A Performance Comparison of MAC Protocols for Underwater Networks using a Realistic Channel Simulator

Federico Guerra, Paolo Casari, Michele Zorzi
Department of Information Engineering, University of Padova
Via G. Gradenigo, 6/B, 35131 Padova, Italy
{fguerra,casarip,zorzi}@dei.unipd.it

***Abstract*—Large-sized underwater testbed deployments still pose a number of challenges, first of all the cost of the nodes and of the deploying ships. Therefore, simulation becomes an important tool for the assessment of network performance, and a precious aid to protocol, topology, and deployment design.**

However, simulations are significant only if the reproduction of acoustic propagation is accurate. To this end, we have joined two well known tools for simulation, namely Bellhop (for acoustic propagation modeling through ray tracing) and ns2-MIRACLE (an event-based network simulator). These tools, together, provide a flexible and customizable environment, fostering more realistic reproduction of propagation, PHY-level behavior, as well as the detailed specification of medium access control, routing, and higher-level protocols.

In this paper, we describe our tool, and give an example of its employment in the comparison of three MAC protocols for underwater networks over different kinds of physical layers. The protocols have been specifically chosen to shed some light on the relationship between the complexity of a protocol and the amount of coordination it enables among nodes: in other words, our results show when it is better to rely on plain random access, and when on some form of handshaking (despite the usually greater complexity of handshake-based protocols).

I. Introduction

In a couple of decades, underwater networks of autonomous fixed and mobile nodes will make up a very important engineering tool: the increasing feasibility of underwater networking, proven by the considerable improvements on PHY-level technologies and communications techniques, is paving the way for the use of autonomous underwater devices (either mobile or fixed) in a number of applications. Primitive examples of underwater networks already exist for water column monitoring: it is in fact a quite common practice to deploy one or more moorings, specifically designed to accommodate sensors of different types and functions. While such moorings usually feature a radio buoy for transmitting sensed data ashore, some examples are available [1], [2] where data are transmitted over an acoustic link, using commercially available transmission devices such as the WHOI MicroModem [3] or the Teledyne-Benthos modem [4]. Underwater networks differ from terrestrial radio networks mainly in terms of the channel where acoustic waves propagate. Water imposes a lower propagation speed to acoustic waves; furthermore, the propagation paths and patterns are tightly coupled to the physical characteristics of water. Temperature, salinity and pressure (i.e., depth) change the refraction index of water (thus the local speed of sound) and bend acoustic waves depending on the local sound speed along the propagation path. The resulting channel exhibits a plethora of diverse behaviors, which depend on the depth of water, sea bottom sediments, bathymetric profile, latitude, time of the year (which influences temperature, thus propagation), position of the transmitter and receiver with respect to the surface and seabed. As a consequence, specific channel characteristics such as multipath propagation may change very significantly depending on all factors cited above. Also, noise sources such as turbulence, wind, ships, animals and weather conditions add up to pure thermal noise in the receiver circuitry to form a more complex, environment-dependent, non-white noise process which affects communications significantly, and must be properly taken into account.

In such a complex environment, setting up communications may prove difficult, starting from the simplest link budget operation. In fact, there is no model for underwater propagation that is widely agreed upon, and only a few steps have been recently taken to characterize the underwater acoustic communication channel (e.g., see [5]–[7]). In substitution of these models, synthetic equations exist that describe the attenuation incurred by a signal in a given frequency band as it propagates over a path of known length [8]; these equations are, however, but mildly representative of underwater propagation effects, and serve only as a coarse approximation.

In this context, at-sea experimentation of communication schemes is of paramount importance to ensure that their predicted performance can be actually achieved, and that those devices predicted to perform well in a given environment keep similar performance, or at least continue to work correctly, when the environment changes. A significant drawback of such experiments, however, is their very high cost, due to underwater devices (modems, transducers, hydrophones, waterproofing equipment, and so forth) as well as ship and manpower required to actually deploy the nodes. When it comes to implementing a whole network, the size of the deployment should be large enough to make the experiment grow beyond a simple proof-of-concept, therefore its costs correspondingly increase, explaining why such deployments as SeaWeb [9] or the testbed employed during the GLINT 2008 experiments [10] are almost unique. This is the main reason why simulation represents a very important tool for underwater networking: it can help pick the best approach out of a number of candidates and let only that approach make it to the more costly effort of real implementation.

So far, no specific simulation tool has been made available to re-create the multi-fold complexity of underwater acoustic

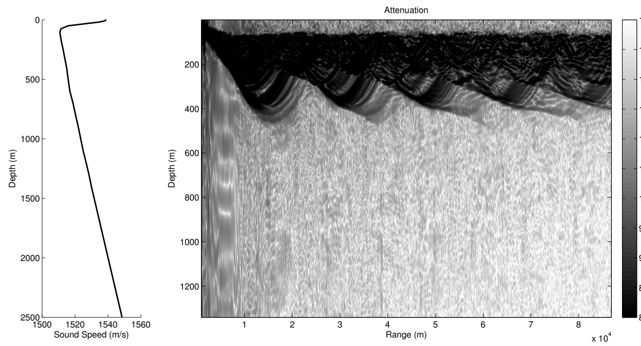

Figure 1. Attenuation incurred by acoustic waves transmitted in August from off the shore of the Italian region of Calabria, 39.5°N, 17°E. A darker shade of grey represents a stronger signal.

