MAC PROTOCOLS FOR MONITORING AND EVENT DETECTION IN UNDERWATER NETWORKS EMPLOYING A FH-BFSK PHYSICAL LAYER

Federico Guerra[a], Paolo Casari[a], Michele Zorzi[a]

[a]All authors are with the Department of Information Engineering, University of Padova via Gradenigo 6/B, 35131 Padova, Italy. E-mail: {firstname.surname}@dei.unipd.it.

The corresponding author is Paolo Casari, Department of Information Engineering, University of Padova via Gradenigo 6/B, 35131 Padova, Italy. E-mail: casarip@dei.unipd.it

Abstract: *Underwater acoustic networks of fixed and autonomous nodes can be a very valuable tool in a number of situations, from environmental monitoring to emergency scenarios (e.g., ships in distress). In this paper, we compare the performance of some MAC protocols for underwater networks in typical scenarios. We consider random access protocols, which provide sufficiently high performance in case of low traffic, and then compare random access with handshake-based access, which achieves better coordination among nodes, at the price of greater control overhead. We consider both periodic traffic and event-driven traffic, and provide insight about which scheme achieves the best performance in terms of relevant network metrics such as throughput, error rate and overhead. In our evaluation, we assume the network protocols to work over a low-rate FH-BFSK-based physical layer, a simple technique that can be easily implemented, e.g., to work as a common PHY for different modem hardware.*

Keywords: *Underwater acoustic networks, FH-BFSK, MAC protocols, random access, handshaking, performance evaluation, simulation, periodic traffic, event-driven traffic.*

1. INTRODUCTION AND RELATED WORK

The growing interest in underwater acoustic networking covers many aspects, from channel access, to routing and topology control [1]-[4] and is well understandable in light of the wealth of applications that could be supported by autonomous networks of underwater fixed and mobile nodes. However, while research has mostly focused on medium to large-sized network simulations so far, these scenarios are difficult to realize, mainly due to the very high cost of underwater nodes. Another impairment to large deployments comes from the substantial differences in the hardware sold by different manufacturers, who generally employ proprietary modulation/coding formats and undisclosed receive algorithms.

In order to harmonize the communication format of underwater nodes, at least for a restricted number of functions, recently simple modulation and coding schemes have been created that could be easily included in standard off-the-shelf hardware. Such schemes may rely, e.g., on Frequency-Hopping Binary Frequency Shift Keying (FH-BFSK), a signalling pattern that offers some resilience to interference, thanks to the different frequency hopping patterns of different transmitted signals, which reduce the probability that two transmissions by different nodes collide, and therefore both packets are lost. Moreover, the FH-BFSK signal is relatively easy to receive using non-coherent detection. Thanks to its simplicity, this transmission format can be supported by a number of hardware implementations. Its main drawback, however, is the very low transmit bit rate, on the order of the tens of bits per second, which limits both its applications and the higher-level protocols that can operate on top of it.

In the present study, we compare three Medium Access Control (MAC) protocols [1], [2], [5], featuring different transmission coordination schemes, from no coordination to full-fledged handshakes. We study the behaviour of these protocol, on top of the previously overviewed physical layer, in the presence of two different applications which must handle periodic and event-driven traffic, respectively; the performance evaluation we carry out aims at providing insight on the access scheme yielding the best performance in terms of throughput, transmit error rate and other relevant network metrics.

2. CONSIDERED PROTOCOLS

In this section, we briefly review the protocols we have chosen for our comparison. The first, ALOHA, is a basic random access protocol, and represents a typical uncoordinated scheme; the second, T-Lohi, is a contention-based access scheme that employs tones to manage contentions; the last one, DACAP is a 4-way handshake-based protocol, where nodes exploit the transmission timings in order to detect possible collisions, and thereby warn the transmitter, or directly refrain from sending packets. We highlight that this form of handshaking yields better coordination but also more overhead than T-Lohi's.

