Performance Evaluation of Forwarding Protocols for the RACUN Network

Cristiano Tapparello[‡], Paolo Casari[‡§], Giovanni Toso[‡], Ivano Calabrese[§]
Roald Otnes[⋆], Paul van Walree[⋆], Michael Goetz[#⋄], Ivor Nissen[⋄], Michele Zorzi[‡§]

[‡]Department of Information Engineering, University of Padova, via Gradenigo 6/B, 35131 Padova, Italy
[⋆]Norwegian Defence Research Establishment (FFI), P.O. Box 115, NO-3191 Horten, Norway
[⋄]Research Department for Underwater Acoustics and Marine Geophysics (FWG), 24148 Kiel, Germany
[#]Fraunhofer Institute for Communication, Information Processing and Ergonomics, 53343 Wachtberg, Germany
[§]Consorzio Ferrara Ricerche, via Saragat 1, 44122 Ferrara, Italy

ABSTRACT

In this paper, we discuss the performance of different network protocols for RACUN, a European Defence Agency project with the objective of demonstrating ad hoc underwater networks for multiple purposes related to security. The RACUN network is designed for long-range communications over areas of large size, hence a very important role is played by the network protocols employed. We show that the channel realizations observed in typical scenarios and the physical layer schemes available in the project lead to significant bit error rates. Therefore, the protocols that offer some inherent form of redundancy, as in the case of flooding-based protocols, tend to yield better performance than protocols based on the exchange of signaling traffic. In support of this statement, we simulate two scenarios for the RACUN network over channel realizations that are statistically derived from real channel measurements. Our results provide insight on the advantages and drawbacks of the different packet forwarding strategies, and confirm that flooding-based approaches perform better. In addition, we prove how splitting packets into multiple fragments to match the modem's maximum service data unit significantly limits the performance.

Categories and Subject Descriptors

C.2.0 [**Communication/Networking and Information Technology**]: General—*Data communications*; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms

Measurement, Performance

Keywords

Underwater acoustic networks, physical layer, protocol design, simulation, RACUN

1. INTRODUCTION

Robust Acoustic Communications for Underwater Networks (RACUN) is a 4-year multi-national research effort under the European Defence Agency (EDA). The project's main objective is to develop and demonstrate the capability to establish an underwater ad hoc acoustic network for multiple purposes with moving and stationary nodes. Scenarios of interest for RACUN include establishing and maintaining a safe operating area, performing intelligence, surveillance, and reconnaissance (ISR) missions, or enact mine countermeasures via multiple AUVs.

The RACUN project focuses on many aspects of underwater communications and networking, including channel characterization and modeling, noise modeling, communication scheme design, network protocol design, and simulation. After one sea trial campaign (ST1), where each participating nation independently performed channel probing experiments, the advancements and findings of RACUN were tested at sea during ST2 in April 2013, and are going to be finally showcased during ST3 in May 2014.

RACUN is one of the few projects that implement a complete workflow from channel characterization to sea trials, involving a number of coordinated software packages and heterogeneous platforms (powerful programmable modems, limited embedded platforms attached to commercial modems, AUVs and bottom nodes). This means that *i*) the channel characteristics measured during ST1 are statistically reproduced in order to perform several simulations for physical layer transmission schemes, employing any number of realizations that are coherent with ST1 measurements; *ii*) the performance of the physical layer schemes are summarized to yield bit and packet error rate figures that account for significant propagation effects such as delay spread, Doppler shift, Doppler spread and reverberation; *iii*) the resulting figures are employed for the simulation of networking protocols before going to sea; *iv*) the simulation software is designed so that the same code is straightforwardly reused both in simulations (with an emulated modem) and in sea trials (where the modem is actually present).