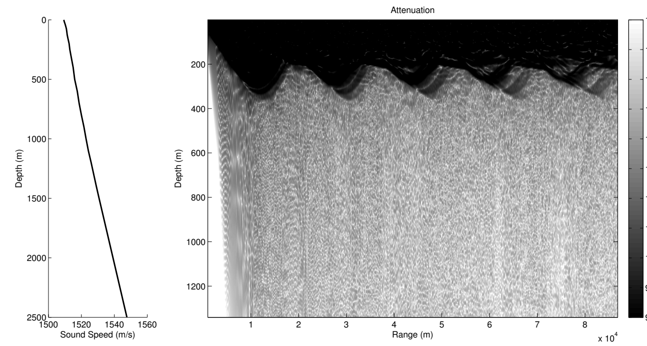

Figure 2. Attenuation incurred by acoustic waves transmitted in August from off the shore of the Italian region of Calabria, 39.5°N, 17°E. A darker shade of grey represents a stronger signal.

networks; conversely, a standard approach has been to inherit communication schemes and protocols from terrestrial radio, whose code is run on top of a modified but simplified physical layer. The main modifications include the approximation of the sound speed as constant ($1500\,\mathrm{m/s}$), and the use of empirical formulas (e.g., see [8], [11]) to approximate attenuation and noise. Unfortunately, this approach is useful only to get a coarse idea of underwater networking performance, and does not capture the more complex effects observed in salted waters. For example, consider the north-eastern coast of the Italian region of Calabria (at around 39.5°N, 17°E) in August (Figure 1) as opposed to January (Figure 2). In August, the temperature of the upper water layer is higher, and therefore the local sound speed is initially larger, and decreasing up to roughly $100\,\mathrm{m}$. From there, it starts increasing again, mainly due to pressure. This bends the signal downwards and tends to confine it at below $150\,\mathrm{m}$ so that little power actually reaches the surface, as shown by the light grey zone close to the surface. Also, note that the sound is confined around the zone of minimum sound speed. Conversely, in winter (Figure 2) the temperature of the upper layers is lower, and so is the sound speed. This makes sound speed constantly increase with depth, and bends sound upwards, improving the insonification of the upper layers, but making very little power reach the lower depths of the site. Such effects would not be correctly captured by approximate propagation models, and yet they represent a key factor in the design of a bottom mounted sensor network (a fairly common approach): by predicting a larger attenuation, one could decide whether or not to change the orientation of the source, or to deploy additional equipment (e.g., a surface buoy with an acoustic and a radio interface, to gather acoustic signals and bridge the bottom-mounted sensors to shore through a radio link).

In this paper, we introduce a simulator for underwater networks which blends a more realistic physical layer reproduction and a flexible tool for network protocol modeling. The equations of underwater propagation for acoustic waves are solved by means of ray tracing using Bellhop [12], while the ns2-MIRACLE framework [13]–[15] provides the necessary harness to set up a networking environment.

After that, we use the simulator for comparing three different protocols, namely ALOHA [16], DACAP [17] and Tone-Lohi [18], under different traffic generation models and network sizes. The protocols have been purposely chosen to bear a different amount of signaling (thus coordination as well as overhead) among nodes. While the reader is referred to Section III for a more detailed description, here suffice it to say that ALOHA represents the class of completely random access protocols, Tone-Lohi the class of light contention-based protocols (where the contention for channel access is driven by transmitters on demand) and DACAP the class of collision avoidance protocols (where handshaking tries to ensure that the communication is not affected by excessive interference).

With respect to the existing literature on the performance of MAC protocols for underwater networks, our study specifically addresses the relationship between the level of node coordination set up by the protocols and the network performance; furthermore, we carry out our evaluation using a realistic physical layer model, that reproduces acoustic propagation significantly better than empirical formulas and models the behavior of modulation and coding schemes in detail. These aspects make our work different from recent performance evaluations of MAC protocols for underwater networks, such as [19] or the comparison in [17]. It should be noted that while in [20] simulations are carried out using a more detailed model of acoustic propagation, the authors considered only the ALOHA protocol, whereas here we extend our evaluation to different protocol paradigms and provide a direct comparison among them. Some preliminary results on this comparison were also presented in [21].

II. Brief description of the simulator

As introduced above, our simulator (named WOSS, for World Ocean Simulation System [22]) exploits the full capabilities and potential of the ns2–MIRACLE framework [13]–[15], including packet-based communications, medium access control (MAC), routing and upper-layer protocols, cross-layer protocol parameter exchange, mobility, traffic models, performance metrics, and so forth. These capabilities are integrated with a more realistic reproduction of acoustic propagation yielded by the Bellhop tool [12]. In order to solve ray equations, however, Bellhop requires contour information involving the sound speed profile (SSP) (i.e., the variation of sound speed with depth, which impacts the amount of bending incurred by waves), the local bathymetry and the type of bottom sediments (both required to characterize bottom reflections). For this reason, the simulator has been interfaced with public databases such as the World Ocean Database [23] (for SSPs), the General Bathymetric Chart of the Oceans [24] and the National Geophysical Data Center's Deck41 data-base [25] (for bottom sediments).

Thanks to our interfaces, the user only has to specify at which geographical coordinates the simulation experiment should take place, and leave it to the simulator to run Bellhop and estimate propagation effects for each pair of nodes in the network (calculations are repeated automatically if nodes move). The whole process is made transparent to the user, so that it is possible to concentrate on the design of the network deployment and protocols, rather than on propagation issues. More details on the simulator are given in [22].

III. Simulated protocols

Before discussing in Section IV-B the performance of the protocols involved in our comparison, we summarize in the following subsections the rules of each protocol.