2.1. ALOHA

ALOHA [5] is a simple random access algorithm, whereby a node immediately transmits whenever it has data to send. This means that collisions may take place if two mes-

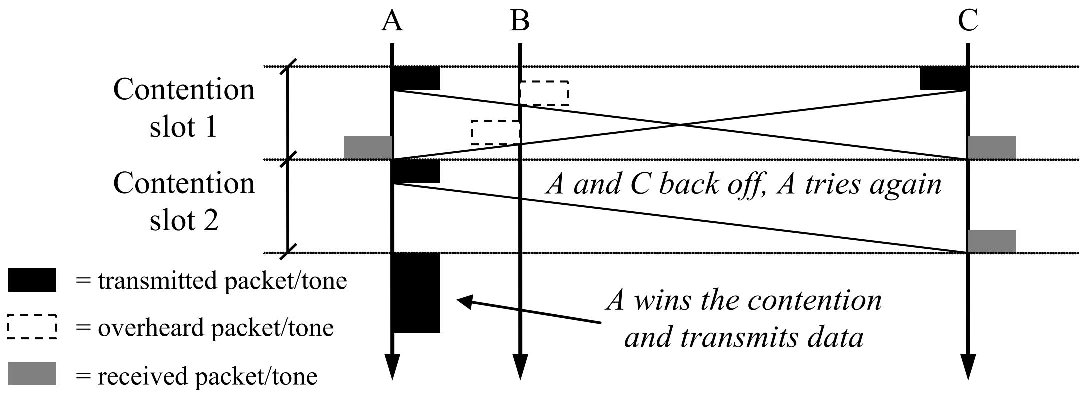

Fig. 1: An example of contention for channel access in T-Lohi.

sages arrive at the receiver at the same time. Standard contention resolution techniques (e.g., random backoff times) are to be applied in this case. ALOHA comes with no specifications as to sending acknowledgment (ACK) messages to confirm correct data reception or not. In case ACKs are used, a backoff policy can be implemented, whereby a terminal refrains from transmitting for a random amount of time if an expected ACK is not received. ALOHA, in general offers poor throughput performance and is very prone to congestion. Nevertheless, it can be a feasible option in the presence of light traffic [3].

2.2. Tone-Lohi

Tone Lohi (T-Lohi) [1] is a reservation based MAC protocol. In T-Lohi, nodes detect and count the number of neighbours simultaneously accessing the channel, and contend through a traffic-adaptive backoff algorithm driven by the estimate of the number of contenders. The slow underwater sound propagation favours this mechanism, as long as the signalling packets have a much lower duration than the propagation delay. Contenders are detected through the use of wakeup tones, which allow the nodes to stay asleep most of the time, thus providing substantial energy savings during the reservation phase. On the other hand, the use of tones requires a specific wake-up tone detector, which should listen to the tone with minimal energy consumption.

To conceptually describe the protocol, it is useful to consider its synchronized version, ST-Lohi. In ST-Lohi, all nodes in the network are aligned to contention slots whose duration is equal to the maximum propagation delay plus the tone length. Any nodes seeking channel access must contend by sending their reservation tones exactly at the beginning of these slots, if they are not restricted by backoff. After sending the tone, a node waits and listens to detect the arrival of other tones for the rest of the contention slot. If it does not hear any other tone, it wins the reservation, and immediately transmits its data. Otherwise, a contention is arbitrated among the multiple nodes trying to reserve the medium. More specifically, the nodes back off for a random number of slots which depends on the number of contenders, and retry at a later time. In order to obtain the number of channel access competitors, the nodes count the received tones, and use this number as their backoff window size. A winner is elected whenever a node is the only one to transmit a tone in its contention slot. An example of such a contention involving two nodes can be seen in Fig. 1.

Note that the long underwater delays are exploited, in that different propagation delays allow tones to be more separated in time at the receivers, and thus help count the number of contenders. When no synchronization can be assumed (e.g., to save signalling efforts or