To the best of the authors' knowledge, such a complete approach has never been observed among the international projects that dealt with underwater networking so far. In fact, many research efforts focused on specific elements of underwater channel modeling, communications, networking or applications. As an example, UAN [1] and CLAM [2] focused mostly on the networking and application layer aspects of underwater networks. In particular, UAN designed and studied the interactions between multiple AUVs and bottom nodes mediated by acoustic communications, whereas CLAM focuses on network protocol design for localization, remote data retrieval and alarm event detection in static equipment monitoring networks interacting with at most one AUV. Channel characterization aspects are present in both projects but, unlike in RACUN, they are employed for basic propagation and Signal-to-Noise Ratio (SNR) prediction rather than for the simulation of physical transmission schemes. The pioneering SeaWeb [3] project focused on the creation and medium-term evaluation of an underwater network using simple MAC and routing protocols, and was among the first to show that underwater networking is in fact possible. Several un-

WUWNET'13, November 11–13 2013, Kaohsiung, Taiwan

http://dx.doi.org/10.1145/2532378.2532401.

derwater channel measurement campaigns have been carried out so far, recent examples including SPACE'08 and KAM'11. The wealth of acoustic data obtained in these campaigns has also been analyzed for networking related purposes, e.g., to yield estimates of the acoustic channel correlation and its evolution over time (e.g., see [4, 5]). However, these results were not yet exploited to design network protocols for experiments in dedicated sea trials. The non-exhaustive examples of efforts listed above suggest that a typical approach in research projects is to decouple channel measurements and network simulations from what is actually done in sea trials. Instead, a very significant effort in RACUN has been devoted to closing the loop from channel measurements to field experiments, making it possible to compare the trends of the simulation results against the outcomes of sea trials.

In this paper, we present the RACUN software framework, highlighting the aspects related both to the physical layer simulations and to the network protocol simulations. We then carry out simulation campaigns in two scenarios of interest for RACUN: the first one relates to the network topology deployed at sea for the RACUN ST2 experiments; the second one incorporates a reconnaissance and intruder detection scenario where a mobile node is tracked by an AUV with the help of detections and messages from a number of bottom nodes deployed in barriers. We simulate four different network protocols over two different physical layer models by realistically reproducing the modem behavior. For instance, the latter implies that the messages sent must match the maximum transmission size allowed by the modem, or be fragmented in case they do not fit, e.g., due to a large header.

The remainder of this paper is organized as follows. Section 2 introduces the software modules employed for network simulations and the physical layer models; Section 2.3 provides an overview of the routing protocols compared in this paper; Section 3 describes the simulation scenarios; Section 4 discusses the results for each scenario; Section 5 concludes the paper.

2. SIMULATION SOFTWARE

2.1 DESERT Underwater

The simulations presented in this paper employ an evolution of the DESERT Underwater libraries presented in [6], integrated with several RACUN-specific modules that make it possible to reuse exactly the same code in both RACUN simulations and sea trials.

The DESERT libraries are based on the well established ns2 and NS-MIRACLE [7] simulation software, customized with additional functionalities, protocols and modules suitable for the simulation of underwater networks. Moreover, DESERT Underwater provides modules that connect the network simulation stack to modem interfaces, so that the same software employed for simulation can be reused in real-world experiments. To achieve this, the internal messages and data structures of the simulator are converted into bit streams that the modem can transmit or receive. Once the rules for bit stream creation have been defined, modem-specific interfaces perform the translation between the two environments in a way that is transparent to the protocol designer. DESERT inherits the modular structure of NS-MIRACLE, in that the user is allowed to specify the elements of a full protocol stack and to assign them an order, akin to the ISO/OSI layer classification. The modules communicate through `sendUp()` and `sendDown()` methods, which maintain the separation among different modules and protocols. In addition, to facilitate cross-layer design, the modules are also allowed to communicate directly via cross-layer messages.

While the general structure of the framework is discussed in [6], several improvements and extensions have been implemented to support RACUN simulations and sea trials. Such extensions are best described by inspecting the protocol stack employed in the present work. Specifically, every node is provided with an instance of the following modules, in a top-down order:

- Application layer:
  1. GUWAL application layer
  2. GUWAL wrapper
  3. Application-DESERT interface
- Transport layer:
  4. UDP
- Routing layer:
  5. Multicast filter
  6. Routing/forwarding module
- Link, MAC and physical layer:
  7. Convergence layer
  8. Adaptation layer
  9. DESERT-Physical interface
  10. Modem module (only simulation)
  11. Physical layer module (only simulation)

In the list above, the *GUWAL application layer* forms messages compliant with the Generic UnderWater Application Language format [8]. These general-purpose messages include, e.g., status reports, position reports, detection information for sensor-equipped nodes, as well as reconfiguration messages for several components of an underwater node, from the acoustic modem up to the network protocols in use. GUWAL also includes operational addresses designed to be defined by the navies, i.e., 6-bit numbers where the first 2 bits identify the type of node (i.e., bottom node, surface node, acoustic/radio gateway node, or AUV) and the other 4 bits identify a unique node or group of nodes within the specified type. The *GUWAL wrapper* and the *Application-DESERT interface* convert the information inside GUWAL messages into data structures that can be processed easily inside NS-MIRACLE, and inject it inside the software protocol stack, respectively. The *UDP* module provides basic transport layer functions in terms of port multiplexing and filtering of duplicate packets forwarded from the lower layers up to the application layer.