A. ALOHA

ALOHA [16] is the most basic random access protocol, its only rules being that any node with data available can immediately send it. A slightly more refined version (that we implemented) requires the node to perform instantaneous channel sensing before transmitting: this makes it possible to detect ongoing channel activity and avoid collisions if the receiver of an ongoing exchange is located in the proximity of the node which is about to transmit. It could be argued that, under high traffic generation rates, this policy tends to synchronize nodes and favor collisions; actually, the propagation delay among different pairs of nodes tends to be different enough so as to decouple packet receptions. Nevertheless, in multiuser networks, collisions may still take place if two users attempt to access the channel at the same time. In this case, standard contention resolution techniques can be applied (e.g., random backoff before rescheduling a transmission attempt following a collision).

The performance of ALOHA is expected to be poor in general. Its throughput is usually very low, as repeated collisions may cause multiple retransmissions and ultimately lead to even more collisions, especially at high traffic. Nevertheless, ALOHA is a feasible option in many underwater networks, if the packet generation rate is sufficiently small and the long propagation delays significantly alter the time of arrival of transmissions between different pairs of nodes (unlike in radio networks, where two simultaneous transmissions in the same area always collide). For example, in [26] the authors show that ALOHA is a feasible solution for MAC in an underwater multihop network, as it offers greater throughput and lower latency, especially compared to more complex 4-way handshaking-based schemes.

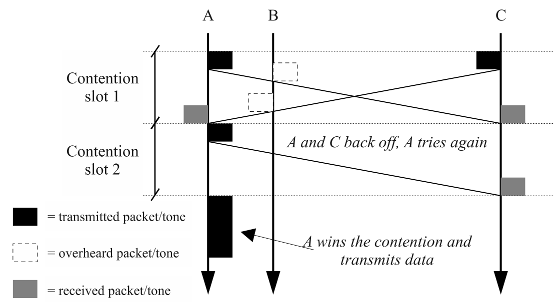

Figure 3. Scheme of transmissions in Tone-Lohi (adapted from [18]).

B. Tone-LOHI

Tone Lohi (T-Lohi) [18] is a reservation-based MAC protocol. In T-Lohi, nodes contend for the channel by sending a wakeup tone during a preliminary reservation phase which precedes data transmission. This phase allows not only to elect a winner, but also to count the number of contenders (by overhearing other tones over a certain time) and consequently the behavior during the contention. Namely, after having detected the presence of other transmitters, a node backs off for a random amount of time before sending another tone: this amount of time depends on the detected number of contenders. Sound propagation in water is slow enough to let nodes count contenders fairly accurately, so long as contention packets occupy the channel for much less than the propagation delay. Other protocol phases (such as reception) are driven by wakeup tones, so that the nodes can mostly stay asleep, and get activated by incoming transmissions. On the other hand, the use of tones requires a specific wake-up tone detector, which permanently operates in a low power listening mode.

The reservation procedure is better described by referring to the synchronized version of T-Lohi, namely ST-Lohi [18]. With reference to Figure 3 assume that all nodes share the same time synchronization (this assumption will be removed later). Time is divided into contention slots of length equal to the maximum propagation delay plus the tone length. All nodes contending for the channel must send their reservation tones at the beginning of a slot. After that, each node waits for the arrival of other tones during the rest of the contention slot (the node can also go to sleep, as the wakeup tones would turn it on upon reception). If the node does not hear any other tone, it assumes it has won the contention and transmits its packet. Otherwise, if other tones are heard, the node starts the contention with the tone senders by picking a random amount of time (in multiples of a slot length) and sending another tone after this backoff period. The objective of this process is to improve the probability that eventually a node is the only one to send a tone in a certain slot, thus being elected the winner of the contention.

To remove the synchronization assumption toward a more realistic implementation of the T-Lohi protocol, assume that the nodes operate in an asynchronous manner, send tones when they have a packet to transmit, and then wait for a prescribed amount of time in order to detect possible contenders. The waiting time can be tuned in order to obtain a conservative

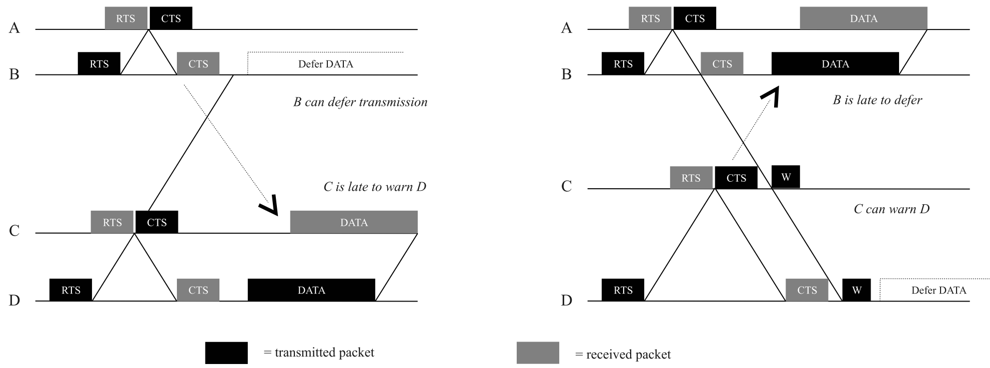

Figure 4. Scheme of transmissions in DACAP (adapted from [27]).

and an aggressive protocol version: namely if the length of the waiting time is equal to the tone duration plus the maximum propagation delay (defined, e.g., as the delay to the distance where a transmitted signal can still be heard) the protocol is more aggressive (and is named aT-Lohi); otherwise if the wait time is the tone duration plus twice the maximum propagation delay, the protocol is more conservative (and is named cT-Lohi) [18].

C. DACAP

The Distance-Aware Collision Avoidance Protocol (DACAP) [17] is designed to exploit the knowledge of the distance among nearby nodes (which can be inferred from the duration of a handshake taking place before transmission), in order to improve the efficiency of the handshake itself. The access scheme is non-synchronized, as ALOHA and T-Lohi.

