforeFailure OR requestState =
rejected
if requestState ≠ approved then
    Go to beginning of function
else {approval was received}
    Begin using newly acquired slot
end if
```

near the boundaries may be subjected to interference. As described in Section 3.4, UW-FLASHR is able to adapt to such unexpected interference.

3.2 Scheduling Transmissions

In order to avoid collisions, a node must not receive interference concurrently with the reception of a signal intended for itself. At first glance, this seems difficult to determine with any precision unless the clocks of some or all of the nodes are synchronized or the propagation delays between all nodes are known. Fortunately, neither is required if *relative* times are used to coordinate and if nodes wait to coordinate until after observing (but not measuring) the propagation delays. Figure 5 shows an example of this process.

We divide the cycle into: (1) a small experimental portion, during which control frames and requests to acquire new transmission time slots within the established portion are made, and (2) a much larger established portion, during which nodes may only transmit in already acquired time slots. We do not divide the cycle into even time slots, instead each node's time slots may start at an arbitrary point within the established portion, have an arbitrary length, and possibly overlap with the time slots of other nodes (as long as no collision occurs). Over time, the transmission schedule in the established portion is gradually built until all nodes are satisfied or no free times are available. Since UW-FLASHR is a TDMA-like protocol, we assume that each application generates isochronous data, that is data of a constant size generated at a constant interval. Note that each application may select a distinct size and interval, although the cycle size must be a common multiple of all intervals used.

Transmissions in the established period, which comprises the majority of the cycle, have a high success rate due to a lack of interference and can be tightly packed. Guard times, equal to the expected short term clock drift and *variation* of t_{prop} (not t_{prop} itself), are only needed between the times when a node is receiving a frame for which it is the intended recipient and when interfering transmissions are received. Since the propagation delays between nodes may be significantly different, these guard times are enforced separately per node instead of on a global basis to allow much higher channel utilization. Due to the lack of clock synchronization, the header of all frames includes T_o, the *current offset* in the cycle (based on the transmitter's clock) at which the frame was transmitted. This allows nodes to communicate using relative times as described below.

When a node requires a new time slot in which to send data, the node requests a new slot by sending a DATA frame at a random time in the experimental portion of each of several consecutive cycles. The repetition increases the chance the all neighboring nodes will hear the request and respond if needed. These transmissions serve both to request a particular new slot and to allow the node to send some data until a new slot is acquired. Any such request frame header contains a value Δ which indicates the difference between the time when the request was sent and the start time of the slot being requested. Since a node sends multiple requests for the same time slot at different random times, responses to the request refer to the offset of the requested slot based on the requester's clock by subtracting Δ from T_o. Nodes that hear the request determine the time (based on their own clock) when a transmission in the requested slot will be received by subtracting Δ from the current offset in the cycle (again, based on their own clock) when the request is received. The duration of the slot requested (and any interference it will cause due to reflections, etc.) can be inferred from the duration of the DATA (request) frame.

Each node checks the requested slot against the node's list of times (based on the node's own clock) when the node is sending and receiving to determine if a conflict exists and respond appropriately. Due to the unknown propagation delays, each node hearing the request must individually check for conflicts with only its own receiving times. RESPONSE frames are then sent, at a random time in the experimental portion of the cycle, based on the following rules:

1. The intended recipient of the new slot sends a (positive) RESPONSE frame approving the request if no conflict exists

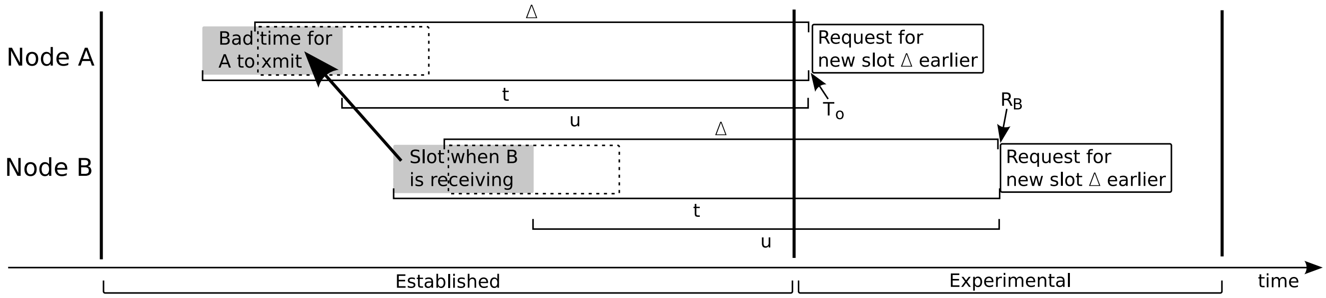


Figure 6: Example of exchanging bad time information.