The *Routing/forwarding module* manages all actions required to forward packets to their destination, and is the core of the analysis presented in this paper. The protocols considered in our study are described in Section 2.3. Depending on the specific scenario and on the application requirements, a node may need to send a packet in multicast to a group of nodes, e.g., to nodes sharing the same GUWAL operational address. In this perspective, the *Multicast filter* module interacts with the routing module, by stopping packet replication once a packet reaches the prescribed multicast group.

The *Adaptation layer* (AL) is the core of the serialization process that converts the internal NS-MIRACLE structures to bit streams, and vice-versa. The module performs three basic functions: i) bit stream creation, based on rules defined by each module included in the node; ii) packet fragmentation, to match the maximum Physical-layer Service Data Unit (PSDU) size allowed by the acoustic modem hardware; iii) re-assembly of received fragments into full packets, that can be re-converted to internal NS-MIRACLE data structures. The *Convergence layer* works together with the AL to avoid the serialization of data that can be inferred from other fields. This is of particular importance in RACUN, due to the small PSDU allowed by some physical layer schemes. The packets (or fragments) ready to be transmitted are managed by the *DESERT-Physical interface*, a module that implements basic MAC functionalities (ALOHA in this case), drives the modem underneath, and reacts to its messages and feedback. For the simulations presented in this paper, the modem has been replaced by a module that emulates its behavior and interactions with the DESERT-Physical interface. The actual packet transmission and reception is finally simulated using two different methods, as detailed in the next subsections.

2.2 Physical Layer

The simulations presented in Section 4 are based on two different physical layer models. The first is the model already available in DESERT, which is based on simple, empirical equations for path loss and noise, and on equations for computing the bit error rate assuming a Binary Phase-Shift Keying (BPSK) modulation scheme. The second one is a more realistic model based on the statistical reproduction of measurements collected during RACUN's ST1. We describe each model in the following subsections.

2.2.1 *Empirical attenuation, noise, interference and bit error probability model*

The standard model available in DESERT for computing the error probability of a packet transmission is based on the empirical equations for path loss and noise in [9, 10]. Starting from a user-defined value for the transmit source level (SL) $P = 10^{P_{\text{dB}}/10}$, where P_{dB} (in dB re μPa^2) refers to a distance of 1 m from the source, DESERT computes attenuation as

$$10 \log_{10} A(d, f) = b \cdot 10 \log_{10}(1000d) + d \cdot a(f) \,, \qquad (1)$$

where b is the spreading factor (set to 1.7 in the present work), d is the distance between the transmitter and the receiver (in km), f is the carrier frequency of the transmission and $a(f)$ is the Thorp absorption factor for sea water (in dB/km).[1] Assuming a received packet is not affected by interference from concurrent transmissions, DESERT computes the signal-to-noise ratio (SNR) as follows [10, Eq. (8)]:

$$SNR(d, f) = \frac{P}{A(d, f)N(f)B} \,, \qquad (2)$$

where $N(f)$ (in μPa^2/Hz) is as in [10, Eq. (6)] and B is the system bandwidth in Hz. Note that (2) approximates noise as an AWGN process, with a constant power spectral density (psd) over the bandwidth B, equal to that computed at the carrier frequency f. The packet error rate (PER) is computed through the BPSK bit error equation and assuming independent bit errors over a packet of length L. Therefore,

$$PER_n(d, f) = 1 - \Big(1 - 0.5 \,\text{erfc}\big(\sqrt{SNR(d, f)}\,\big)\Big)^L \,, \qquad (3)$$

where n stands for "noise." At this point DESERT flips a coin with probability $PER_n(d, f)$ and decides whether the packet is correct or not. This information is passed on to the upper layers of the protocol stack, which will decide how to process or discard the packet. The resulting PER as a function of the distance d is shown in the left plot of Fig. 4, for the case of 128 bits per packet.