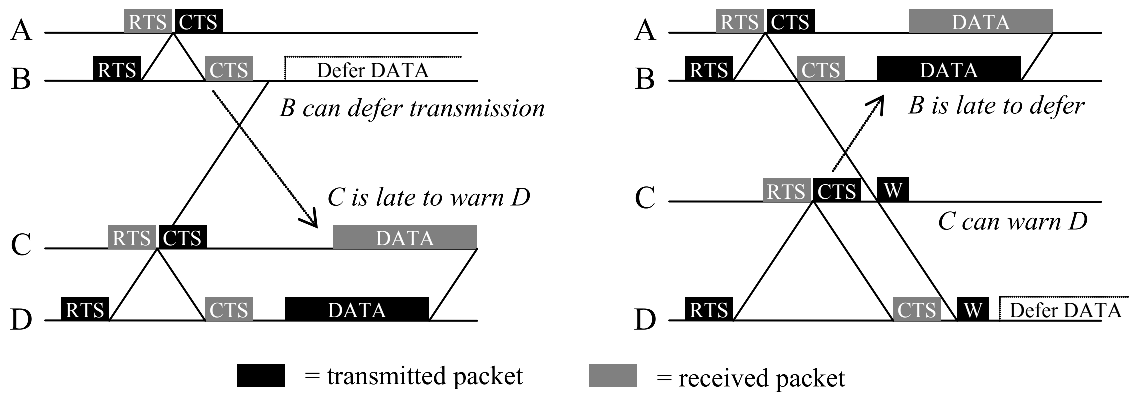

Fig. 2: Two situations where DACAP successfully avoids collisions.

because of the difficulty of synchronizing a multihop network), the timing of tone transmissions can be slightly modified to obtain a conservative (cT-Lohi) and an aggressive (aT-Lohi) version (where contention slots are asynchronous among nodes, and the length of a slot is respectively longer or shorter [1]).

2.3. DACAP

Distance-Aware Collision Avoidance Protocol (DACAP) [2] is a non-synchronized handshake-based access scheme that aims at minimizing the average handshake duration by allowing a node to use different handshake lengths for different receivers. The protocol is specified as follows. The transmitter and receiver notify their intention to set up a link through an RTS/CTS exchange. If, after sending the CTS, the receiver overhears a packet threatening the pending reception, the node sends a very short warning packet to its transmitter. To exploit the advantage granted by this further signalling the sender waits some time before transmitting the data packet. If it overhears a packet meant for some other node or receives a warning from its partner, the sender defers its transmission. These situations are depicted in Fig. 2. In some cases the warning arrives while the node is transmitting the data, and hence is lost because modems are half-duplex. The length of the waiting period is chosen so as to guarantee absence of collisions, and depends on the distance between the nodes, which the sender can learn by measuring the RTS/CTS round-trip time. We note that handshakes only need to avoid collisions from nodes closer than a certain distance, as farther nodes would create little interference, and thus allow the packet to be received in any case. Hence, handshakes between close neighbours can be made short, while those between far apart nodes need to be longer. To achieve a trade-off that maximizes the throughput of a given network, a minimum hand-shake length $t_{\min}$ is predefined for all the nodes. For a network in which most links are close to the transmission range, $t_{\min}$ needs to be nearly twice the maximum propagation delay. When some links are shorter, it can be reduced. Two versions of the protocol can be envisioned: with acknowledgments (ACKs) sent right after receiving a correct data packet, and without ACKs. In the first case, the protocol requires slightly different timings with respect to the second case, in order to accommodate the ACK message [2]. The DACAP protocol (with or without ACK) has been designed to provide a controlled collision environment where multiple data communications can coexist without harming one another.

3. SIMULATION RESULTS

3.1. Network settings and parameters

Our main purpose in the following comparison is to test the applicability of the protocols described before to underwater communications performed using a FH-BFSK modulation with 13 subcarriers. In our implementation, a convolutional code of rate 1/2 is also employed to yield further resilience to bit errors, and all packets are preceded by an 8-byte header, which is also encoded. We recall that the transmission format is designed to be easily supportable by diverse acoustic modem implementations. To this end, transmissions are performed in the 9-14 kHz band. The very low bit rate of the resulting system, 160 bps including PHY-level coding, restricts the effective transmit rate to at most 80 information bits per second. Hence, protocols must be very effective at employing such a scarce resource, and should impose little overhead, as a general rule. This explains why we chose to compare a protocol with no overhead (ALOHA) with another bearing light overhead (T-Lohi) and a third imposing greater overhead (DACAP). For each protocol, we considered both an ACK and a no ACK version: in the latter case, we set the maximum number of retransmissions of any packet to 5.