The access scheme follows the well known 4-way collision avoidance paradigm, whereby the sender transmits a short Request-To-Send (RTS) packet to communicate its request to access the channel; the receiver issues a Clear-To-Send (CTS) message back to the sender if it is ready to receive the data packet. Depending on the presence of simultaneous handshakes nearby and on the relative distances of the nodes, two relevant scenarios may arise where potential collisions can be detected and avoided in time: i) the receiver overhears an RTS after sending the CTS, indicating likely interference coming from a data packet being transmitted shortly thereafter by a nearby node; in this case the receiver of the CTS warns it own sender (provided that there is sufficient time to do so), so that it defers data transmission; ii) if any node overhears a packet meant for another neighbor, or receives a warning from a receiver, it defers the data transmission [17]. These scenarios are exemplified in Figure 4. The proper tuning of the idle period length allows to avoid collisions: the objective is to let nodes be aware of any neighbors threatening data transmissions and react in time. The timing is also designed so that any signals coming from far nodes are neglected, as they are foreseen to yield little impact on the signal-to-interference-and-noise ratio at the wanted receiver. This effectively shortens waiting times and improves the efficiency of the protocol, provided that the neglected signal are actually harmless.

DACAP's handshaking pattern (which is indeed heavier than the method employed in T-Lohi) may of course be impaired if warning packets arrive too late, or the nodes become aware of potential collisions too late to defer data transmissions (see also Figure 4); however, the degree of protection is still fairly high. In addition, the user could achieve further immunity to interference by adjusting protocol timings (thus trading off speed for interference protection). In order to achieve a tradeoff that maximizes throughput of a network, a minimum handshake length is set for all the nodes. In a network where most links are as long as the transmission range, this minimum length must be as high as twice the maximum propagation delay; in a deployment such as ours, where all links are shorter, it can be reduced. We consider two versions of the protocol, namely with 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. The DACAP protocol (with or without ACK) has been designed to control collisions and thus allow the coexistence of multiple data communications in the network. Results show [17] that DACAP represents an intermediate solution with respect to ALOHA with channel sense (high throughput for low offered traffic, but very prone to congestion) and more advanced slotted access protocols such as Slotted-FAMA [28].

IV. Network Simulation using WOSS

A. Description of the scenario

To demonstrate the results obtainable using our simulator, we present a comparison of three medium access control (MAC) protocols, ALOHA [16], Tone Lohi (T-Lohi) [18] (both its aggressive and its conservative version) and DACAP [17]. We recall that the choice of these protocols has been driven by their different channel access concept. Specifically, ALOHA requires no coordination, T-Lohi introduces a form of contention among nearby nodes that are simultaneously accessing the channel, whereas DACAP relies on a 4-way signaling scheme (RTS/CTS/DATA/ACK + an additional Warning message) between the transmitter and the receiver, achieving better coordination among nodes at the price of extra overhead and increased interference. More specifically, the presence of these signaling packets has two negative effects: i) signaling

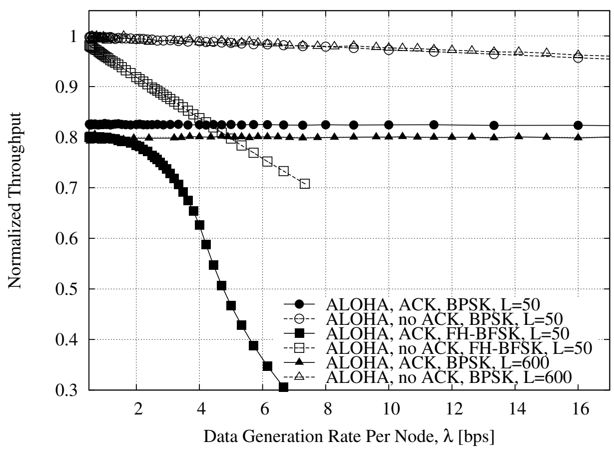

Figure 5. Throughput as a function of traffic for ALOHA in August, using FH-BFSK and BPSK, $L = 50, 600\,\mathrm{Bytes}$, 10 nodes.

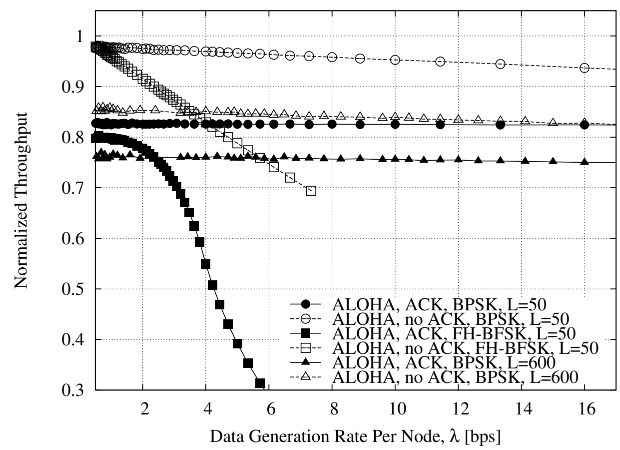

Figure 6. Throughput as a function of traffic for ALOHA in January, using FH-BFSK and BPSK. $L = 50, 600\,\mathrm{Bytes}$, 10 nodes.