2. Any node, including the intended recipient, that determines that a conflict exists at itself sends a (negative) RESPONSE frame rejecting the request.

This process is illustrated in Figure 5 where node A requests a new slot. Nodes B and C hear this request frame after a propagation delay of x and y, respectively. They each subtract Δ from the time when they receive the frame (R_B and R_C, respectively). Using *only* Δ and R_B or R_C (x and y remain unknown), nodes B and C can determine exactly when a transmission by node A in the requested slot would arrive. Here, no collision will occur at node C because node C is transmitting (not receiving) during the time it will receive interference from the proposed slot. No collision will occur at node B even though A would be transmitting while B is receiving, because the propagation delay between A and B causes B's reception of A's transmission to occur at least one guard time after the time at which B finishes receiving a transmission from another node. Node B then sends a RESPONSE frame approving the request in the next experimental portion (not shown), while node C takes no action since there is no conflict.

3.3 Finding A Time Slot

The mechanism described above allows a node to request a particular time slot, however the time slot may cause collisions with one or more existing time slots. In fact, due to the unknown propagation delays, in an even slightly congested network it may be difficult for a node to randomly guess a time slot to request which does not cause any collisions. A method of narrowing the search to good candidates is therefore needed.

To allow this, each node maintains a per-neighbor list containing bad times during which the neighbor has indicated it receives interference and so cannot receive from the node. Each node maintains a similar list for itself of times when it cannot receive due to interference. Separate lists per neighbor are necessary because: (1) the propagation delays between nodes are different and unknown, and (2) nodes clocks are not synchronized and only relative time information is known as described in the previous subsection. Each node additionally maintains a "global" list of bad times in which a transmission by the node would interfere with one of its neighbor's reception of a transmission in an existing time slot, and thus can not be used by the node to transmit regardless of the intended recipient. A node may only request a new slot which does not overlap with both the global bad time list and the bad time list for the intended recipient. Bad times expire after a period of time.

When a node sends a RESPONSE frame rejecting a request for a new time slot, the frame also includes one or more ranges of offsets adjacent to the requested offset which would cause a collision. The rejecter knows T_o and the current offset according to the rejecter's clock at which the request was received. Thus, the rejecter can precisely refer to offsets in the cycle with respect to the requesting node's clock using relative times. If the rejecter is the intended recipient of the requested new slot, the requesting node adds any bad time ranges on the RESPONSE to its list of bad times for that neighbor. Otherwise, the requesting node adds the ranges to its global list of bad times, since the interference from any transmission by the requesting node during this time range would cause a collision at the rejecter.

This process is illustrated in Figure 6. Here, R_B is the current offset, according to node B's clock, at which the request was received. t and u are calculated by subtracting from R_B the start and end offsets, respectively, of the gray slot when node B is already receiving. Node B can then notify node A that transmissions sent during the time range from $T_o - t$ to $T_o - u$, according to node A's clock, will cause a collision. Again, no knowledge of propagation delay or clock synchronization is needed.

3.4 Interference and Aborting Slots

The process of requesting a new time slot in no way guarantees that all nodes have a chance to reject new time slots that will cause a collision with an existing transmission they are receiving. The requests may not be received either due to collisions between signals sent during the experimental portion, or due to the signal strength between two nodes being too weak to decode the request frame but strong enough to cause interference if the requested slot is used. Similarly, a RESPONSE frame rejecting a request may not be heard by the requester. While nodes can adapt by simply acquiring a new slot if such interference occurs, the rate at which this occurs must be kept low to ensure network stability.