We arrange nodes in a rectangular grid topology that covers an area of 5 km × 2 km. We deploy either 4 or 10 nodes, by dividing the area in 4 (respectively, 10) rectangles and placing a node at the centre of each rectangle. These nodes must communicate to a sink placed at the centre of the network area. We reproduce two different traffic scenarios: in the first one, nodes periodically report environmental data. The corresponding traffic is generated according to a Poisson process of rate λ packets per second per node. In the second scenario, nodes must detect moving objects that traverse the network area. To emulate the corresponding bursty traffic pattern, we assume that an object crosses the network and triggers packet generation events at the rate of 1 packet every 10 seconds whenever it comes within the detection range of a node. For simplicity, we assumed that the detection range of a node is 1.5 km, and consider only a 10-node topology, which adequately covers the area.

We have implemented the protocols described in Sec. 2 using ns2 [7] and the ns2-MIRACLE extensions [8]. In order to reproduce acoustic propagation, we have fixed a location for our experiments at 49.25°N 10.125°E, close to the Pianosa island, off the north-eastern coast of Italy. Bathymetry data have been taken from the General Bathymetric Chart of the Oceans [9], a public database offering 30-arcsecond spaced samples; bottom sediment parameters are taken from the National Geophysical Data Center's Deck41 database [10]. Sound speed profiles are computed as averages of the profiles measured during the GLINT'08 sea trials [11] or from the World Ocean Database [12] if the GLINT data set did not cover the wanted location and month of the year. The Bellhop ray tracer [13] is finally used to simulate signal propagation among all nodes, including the sink.

3.2. Results for periodic traffic

We start our performance evaluation by considering throughput, defined as the fraction of the offered traffic per node (indicated in the abscissa) that the protocol can manage. In

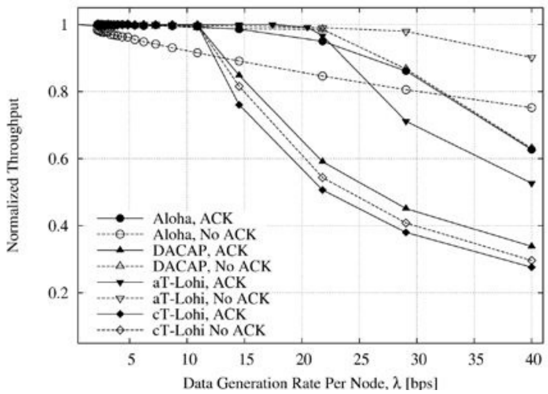

Fig. 3: Throughput against λ, 4 nodes.

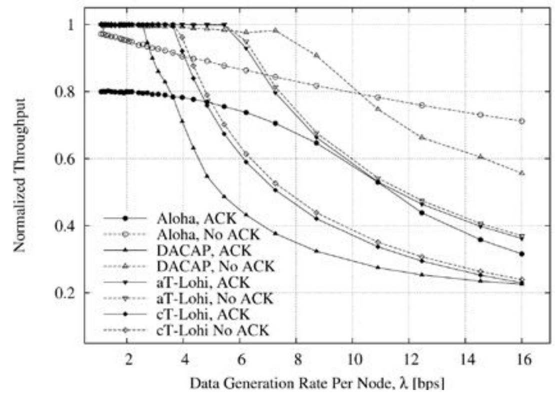

Fig. 4: Throughput against λ, 10 nodes.

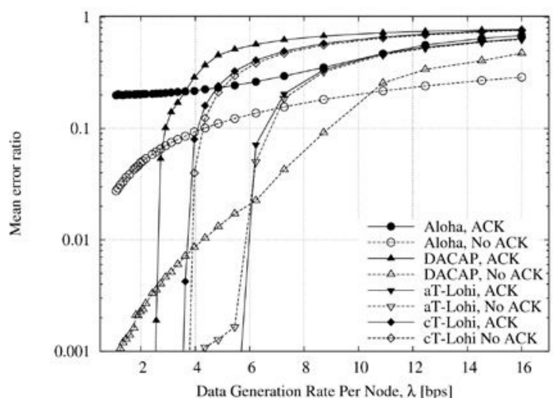

Fig. 5: Error rate against λ.