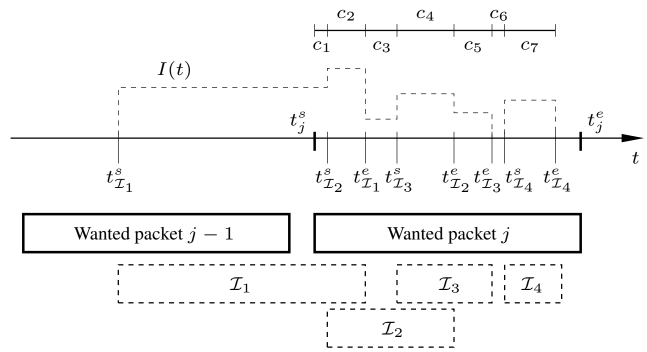

Figure 1: Chunk interference model implemented in DESERT Underwater.

In the presence of interference, DESERT leverages on the capability of NS-MIRACLE to track the typically time-varying interference power due to concurrent transmissions $I(t)$ (in μPa^2), in order to divide a received packet in chunks where the interference is constant. With reference to Fig. 1, assume that the reception of packet $j-1$ has finished and that DESERT is about to start the reception of packet j, which lasts from t_j^s to t_j^e where s and e stand for "start" and "end," respectively. Assume also that the interference from four concurrent transmissions is superimposed to the reception of the packet. These interfering packets are labeled $\mathcal{I}_1$ to $\mathcal{I}_4$ in Fig. 1. DESERT tracks the start and end of each interfering packet and sums their respective received powers to yield the overall interference level over time. Based on the start and end time of each interfering reception, DESERT divides packet j into chunks where the average interference level is constant. These chunks are labeled c_k, where $k = 1, \dots, \mathcal{C}$, and $\mathcal{C} = 7$ for packet j in Fig. 1. Each chunk is defined as a time interval $c_k = [c_k^s, c_k^e)$ where the interference is constant, or $I(t) = I(t_k)\ \forall t_k \in c_k$. In addition, $c_{k-1}^e = c_k^s$, $k = 2, \dots, C$. For each chunk c_k, DESERT computes the signal-to-interference-and-noise-ratio (SINR) assuming interference is also a white Gaussian process. Denoting the interference level during chunk c_k as $I(t_k)$ with $t_k \in c_k$, we have

$$SINR_k(d, f) = \frac{P}{A(d, f)N(f)B + I(t_k)} \,, \qquad (4)$$

which is plugged into (3) along with the number of bits L_k within chunk k, to yield the probability $PER_k(d, f)$ that chunk k is incorrectly received. The packet is then declared correct if and only if all chunks are correct, hence

$$PER_i(d, f) = 1 - \prod_{k=1}^{\mathcal{C}} \big(1 - PER_k(d, f)\big) \,, \qquad (5)$$

where i stands for "interference."[2] In order to understand whether a packet is corrupted due to noise or interference, in the presence of the latter DESERT first tests whether the packet is lost due to noise, and if such test is passed, it flips another coin to test if any chunks have been corrupted by interference. The packet is finally declared correct only if both tests are passed.

2.2.2 *Modulation-specific Lookup Tables*

Lookup tables (LUTs) with error rates based on *in situ* channel measurements were realized according to the procedure sketched

[1] It should be noted that (1) physically only makes sense for spherical spreading ($b = 2$). A term is missing for $b \neq 2$, which gives rise to a systematic error. This can be illustrated by considering the case of cylindrical spreading ($b = 1$). There may exist a range interval where the *spreading* is cylindrical, but the computation of the *transmission loss* has to consider the propagation from source to receiver. Close to the source, the spreading is always spherical. A more realistic transmission loss estimate for cylindrical spreading reads $10 \log_{10} A(d, f) = 10 \log_{10}(1000d) + 10 \log_{10}(H/(2 \sin \theta)) + d \cdot a(f)$, where H is the waveguide depth in m, and θ is the angle below which the sound becomes trapped in the waveguide. Even in the case that all sound becomes trapped ($\theta = \pi/2$), (1) underestimates the transmission loss by a term $10 \log_{10}(H/2)$. The error for $b \neq 2$ has carried over to a vast literature that studied underwater networks using the sonar equations in [9, page 102] and the SNR model in [10]. For appropriate range intervals, however, the error is a constant term. Hence its effect is that the simulations are still valid for the typical ranges considered in this paper, but for a different source level than the value of P defined above. Quantifying the missing term(s) in (1) for typical RACUN scenarios is left for further study. Such terms have been computed analytically for selected cases in [11, 12]. An alternative attenuation model has been presented in [13].