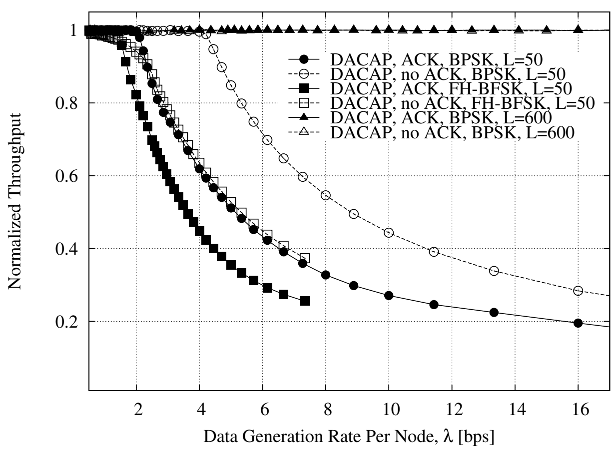

Figure 7. Throughput as a function of traffic for DACAP in August, using FH-BFSK and BPSK. $L = 50, 600\,\mathrm{Bytes}$, 10 nodes.

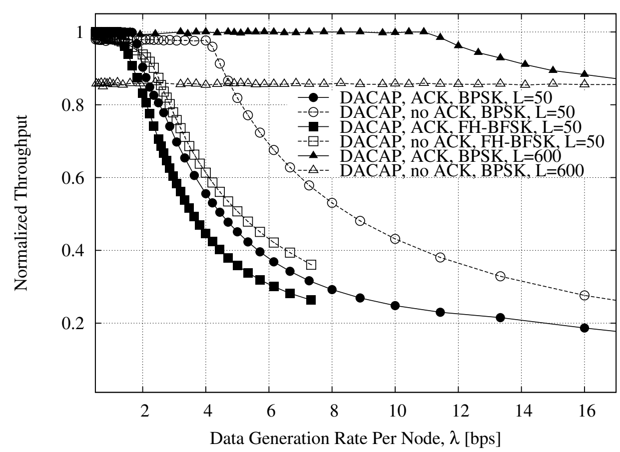

Figure 8. Throughput as a function of traffic for DACAP in January, using FH-BFSK and BPSK. $L = 50, 600\,\mathrm{Bytes}$, 10 nodes.

packets consume bandwidth, and waiting for a reply to one such packet imposes silent times which cannot be used for transmission; ii) due to the infeasibility of synchronizing all nodes, the possibly different propagation times may cause further collisions, both among signaling packets and between signaling and data packets. The comparison between such different schemes allows a direct assessment of an interesting relationship, namely, between the amount of coordination enforced by the protocol (also in terms of the corresponding overhead) and the network performance. Understanding this relationship gives clues about how much it pays off to invest resources (signaling overhead, time, etc.) and on how much this translates into actual performance gains.

In our comparison, we assume that 10 (20) nodes are arranged in a 5×2 (5×4) grid, with nearest neighbors $1\,\mathrm{km}$ apart, close to the Pianosa island, a protected site off the north-western coast of Italy. That corresponds to geographic coordinates 49.25°N 10.125°E, which is also the location we set in the simulator to fix environmental parameters. We also assume these nodes are equipped with standard hardware, supporting a low-rate ($160\,\mathrm{bps}$ including $1/2$-rate convolutional encoding), collision-resistant Frequency Hopping Binary Frequency Shift Keying (FH-BFSK) modulation, reflecting, e.g., the choice of parameters for robust unsolicited communications and beaconing; such choices are at the base of the JANUS protocol described in [29], [30]; we also consider a higher rate ($4800\,\mathrm{bps}$) Binary Phase Shift Keying (BPSK) modulation. We simulate the MAC protocols on top of both modulation schemes, and consider both a version with and a version without ACK messages to confirm correct reception. Packets are generated according to a Poisson process of rate λ packets per second per node. The length of each data packet is fixed to $L = 50\,\mathrm{Bytes}$ in all experiments, with the exception of some results obtained using $L = 600$. ACKs and DACAP's signaling packets are $4\,\mathrm{Bytes}$ long. All data gathered by the nodes is to be reported to a common sink, which is assumed to be located on the surface, at the center of the network deployment: in other words, all communications are from the sea bottom to the surface. In order to stress the protocols under comparison, all nodes are located within the coverage range of the sink, and therefore their communications are very likely to collide.

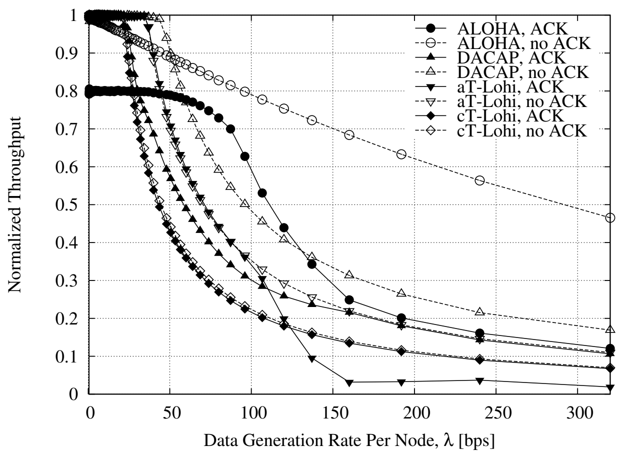

Figure 9. Throughput as a function of traffic for all protocols in August, using BPSK. $L = 600\,\mathrm{Bytes}$, 10 nodes.

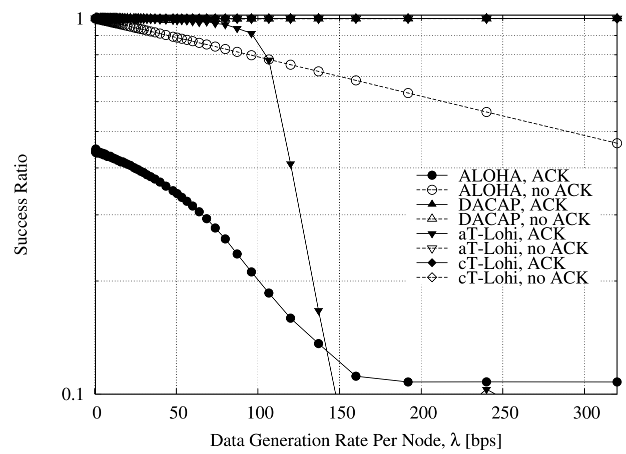

Figure 10. Success ratio as a function of traffic for all protocols in August, using BPSK. $L = 600\,\mathrm{Bytes}$, 10 nodes.