When a receiver detects that a collision has occurred between the reception of a transmission sent in an existing slot and some other slot (by consecutively being unable to successfully decode a frame received in the slot), the node notifies its neighbors that the offending slot must be aborted and no longer used by the offending node. This *abort* is included in the headers of all frames sent by the originator of the abort, to ensure as many neighbors as possible receive it. Since the offender may not hear the initial abort, nodes which hear the abort in turn spread it throughout the network until the abort's hops-to-live has expired. Since neither the offender or the exact time of the offending slot can be easily determined, any node which receives an abort that is within the maximum possible propagation delay of one of its *newly* acquired or requested slots must immediately discontinue use of the slot and add the slot to its global bad time list. Any node that does hear a request for a new slot and sends a RESPONSE frame rejecting the request also begins sending an abort request for the requested time slot, to increase the probability that the new slot is not actually used in the first place.

3.5 Acknowledgments

Many time-based acoustic MAC protocols do not use acknowledgments since their use in conjunction with the exclusive access constraint results in a channel utilization of

$$\frac{t_{DATA} + t_{ACK}}{t_{DATA} + t_{ACK} + 2 * t_{prop}} \quad (2)$$

where t_{DATA} is the DATA frame transmission delay and t_{ACK} is the ACK frame transmission delay. For our purposes, we include the entire duration (including headers, preamble, etc.) of all frames transmitted when calculating channel utilization, so that the ideal channel utilization is 100%, and is not directly affected by factors (payload size, headers, etc.) that are out of the MAC protocol's control.

Since ACK frames are often much smaller than DATA frames, channel utilization with ACKs can be much worse than without (given by Equation 1) if t_{prop} is large. Since ACKs are useful in many applications, we incorporate their use into UW-FLASHR. By scheduling ACK transmissions using the same request procedure already described for DATA transmissions, ACKs can be supported without significantly impacting channel utilization. In fact, since ACKs are small, a suitable time slot can often be more easily found for ACKs than for DATA frames.

ACKs serve to confirm to the sender that transmissions in a given slot are working. The receiver precisely identifies the acknowledged slot to the transmitter by including the T_o value from the corresponding DATA frame that was received. Similarly, the receipt of an ACK from a given node is also confirmed via header fields in the next DATA frame sent to that node.

If ACKs are not required for every frame, receivers could instead periodically confirm to their corresponding senders that the slots being used are working by sending ACKs at random times in the experimental portion of the cycle. ACKs could also be aggregated together to improve channel utilization (as could DATA frames). However, we do not consider these cases as part of this work, and instead focus on the more difficult assumption that all frames must be sent individually.

4. EVALUATION

We compare the measured channel utilization of UW-FLASHR with the maximum channel utilization of existing time-based exclusive access acoustic MAC protocols. While UW-FLASHR uses ACKs, most existing protocols do not, thus we show this maximum for existing protocols both with ACKs (using Equation 1) for a fair comparison, and without ACKs (using Equation 2) for additional comparison. We compare against these curves instead of any particular MAC protocol since, for reasons discussed previously, these curves represent the theoretical best case performance of all existing time-based exclusive access protocols. Note that we do not make any claims whether existing protocols can actually achieve these limits in practice, while UW-FLASHR's performance is evaluated experimentally, so the comparison may be slanted slightly against UW-FLASHR.

To evaluate UW-FLASHR, we use QualNet 3.7 [6] modified to simulate an acoustic channel. We use spherical path loss and Thorp attenuation, regardless of the distance between nodes to isolate propagation delay and eliminate any changes in interference patterns caused by scaling the terrain dimensions. In all scenarios, we randomly place 10 nodes in a flat, square terrain and use a transmission range equal to the maximum possible distance between nodes. Cycle boundaries are randomly initialized at each node. We generate CBR traffic between randomly selected node pairs. We determine channel utilization by measuring the maximum total duration of the DATA and ACK frames received by their intended recipients in any one cycle during the simulation, and divide this value by the cycle size. Results shown for UW-FLASHR are the average of 30 trials with 95% confidence intervals. Comparison lines are calculated using Equations 1 and 2. Other parameters are shown in Table 1.

Table 1: Simulation parameters used.