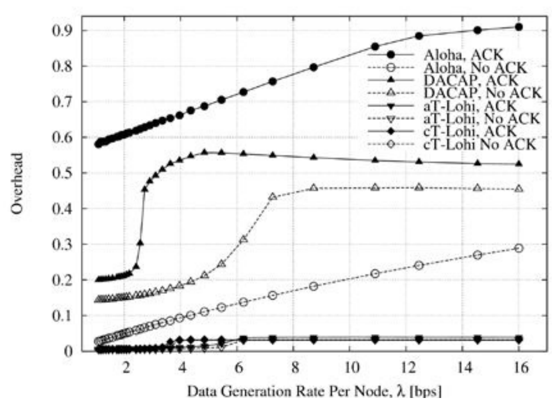

Fig. 6: Overhead against λ.

particular, Figs. 3 and 4 show throughput for a network with 4 nodes and 10 nodes, respectively. With 4 nodes, the overall traffic load is very light, therefore light handshakes (such as in aT-Lohi) or no handshakes (ALOHA) are preferred with respect to more complicated handshakes (DACAP) and cT-Lohi, which forces the nodes to stay silent for longer times before transmitting. From Fig. 3 we also infer that, in general, using ACKs tends to decrease the throughput, due to both the longer handshake duration and the probability that an ACK message collides with data packets. We observe similar trends in Fig. 4, where the presence of 10 nodes yields greater traffic. In this case, handshake-based protocols are put under heavier stress: at low to medium traffic aT-Lohi (both with and without ACK) and DACAP (no ACK) offer good throughput, but are eventually outperformed by ALOHA (no ACK), which is much lighter for greater λ.

The drawback of this version of ALOHA can be seen in Fig. 5, which depicts the fraction of erroneous transmissions over all transmissions. This figure shows that ALOHA bears a high error rate (roughly 1/3), yet lower than the failure rate of coordinated protocols, which undergo a greater probability that signalling transmissions interfere with data transmissions. On the other hand, the benefit of coordination among nodes is observed in particular at low rate, where the error probability of ALOHA is outperformed by all versions of T-Lohi and DACAP. To conclude this first part of our comparison, we show in Fig. 6 the protocol overhead, defined as the ratio of non-data bits sent over all transmitted bits (purged of convolutional coding overhead). Hence, this figure accounts for both signalling overhead and erroneous packets (which have to be retransmitted). We see that the lowest overhead is achieved by the no ACK versions of cT-Lohi and aT-Lohi, whereas ALOHA with no ACK has slightly greater overhead due to the larger number of collisions between data packets that take place as traffic increases. We recall in fact that collisions are the only source of overhead for no ACK ALOHA, whereas for other protocols the number of signalling messages increases, and consequently collisions of data and signalling packets increase as well. This also shows that DACAP has a worse overhead performance, due to its longer handshakes.

3.3. Results for event-driven traffic

In this scenario, we recall that an object moves through the network and is detected by the nodes. Unlike in the previous evaluation, here we focus on the arrival time of the first packet triggered by the event detection (a measure of the readiness of a protocol) and on the arrival time of the last packet (a measure of the ability of a protocol to handle bursty traffic). Again we consider ACK and no ACK versions of all protocols, and assume that ACK versions are fully reliable (infinite number of retransmissions). For the no ACK versions, we also measure the packet error rate. We report in Table 1 the packet arrival times and the error rate for ALOHA, DACAP, aT-Lohi and cT-Lohi. Unlike in the previous set of results, here we observe that ALOHA with ACK yields the lowest arrival times of all fully reliable protocol versions. This is due to the bursty traffic pattern, which causes packet generation events to eventually take place away of the first nodes that sense the moving object. A reduction of local interference follows, so that the FH-BFSK modulation format and the convolutional code together can sufficiently protect transmissions from bit errors. In addition to the simplicity of the ALOHA scheme, this also explains why ALOHA performs better.