B. Numerical Results

The first set of results aims at demonstrating how environmental factors affect network performance. In this case, the network runs differently in August than it does in January. The main reason is the change in sound propagation due to the different temperature, as was observed in Figures 1 and 2: August's warmer waters tend to make received signal power locally stronger on average. Figures 5 to 8 show the performance variation (in terms of throughput) for ALOHA and DACAP. Both FH-BFSK and BPSK are considered; furthermore, for each protocol, we report the results of both its ACK and no ACK versions. The (normalized) throughput is defined here as the number of data bits per second that successfully reach their intended destination, divided by the number of generated data bits. For example, ALOHA generally incurs better performance in August (Figure 5) than in January (Figure 6), especially for longer packet sizes. A larger packet is in fact more prone to errors generated by a low SNR at the receiver: in January the lower signal level makes the reception of longer packets less likely to be correct. Consider, for instance, the ALOHA versions for $L = 600$ (black and white triangles for ACK and no ACK respectively): throughput decreases from roughly 1 in August to around 0.8 in January for the no ACK version, and from 0.8 to 0.75 for the ACK version. Similar effects can be inferred for the shorter $50\,\mathrm{bytes}$ packets, even though this size makes the transmission more resilient to noise.

Other interesting remarks can be made by looking at DACAP's performance in August (Figure 7) and January (Figure 8). While the general effect of the environment on transmissions is the same as before, we recall that DACAP features a 4-way handshaking scheme, which makes it very different from the completely random access performed by ALOHA. With DACAP, the communication setup phase (RTS/CTS/Warning exchange) allows for a low interference area around the transmitter and receiver, which improves the chances of success. This reflects on throughput, which is actually greater than ALOHA's in both August and January for most values of the traffic rate λ. In DACAP, however, the handshaking overhead introduces inefficiencies if packets are too short, as round trip times tend to be longer than transmission times. Therefore, the full capabilities of DACAP are exploited with long packets (again, black and white triangles for ACK and no ACK respectively). For example, DACAP outperforms ALOHA both in August and in January for $L = 600$, whereas it is outperformed by ALOHA (at medium to high traffic) for $L = 50$.

To clarify this point, let us consider the next set of results comparing all protocols (ALOHA, DACAP, and both the conservative and the aggressive flavors of T-Lohi). Figure 9 shows throughput; Figure 10 reports packet success ratio, defined as the fraction of all transmitted packets that successfully reach their destination; Figure 11 shows overhead, defined as the average fraction of transmitted bits dedicated to signaling packets and tones; finally Figure 12 depicts application-level success ratio, defined as the ratio of the correctly received packets over all *generated* packets. The rationale behind this last metric is that while MAC-level success ratio may be sufficiently high, MAC-level mechanisms (handshakes, backoffs, etc.) may prolong the time required to actually deliver a packet over the acoustic channel. In this case, the backlog of the nodes tends to rise, and newly generated packets may get discarded due to full transmit queues. This cannot be inferred by simply looking at throughput, success ratio or overhead, as these metrics are calculated over transmitted packets only. Because any application running on the nodes (e.g., environmental monitoring and data gathering) usually assumes that a certain minimum performance level is provided, it is important to check, e.g., if generated packets get dropped. By virtue of the previous discussion on the efficiency of handshake-based schemes, all results from this point on are obtained in August, using BPSK and a packet length of $L = 600\,\mathrm{Bytes}$.

Figure 9 opens the discussion by summarizing the previous results on throughput and adding a further comparison with cT-Lohi and aT-Lohi. This figure shows that the two T-Lohi flavors and DACAP achieve the maximum throughput, and moreover they are capable of maintaining such level up to a certain generated traffic rate, after which performance starts to drop. This behavior is explained by the higher level of

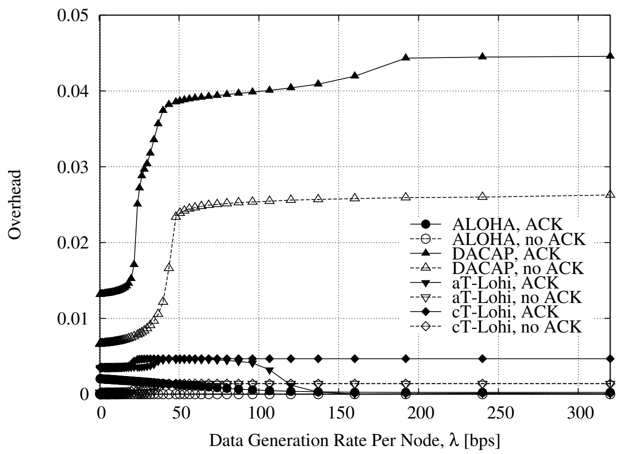

Figure 11. Overhead as a function of traffic for all protocols using in August, BPSK. $L = 600\,\mathrm{Bytes}$, 10 nodes.