Parameter	Value
Data Rate	15 kbps
Propagation Speed	1500 m/s
t_{ACK}	32 ms
t_{DATA} for 50-byte payload	76 ms
t_{DATA} for 1000-byte payload	582 ms
Cycle Size	10 s
Experimental Portion Size	1500 ms
Guard Time	2 ms
Bad Time Expiration	60 cycles
Approved Slot Expiration	200 cycles
Abort Hops-to-live	8
minAttemptsIfApproved	5
maxAttemptsBeforeFailure	100

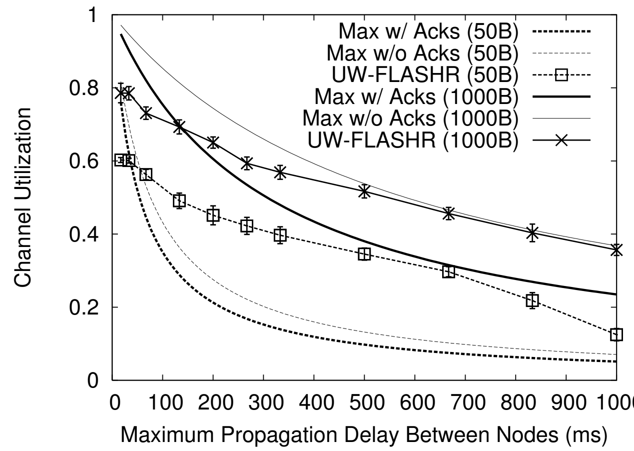


Figure 7: Utilization versus maximum propagation delay.

4.1 Effects of Propagation Delay

Figure 7 shows the impact of varying the terrain dimensions (and thus t_{prop}) between 25 m and 1500 m (17 ms and 1000 ms) when 50-byte and 1000-byte data payloads are used. For the smaller 50-byte payloads ($t_{DATA} = 76$ ms), UW-FLASHR achieves approximately twice the maximum of existing protocols except when t_{prop} is below t_{DATA}. This is due to: (1) the 1500 ms size of the experimental portion which alone makes the maximum utilization achievable by UW-FLASHR 85%, (2) the fragmentation of free times caused by the varying propagation delays (i.e. free times created by scheduling decisions that are too small to be used for a transmission), and (3) the thickness of the boundaries between the experimental and established portions of the cycle. In reality, fragmentation along with medium acquisition (e.g. carrier sensing) prevent any time-based protocol from actually achieving utilization near the 100% suggested by Equations 1 and 2 even when t_{prop} is negligible.

For 1000-byte payloads ($t_{DATA} = 582$ ms), UW-FLASHR achieves

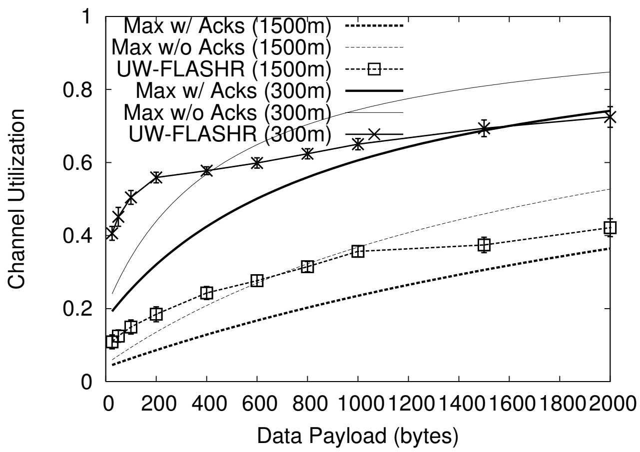


Figure 8: Utilization versus size of data payload.

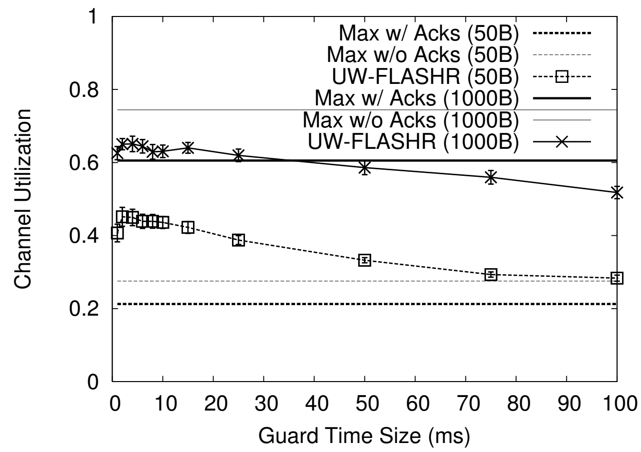


Figure 9: Utilization versus size of guard times.