[2] The modeling of coding schemes is a planned extension for a future release.

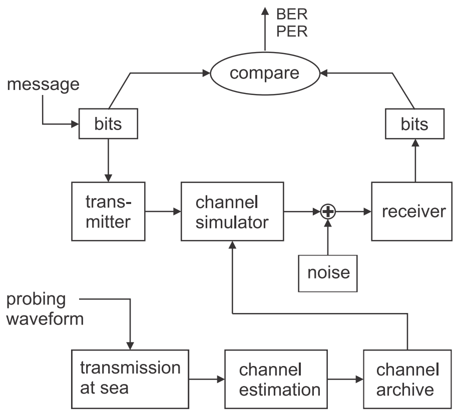

Figure 2: Methodology for creating lookup tables based on measured channels.

in Fig. 2. A set of channel probe signals was distributed to the RACUN partners, who transmitted these signals during national sea trials in various environments. Recorded data were processed with a common channel estimation tool to obtain the time-varying impulse response. The combined efforts resulted in a RACUN channel archive with about 60 different tracks spread over five geographical areas. Multipath propagation and Doppler effects are naturally included in these channels. The Doppler effects include sea-surface interactions, ocean volume fluctuations, and platform motion. Platforms used during the data collection are surface ships, bottom nodes, and an AUV.

Archived channels are used to drive the Mime channel simulator [14]. Once a channel archive has been created, the procedures of the upper half of Fig. 2 can be applied to any physical layer scheme that is compliant with the frequency band of the channels in the archive (4–8 kHz for RACUN). Colored noise is added to the simulator output, using the model in [15]. "Sea state 3" noise is used, which is colored and non-Gaussian. Channel simulations are performed at SNR values ranging from -12 to 30 dB, in steps of 3 dB. At each SNR value, the waveform delivered by the transmitter is filtered with 100 realizations of each measured channel, and for each channel realization a different noise realization is added. The channel simulator, and the procedures to create different realizations of a measured channel, are described and validated in [14].

The result of the simulations in Fig. 2 is a LUT of PER vs. SNR. In RACUN, such LUTs are available for several communication schemes, using different packet sizes and data rates, and 60 channels. However, in the present paper we only consider a set of channels measured in the Mediterranean Sea, and one modulation scheme. This scheme is filtered multi-tone (FMT) [16, 17], which uses burst communications to convey 128 bits at an effective data rate of 340 bit/s. The signal duration is therefore 0.38 s. An example of a LUT is shown in Fig. 3. Note that the PER does not go to zero at high SNR, because the communication performance becomes limited by delay spread and Doppler spread.

Compared with existing approximations for the physical layer in acoustic network simulations, the main advantage of the RACUN LUT method is that all channel dynamics are taken into account. Unless the *in situ* measured channel is (severely) overspread, the accumulated effect of channel estimation errors and channel simulation errors is small [14]. The main disadvantage of the RACUN approach is also evident: physical layer statistics are only available for the conditions pertaining to the initial data collection. For instance, it is not possible to extract PER figures for a distance of 10 km, if there was no channel measurement over 10 km.

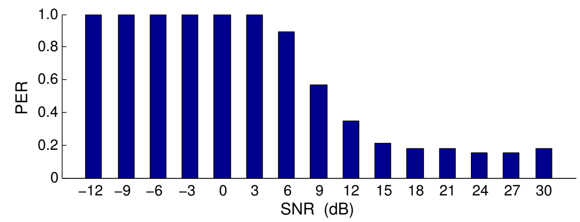

Figure 3: Example of PER vs. SNR for FMT. (Mediterranean Sea, 850-m range.)