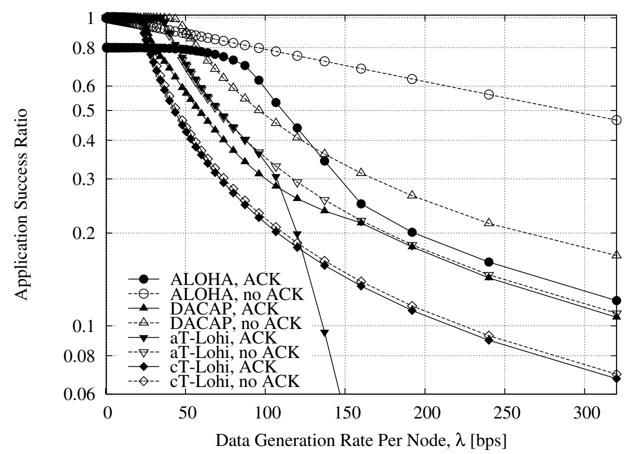

Figure 12. Application-level success ratio as a function of traffic for all protocols in August, using BPSK. $L = 600\,\mathrm{Bytes}$, 10 nodes.

coordination enforced among nodes. Conversely, ALOHA's throughput starts to decrease even at low traffic, due to the increasingly frequent collisions that random access protocols inherently incur. However, it is worth noting that the throughput of ALOHA (both with and without ACK) decreases more mildly than for other protocols, thanks to the absence of overhead messages and waiting times before data transmission. Also note that T-Lohi achieves worse performance with respect to DACAP. The reason is two-fold: on one hand its one-way contention mechanism makes it possible that two nodes, hidden to each other, transmit a tone, sense a free channel and simultaneously transmit data, thus colliding: this hidden terminal problem is inherently absent in DACAP thanks to the RTS/CTS handshaking; on the other hand, especially at high traffic, prospective transmitters tend to prolong backoff time in order to resolve contentions. Thus, the transmission latency becomes larger, and correspondingly decreases throughput.

This last observation is confirmed by Figure 10, showing packet success ratio: as expected, ALOHA's success ratio consistently decreases with offered traffic. Conversely, DACAP's and T-Lohi's success ratio is always close to 1, meaning that throughput decreases mostly because of longer waiting times, rather than due to collisions. An exception to this behavior is aT-Lohi with ACKs, where the aggressiveness of the protocol leads to a higher chance of collisions among ACKs and between data packets and ACKs, thus decreasing the success ratio.

Overhead, depicted in Figure 11, gives a different point of view by measuring the further effort required by the protocols to coordinate transmissions. As expected, DACAP yields the greatest overhead, as its RTS/CTS handshake proves more resource-consuming. The overhead of the other protocols is 2 to 4 times lower, depending on the presence or absence of ACKs. It is also expected that for ALOHA the overhead is zero (for the no ACK version) or close to zero (for the ACK version). Figure 11 suggests that DACAP's larger overhead pays off in terms of general performance, as its throughput is higher (compared to T-Lohi and also to ALOHA up to a certain traffic rate), for comparable or better success ratio.

However, we recall that overhead, as well as throughput and

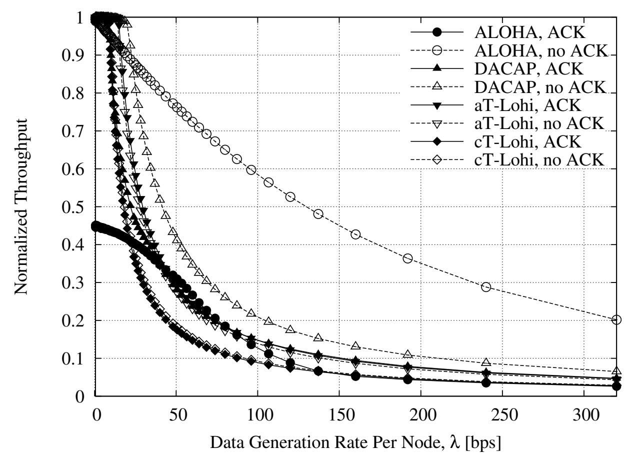

Figure 13. Throughput as a function of traffic for all protocols in August, using BPSK. $L = 600\,\mathrm{Bytes}$, 20 nodes.

success ratio, consider only the transmitted packets, neglecting the generated packets which get discarded due to excessive backlog at the nodes. An indication of this specific effect can be inferred from Figure 12, depicting the application-level success ratio. We observe that all coordinated protocols undergo packet losses at the application level, not necessarily because of transmission errors (recall from Figure 10 that success ratio is 1 for most protocols), but rather because of packets being discarded due to full queues. The worst performance in this sense is shown by T-Lohi, and in particular aT-Lohi with ACKs, where the combined effects of collisions and long protocol operations determine significant packet losses.

We conclude our discussion by analyzing in Figure 13 how throughput changes if the number of network nodes increases from 10 to 20. In this scenario, collisions are much more frequent; moreover, protocol operation times are expected to be longer due to the greater network activity causing, e.g., DACAP to silence more nodes during the RTS/CTS exchange, and T-Lohi to incur longer backoffs. Even ALOHA (both with and without ACK) experiences lower throughput, in this case because of the repeated collisions. However, it is worth remarking that the general observations made above are still valid: coordinated protocols still perform better at low traffic, before throughput starts to decrease, and ALOHA with no ACK achieves the greatest throughput. Furthermore the relative order of the curves is almost the same, as DACAP with no ACK achieves the best performance of all coordinated protocols; with 20 nodes, however, the difference among all other protocols is less pronounced. It should also be observed that ALOHA with ACK does not experience the fairly good performance it reached with 10 nodes, as it never reaches 0.5 (against 0.8 with 10 nodes) and its performance decreases faster.

V. Conclusions

Our contribution in this paper is two-fold. We have outlined the concept, structure and motivation behind a network simulator for underwater acoustic networks blending the Bellhop tool (for propagation modeling) and the ns2-MIRACLE network simulator. We have also presented the extensions we implemented to the simulator, especially in terms of data to be supplied to the propagation modeler, which required interfacing with oceanographic databases. Then we employed this simulator to carry out a comparison among protocols specifically designed for underwater networks.