nearly 80% utilization when t_{prop} is low, close to UW-FLASHR's limit of 85%. As t_{prop} increases, the utilization of UW-FLASHR declines primarily due to fragmentation, however UW-FLASHR maintains higher utilization than the maximum of existing protocols with ACKs, and approaches (coincidentally) that of protocols without ACKs.

Note that although UW-FLASHR's performance is significantly affected by the fragmentation of free times, particularly when both t_{DATA} and t_{prop} are high, we do not mean to imply that this is a flaw in UW-FLASHR. Instead, the scheduling constraints imposed by the acoustic medium as t_{prop} increases significantly limit the number of non-colliding transmissions that can occur in a given period of time, even with perfect knowledge. While some improvement could likely be made by using a scheduling heuristic or aggregating frames, most of the improvement can only be achieved by changing the actual load conditions and/or network topology to improve these constraints. In other words, scenarios exist where *no* time-based acoustic MAC protocol can achieve high utilization.

4.2 Effects of Payload Size

Figure 8 shows the effect of varying the data payload size on channel utilization in 300 m by 300 m and 1500 m by 1500 m square terrains. This graph show that UW-FLASHR's key strength over existing protocols is when 400 byte or smaller data payloads are used. With payloads below 400 bytes, UW-FLASHR achieves utilization higher than the maximum of existing protocols even without ACKs, and approximately twice the max for payloads of 100 bytes or less, regardless of t_{prop}. While small payloads are often avoided in evaluating acoustic MAC protocols since utilization necessarily is lower (as this graph shows), this is not always possible in real deployments. Intuitively, small frames can be more easily scheduled without collisions because their size allows them to be more easily fit in between the bad times in the transmission schedule, as long as the exclusive access constraint is not enforced.

4.3 Effects of Short-Term Variation

In Figure 9 we evaluate UW-FLASHR under a range of guard times in a 300 m by 300 m terrain. This allows us to consider the total impact of short-term variation due to sources such as: (1) small variations in propagation delay due to fluctuations in the medium, (2) variation in propagation delay and interference due to node mobility, and (3) variable bit rate (VBR) data due to compression or other factors. Note that while (2) and (3) may be significant in some

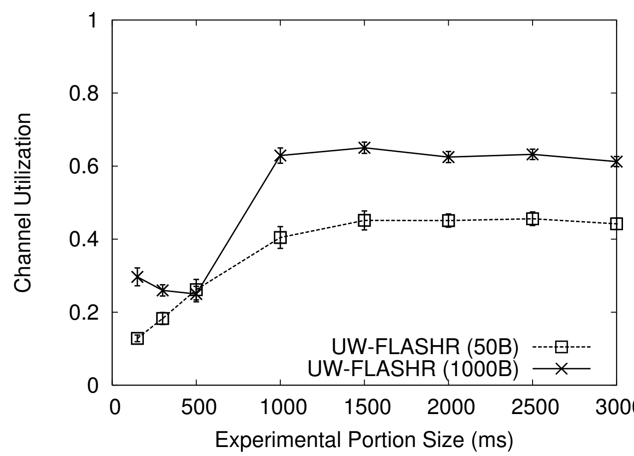


Figure 10: Utilization versus size of experimental portion.

deployments, UW-FLASHR's sensitivity to them is not unique, and any TDMA, CDMA, or FDMA protocol will be affected similarly. Only CSMA protocols are unaffected by these variations since no forward-looking assumptions are made.

Guard times must be sized to accommodate the total expected range of these short-term variations. Since existing TDMA and CSMA protocols do not use a guard time equal to the *variation* in propagation delay, we assume their maximum is unaffected. UW-FLASHR's utilization is relatively unaffected when the guard time is below 25 ms since the non-uniformity in propagation delays between nodes already prevent transmissions from being perfectly packed in time. Above 25 ms, the guard times are large enough to exclude small gaps where frames, specifically ACK frames ($t_{ACK} = 32$ ms), previously could have fit. UW-FLASHR is therefore suitable for a range of scenarios where the total short-term variation is moderate, but its channel utilization benefits over CSMA approaches may be overcome by the large guard times needed in highly variable environments. Even in such dynamic environments, UW-FLASHR remains a viable choice due to the potential energy savings of a TDMA-like MAC.