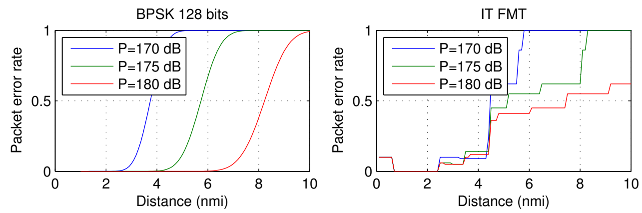

Figure 4: Left: PER vs range using the simple BPSK-based model. Right: PER vs range using modulation-specific LUT (FMT, Mediterranean Sea)

When the LUTs are used by DESERT, the SNR for a given range is determined in the same manner as described in Sec. 2.2.1, and nearest-neighbor interpolation is applied in the two-dimensional matrix of PER as a function of range and SNR. Other interpolation schemes (e.g., linear) are not more suitable since the underlying function is unknown and highly nonlinear. The resulting PER vs. range is shown in the right panel of Fig. 4. Using this PER for the desired range, DESERT flips a coin and decides whether the packet is correct or not.

Collision LUTs have also been created. This is achieved by considering two signals, A and B, which are packets of the same modulation type, but carry different messages. A arrives first, B second. Two parameters are introduced to specify the strength of the interference. The signal-to-interference ratio (SIR) gives the ratio of the power of signal A to the power of signal B. The overlap ω says how much overlap there is in time between A and B. $\omega = 50\%$ means that the signal B arrives at halfway reception of signal A; $\omega = 100\%$ means that the two signals A and B arrive at exactly the same time. To improve the statistics, the LUT is averaged over different realizations of the bit stream carried by A and B. The LUTs are hooked to DESERT by having the physical layer module compute the SNR, the SIR and the overlap between interfering packets. For the latter two parameters, consider again packet j in Fig. 1. The SIR for packet j is computed as

$$SIR = \frac{P_j}{\int_{t_j^e - \omega(t_j^e - t_j^s)}^{t_j^e} I(\tau)\,\mathrm{d}\tau}, \tag{6}$$

where P_j is the received power for packet j and the denominator is the average interference level, computed over the duration of packet j's reception. The computation of the overlap ω is approximated using the time of arrival of the first interfering packet. In Fig. 1, the first interferer is $\mathcal{I}_1$, hence $\omega = (t_j^e - \max\{t_j^s, t_{\mathcal{I}_1}^s\})/(t_j^e - t_j^s) = 1$. If there were no $\mathcal{I}_1$ and $\mathcal{I}_2$, the first interferer would be $\mathcal{I}_3$, and $\omega = (t_j^e - \max\{t_j^s, t_{\mathcal{I}_3}^s\})/(t_j^e - t_j^s) \approx 2/3$. Note that to select the appropriate entries of the LUT, we approximate the overlap by considering the starting epoch of the first interferer, and by assuming that the interference continues until the end of the packet. If this is not the case, our model will yield a smaller SIR per (6), thereby increasing the probability of error as expected.

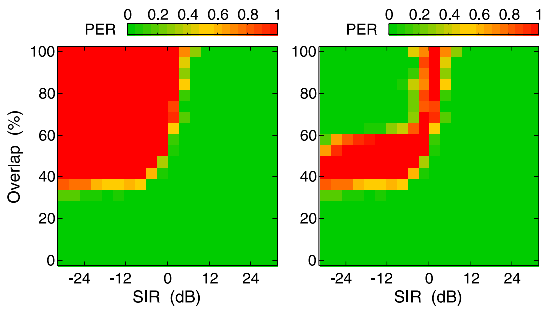

Figure 5: Collision LUT for FMT. Left: PER for packet A. Right: PER for packet A or B.

The collision LUT for FMT is given in Fig. 5. The left panel gives the PER under the assumption that collisions are handled by always treating the first arriving packet (A) as the signal and the second one (B) as interference.[3] The right panel gives the PER under the assumption that, depending on the overlap and SIR, an actual receiver may synchronize on packet B, hence A becomes the interferer. This is the green area in the top-left corner. In the DESERT simulations, the former behavior is implemented, hence the PER of the left panel is used. The collision LUTs use an ideal channel with no delay-Doppler spread. In principle the RACUN physical layer simulations could combine acoustic channels, collisions, and additive noise, to create LUTs with a higher dimensionality, but this becomes a cumbersome and time-consuming task.

2.3 Routing Protocols

2.3.1 GUWMANET

The first protocol named *Gossiping in Underwater Mobile Adhoc Networks* (GUWMANET) was developed in cooperation between the FWG and Fraunhofer FKIE. It is specifically designed for underwater networks and reduces the network overhead to a minimum. Beside the operational source and destination addresses included in the application data of GUWAL, it needs only an additional transmitter and last hop address (which is a unique nickname in the local neighborhood) for routing purposes.