A second observation is in order here: while ALOHA without ACK bears an unacceptable error rate, the no ACK versions of the other handshake-based protocols trade off error rate for the overall duration of transmissions. In particular, cT-Lohi yields almost zero errors even without ACK, but takes 183 s to complete transmissions. aT-Lohi's errors are on the order of 0.3%, but the protocol takes 116 s to complete, which is very close to no ACK ALOHA. Finally, DACAP yields a 3.7% error rate, but completes in a shorter time than fully reliable ALOHA (98 s against 105 s). Therefore, if such error rates can be withstood by the application running on top of the protocols, it might be worthy to consider non-reliable handshake-based schemes instead of fully reliable ALOHA. Furthermore, we recall that, unlike DACAP and ALOHA, aT-Lohi and cT-Lohi keep nodes in a low-power

Protocol	First packet arrival (s)	Last packet arrival (s)	Error rate
ALOHA (ACK)	1.15	105	---
DACAP (ACK)	1.92	178	---
aT-Lohi (ACK)	1.43	127	---
cT-Lohi (ACK)	1.72	202	---
ALOHA (no ACK)	1.15	29.6	0.43
DACAP (no ACK)	1.80	98.0	3.7e-2
aT-Lohi (no ACK)	1.43	116	1.9e-3
cT-Lohi (no ACK)	1.72	183	3.4e-4

Table 1: Packet arrival times and error rate for the event-driven traffic scenario.

state while not transmitting: this offers the opportunity to save energy, which must also be accounted for when choosing the network protocol to be employed in the network.

4. CONCLUSIONS

In this paper, we have presented a comparison among three different protocols that are suitable for use in underwater networks employing a low rate FH-BFSK physical layer with convolutional coding. Each protocol bears a different level of coordination and a correspondingly different handshake complexity: ALOHA (no coordination), T-Lohi (light coordination) and DACAP (strong coordination). For each protocol, we have tested both a reliable and a non-reliable version, under both periodic and event-driven traffic. Our study highlights that while the aggressive version of T-Lohi yields the best performance under periodic traffic, there is no clear winner in the event-driven traffic case, as the best protocol choice actually depends on the capability of the application to withstand a certain fraction of packet errors.

5. ACKNOWLEDGEMENTS

This work has been partially supported by the NATO Undersea Research Centre, La Spezia, Italy, under contract no. 40480700 (ref. NURC-010-08).

REFERENCES

[1] **A. Syed**, **W. Ye**, **J. Heidemann**, Comparison and evaluation of the T-Lohi MAC for Underwater Acoustic Networks, *IEEE J. Sel. Areas. Commun.*, 26 (9), pp. 1731-1743, 2008.

[2] **B. Peleato**, **M. Stojanovic**, Distance Aware Collision Avoidance Protocol for Ad Hoc Underwater Acoustic Sensor Networks, *IEEE Comm. Letters.*, 11 (12), pp. 1025-1027, 2007.

[3] **M. Zorzi** *et al.*, Energy-Efficient Routing Schemes for Underwater Acoustic Networks, *IEEE J. Sel. Areas. Commun.*, 26 (9), pp. 1754-1766, 2008.

[4] **L. Badia** *et al.*, An Optimization Framework for Joint Sensor Deployment, Link Scheduling and Routing in Underwater Sensor Networks, In *Proc. of ACM WUWNet*, pp. 56-63, 2006.

[5] **L.G. Roberts**, ALOHA packet system with and without slots and capture, *ACM SigComm Comp. Comm. Rev.*, 5 (2), pp. 28-42, 1975.

[6] The Network Simulator, ns2. [Online] Available: http://www.isi.edu/nsnam/ns/.

[7] **N. Baldo** *et al.*, NS2-MIRACLE: a Modular Framework for Multi-Technology and Cross-Layer Support in Network Simulator 2, In *Proc. of ACM NSTools*, 2007.

[8] The General Bathymetric Chart of the Oceans. [Online] Available: www.gebco.net.

[9] National Geophysical Data Center, Seafloor Surficial Sediment Descriptions. [Online] Available: http://www.ngdc.noaa.gov/mgg/geology/deck41.html.

[10] **H. Schmidt** *et al.*, GOATS 2008: Autonomous, Adaptive Multistatic Acoustic Sensing, *Project Report*, 2008. [Online] Available: www.onr.navy.mil/sci_tech/32/reports/docs/08/oaschmi3.pdf.

[11] World Ocean Atlas. [Online] Available: www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html.

[12] **M. Porter** *et al.*, BELLHOP code. [Online] Available: http://oalib.hlsresearch.com/Rays/index.html.