4.4 Effects of Size of Experimental Portion

Figure 10 shows how the size of the experimental portion affects channel utilization. Since the experimental portion is essentially wasted, a small experimental portion is desirable. However, the graph shows that when the experimental portion is too small (below 1000 ms), nodes are unable to request new time slots or respond to requests, particularly when the data payloads (which determine the size of the request frame) are large since the transmission time of a 1000 B packet is 582 ms. The rate at which nodes must request new slots, the data payload size, and the network density, all impact the optimal size for the experimental portion.

5. CONCLUSION

We have presented UW-FLASHR, a time-based acoustic MAC protocol which meets the five goals outlined in Section 1, allowing for higher channel utilization. The large propagation delays and difficulty in precisely synchronizing node clocks present challenges to scheduling transmissions in UWANs. However, we have shown that UW-FLASHR can overcome these and exceed the maximum utilization of protocols which use exclusive access constraint carried over from RF networks. In particular, when data payloads are small, a scenario where existing time-based acoustic MAC protocols suffer from poor performance, we have demonstrated a twofold increase in channel utilization.

In addition to improving channel utilization, the TDMA-like nature of UW-FLASHR also allows nodes to more efficiently use energy. UW-FLASHR's scheduling ensures that few collisions will occur, except in the experimental portion which nodes only use when requesting a new time slot. Further, the times when a node will receive in the established portion are known, allowing nodes to sleep during the majority of the established portion if desired. When needed, nodes can stay awake during some or all of the established portion to update their own bad times list. Other techniques can be used to minimize listening during the experimental portion, such as decreasing the size of the experimental portion after the network has stabilized or only listening during every few experimental portions since new slot requests are repeated multiple times.

6. REFERENCES

[1] Leonardo Badia, Michele Mastrogiovanni, Chiara Petrioli, Stamatis Stefanakos, and Michele Zorzi. An optimization framework for joint sensor deployment, link scheduling and routing in underwater sensor networks. In *Proc. of WUWNet*, Los Angeles, CA, USA, September 2006.

[2] Xiaoxing Gui, Michael R. Frater, and Michael J. Ryan. An adaptive propagation-delay-tolerant mac protocol for underwater acoustic sensor networks. In *Proc. of IEEE OCEANS*, Aberdeen, Scotland, June 2007.

[3] Min K. Park and Volkan Rodoplu. UWAN-MAC: An energy-efficient MAC protocol for underwater acoustic wireless sensor networks. *IEEE Journal of Oceanic Engineering*, 32(3):710–720, July 2007.

[4] Borja Peleato and Milica Stojanovic. A MAC protocol for ad-hoc underwater acoustic sensor networks. In *Proc. of WUWNet*, Los Angeles, CA, USA, September 2006.

[5] Dario Pompili, Tommaso Melodia, and Ian F. Akyildiz. Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks. In *Proc. of MobiCom*, pages 298–309, Los Angeles, CA, USA, September 2006.

[6] Scalable Network Technologies, Inc. QualNet Simulator. http://www.scalable-networks.com.

[7] Milica Stojanovic. Optimization of a data link protocol for an underwater acoustic channel. In *Proc. of IEEE OCEANS*, pages 68–73, Brest, France, June 2005.

[8] Affan Syed and John Heidemann. Time synchronization for high latency acoustic networks. In *Proc. of IEEE Infocom*, pages 1–12, Barcelona, Spain, April 2006.

[9] Affan Syed, W. Ye, and John Heidemann. T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks. In *Proc. of IEEE Infocom*, pages 231–235, Phoenix, AZ, USA, April 2008.

[10] Hwee-Xian Tan and Winston K.G. Seah. Distributed CDMA-based MAC protocol for underwater sensor networks. In *Proc. of 32nd IEEE Conf. on Local Computer Networks*, Dublin, Ireland, October 2007.

[11] Justin Yackoski and Chien-Chung Shen. Managing delay and jitter in mesh networks through path-aware distributed transmission scheduling. *ACM Mobile Computing and Communications Review*, 11(2):49–50, April 2007.