The route establishment operates on the principle of reactive routing protocols. The first message is flooded through the network. Each node repeats the message and sets the last hop field to the network address of the first node it received the message from. If a node overhears that it was selected as last hop by one of its neighbors, it generates a temporary routing entry. After the message reaches one of the destination nodes, the node replies with an acknowledgement which is routed back with the temporary routing entries to the source. On the way back the route is confirmed for all further packets. If a route breaks, the route establishment is initiated again. More details of GUWMANET can be found in [18].

In Section 4 two variants of GUWMANET are used. Version A, without packet retransmissions, and version B, which allows a maximum of 3 repetitions (respectively after 30, 60 and 120 s) if no packet forwarding is overheard from nodes downstream.

2.3.2 DESERT Flooding

DESERT Flooding is a baseline flooding algorithm with a maximum Time-To-Live (TTL). The rules of the protocol are simple, in that a node that receives a packet checks i) whether the TTL of the packet is not zero and ii) whether the same packet has not been transmitted before within a user-defined time interval. If both conditions are met, the node plainly forwards the packet. To avoid reforwarding the same packet more than once, every node keeps track of the source address and ID of every forwarded packet. In turn, this will make it possible to perform check ii) above on an updated set of packets in transit through the node. Although simpler than the GUWMANET protocol discussed in Section 2.3.1, DESERT Flooding provides an interesting term of comparison, in that it fully exploits the DESERT framework, including packet fragmentation down to the PSDU size allowed by the physical layer modulation scheme in use. DESERT Flooding allows the user to define the TTL of flooded packets, which we set to 6 for the results of Section 4.

2.3.3 Dflood

Dflood is the name given to the protocol introduced in [19]. This protocol strives to reduce the number of duplicates in a flooding-based underwater acoustic communication network. Each node applies a backoff time before forwarding a copy of an incoming message, and this backoff time is adaptively increased each time the node overhears that another node also forwards a copy of the same message. If a node overhears more than a given number of copies of the same message being forwarded by other nodes, it will refrain from forwarding that message. The latter rule is called "counter-based scheme" in [20], and we combine it with an additional rule where the backoff time is adapted to the value of the counter. The complete rules of the protocol are described in [19], to which we refer the interested reader.[4] The parameter settings for the Dflood protocol used for the simulations in this paper are (with reference to [19] for parameter definitions): $T_{\text{min}} = 5$ s, $T_{\text{max}} = 65$ s, $T_{\text{Dupl}} = 20$ s, and $N_{\text{Dupl}} = 4$.

2.3.4 MSUN

MSUN stands for Multi-sink Source routing for Underwater Networks and is an improvement and extension of the SUN protocol [21]. The protocol conforms to the source routing paradigm, whereby the source of a data packet probes the network for routes to the destination, and authoritatively decides which route to use among possibly several detected paths. In detail, when a source generates packets for a given destination, the packets are put in a common FIFO queue. The first element of the queue is checked periodically. If a route for its destination is known, the packet is immediately sent via source routing. Otherwise, the source forms a Path Request packet which is flooded through the network. As the nodes forward the Path Request, they write their own address into it, so that each Path Request brings a record of the route it followed. Each Path Request is forwarded only once by each node, in order to avoid loops and to limit the number of transmissions.

The destination answers all received Path Requests that followed different paths with a different Path Reply packet, which is sent back through the same route of its related Path Request. This automatically eliminates routes containing asymmetric links. Upon receiving multiple Path Replies, the source chooses the best path according to some metric (e.g., the hop count of the route, or the minimum SNR experienced over the hops of the route) and stores the path. The next time the packet queue is checked, the route towards its destination will be known.

[3] In Fig. 1, this is the case, e.g., for packet j, which is locked on, whereas $\mathcal{I}_2$ is considered interference. The only exception to this rule is "residual interference:" referring again to Fig. 1, the receiver first locks on packet $j-1$ ($\mathcal{I}_1$ arrives later, and is this considered an interferer); since the receiver never locked on $\mathcal{I}_1$, it becomes a residual interferer for packet j: it contributes to decreasing the SIR and leads to $\omega = 100\%$.

[4] The rules in [19] should be augmented by one rule which was inadvertently left out of that paper: "Forwarding is delayed by a time, T_{Dupl}, when a forwarding from another node is overheard (with hop counter greater than that of the original reception)."