Our results confirm that environmental parameters may have a significant impact on network performance; since Bellhop takes such parameters into account when modeling propagation, it is a very useful tool for helping with the evaluation of network performance as well. In light of the greater accuracy of our simulator, we also discussed whether or not resorting to a greater degree of node coordination (through more complex signaling patterns) yields better performance. In this regard, we conclude that a deeper coordination indeed helps, but only insofar as traffic is sufficiently small or the network is not too dense. Conversely, the lower overhead and more prompt timing of random access (i.e., ALOHA) translates into generally better performance.

Acknowledgment

This work has been supported in part by the NATO Undersea Research Center under contracts no. 40800700 (ref. NURC-010-08) and 40900654.

The authors would like to thank those people at NURC who have contributed to the genesis and improvement of the ideas behind this work: Kim McCoy and Giovanni Zappa, for inviting the authors at the workshops where the JANUS protocol was being defined (see [30]) as well as for the many ensuing discussions and useful tips; Alessandro Berni, Diego Merani, Robert Been and again Kim McCoy, for being first points of contact within NURC.

References

[1] B. Benson *et al.*, "Design of a low-cost acoustic modem for moored oceanographic applications," in *Proc. of ACM WUWNet*, Los Angeles, CA, Sep. 2006.

[2] M. Martini *et al.*, "Field tests of acoustic telemetry for a portable coastal observatory," in *Proc. of MTS/IEEE OCEANS*, Boston, MA, Sep. 2006.

[3] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms," http://www.whoi.edu, 2005.

[4] "Teledyne benthos undersea systems and equipment," www.benthos.com.

[5] P. Qarabaqi and M. Stojanovic, "Statistical modeling of a shallow water acoustic communication channel," in *Proc. of IACM UAM*, Nafplion, Greece, Jun. 2009.

[6] F.-X. Sochelau *et al.*, "Characterisation of time-varying underwater acoustic communication channel with application to channel capacity," in *Proc. of IACM UAM*, Nafplion, Greece, Jun. 2009.

[7] M. Chitre and J. Potter and O. S. Heng, "Underwater acoustic channel characterisation for medium-range shallow water communications." [Online]. Available: http://www.arl.nus.edu.sg/objects/323.pdf

[8] R. Urick, *Principles of Underwater Sound*. McGraw-Hill, 1983.

[9] J. Rice *et al.*, "Evolution of seaweb underwater acoustic networking," in *Proc. MTS/IEEE Oceans*, 2000, pp. 2007–2017.

[10] H. Schmidt *et al.*, "GOATS 2008: autonomous, adaptive multistatic acoustic sensing," Massachusetts Institute of Technology, Tech. Rep., 2008. [Online]. Available: www.onr.navy.mil/sci_tech/32/reports/docs/08/oaschmi3.pdf

[11] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM Mobile Comput. and Commun. Review*, vol. 11, no. 4, pp. 34–43, Oct. 2007.

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

[13] ns2 Network Simulator, http://www.isi.edu/nsnam/ns/.

[14] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "NS2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2," in *Proc. of ACM NSTools*, Nantes, France, Oct. 2007.

[15] "Ns2-miracle source code download page." [Online]. Available: http://telecom.dei.unipd.it/pages/read/58/

[16] L. G. Roberts, "ALOHA packet system with and without slots and capture," *ACM SigComm Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[17] B. Peleato and M. Stojanovic, "Distance aware collision avoidance protocol for ad hoc underwater acoustic sensor networks," *IEEE Commun. Lett.*, vol. 11, no. 12, pp. 1025–1027, Dec. 2007.

[18] A. Syed, W. Ye, and J. Heidemann, "Comparison and Evaluation of the T-Lohi MAC for Underwater Acoustic Sensor Networks," *IEEE J. Select. Areas Commun.*, vol. 26, pp. 1731–1743, Dec. 2008.

[19] C. Petrioli, R. Petroccia, and M. Stojanovic, "A comparative performance evaluation of MAC protocols for underwater sensor networks," in *Proc. MTS/IEEE Oceans*, Québec City, Canada, 2008.

[20] N. Parrish, L. Tracy, S. Roy, P. Arabshahi, and W. Fox, "System design considerations for undersea networks: link and multiple access protocols," *IEEE J. Select. Areas Commun.*, vol. 26, no. 9, pp. 1720–1730, Dec. 2008.

[21] F. Guerra, P. Casari, and M. Zorzi, "MAC protocols for monitoring and event detection in underwater networks employing a FH-BFSK physical layer," in *Proc. of IACM UAM*, Nafplion, Greece, Jun. 2009.

[22] ——, "World Ocean Simulation System (WOSS): a simulation tool for underwater networks with realistic propagation modeling," in *Proc. of WUWNet 2009*, Berkeley, CA, Nov. 2009, to appear.

[23] "World ocean atlas." [Online]. Available: www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html

[24] "General bathymetric chart of the oceans." [Online]. Available: www.gebco.net

[25] "National geophysical data center, seafloor surficial sediment descriptions." [Online]. Available: http://www.ngdc.noaa.gov/mgg/geology/deck41.html

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

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

[28] M. Molins and M. Stojanovic, "Slotted FAMA: a MAC Protocol for underwater acoustic networks," in *Proc. MTS/IEEE Oceans*, Singapore, Sep. 2006.

[29] K. McCoy, "JANUS: from primitive signal to orthodox networks," in *Proc. of IACM UAM*, Nafplion, Greece, Jun. 2009.

[30] "JANUS workshop proceedings." [Online]. Available: http://nrcsp.zftp.com/users/janus-tmp