When a packet is forwarded, link layer ACKs are used to confirm the correct reception of a packet by the next hop. Incorrect packets can be retransmitted using a Stop-and-Wait ARQ policy. MSUN manages packet retransmissions in a cross-layer fashion, and employs the lack of expected ACKs to detect broken links along a path, to notify them to the source of a data packet and thereby decide whether a new route is to be searched from scratch. MSUN also provides a simple broadcast mode and a multicast mode, both based on DESERT Flooding. MSUN can be configured in several respects, including the number of retransmissions, the route search and maintenance timers (to tune the amount of control traffic injected in the network) and the period of the packet queue checking (to reduce the possibility of congestion or, on the contrary, to push more traffic over under-utilized channels). The latter period can be chosen a priori in small and controlled environments, or it can be adapted automatically by an internal algorithm.

For the results of Section 4, we set a maximum of 2 retransmissions per packet upon errors, a processing time of 60 s for the FIFO buffer, and a waiting time between subsequent path searches of 120 s. In case MSUN resorts to flooding, the TTL assigned to the packets is 6.

2.4 Discussion

The protocols that are going to be compared in Section 4 have been listed above in order of increasing complexity, in terms of both protocol design and amount of traffic to be generated in order to forward the packets to their destination. The latter aspect is of particular importance in our simulation study. From Section 2.2.2, we recall that the PSDU size of the FMT scheme considered in this paper is 128 bits. This PSDU size is fed to the DESERT framework, and used by the AL module for packet fragmentation and re-assembling (see Section 2.1). Of the four protocols described above, GUWMANET manages to fit one packet transmission into a single fragment. All other protocols require 2 fragments per packet to accommodate additional header field data. In addition, MSUN requires a single fragment for every signaling packet sent for path establishment and link layer ACKs. The immediate consequence of these facts is that the packet error rate for a single packet transmission is lower for GUWMANET than for the other protocols, because GUWMANET requires only one fragment per packet. The simulation results in Section 4 will reflect these issues.

3. SIMULATION SCENARIOS

In this Section, we describe two simulation scenarios used to evaluate the performance of the forwarding protocols of Section 2.3. To this end, we consider as a reference scenario the largest RACUN network that have been deployed so far. Moreover, we also provide some numerical results for a scalable size scenario. The latter is particularly useful to evaluate the impact of the number of nodes on the considered metrics.

3.1 Sea Trial 2 Scenario

The RACUN Sea Trial 2 (ST2) was the first international trial of the project. It was carried out in April 2013 with the objectives of performing system and network integration tests, testing specific network features (such as relaying, multi-hop, re-routing, collision recovery, etc.), as well as to collect experimental data for validation of network simulations for small networks. In this paper, we consider the largest RACUN network that has been deployed during ST2. The complete topology, with the corresponding node coordinates, is presented in Fig. 6, where the position of the nodes is represented to scale, in order to give some insight on the actual distances among the nodes. As an example, the maximum distance

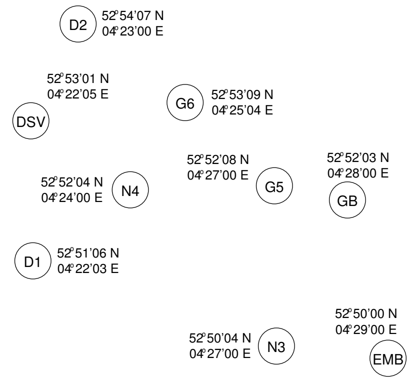

Figure 6: ST2 scenario. The circles are communication nodes. The Elisabeth Mann Borgese (EMB) is the ship node acting as the sink. GB is a gateway buoy.

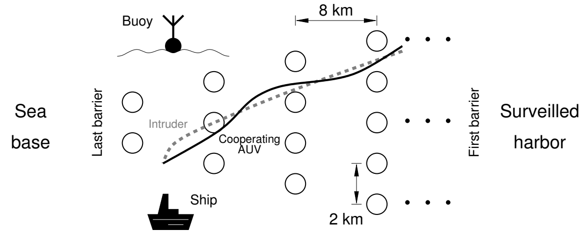

Figure 7: Topology of the barrier scenario.

between two nodes (i.e., between D3 and EMB) is approximately 5.5 nmi (10.2 km), while the minimum distance (i.e., between G5 and GB) is approximately 0.6 nmi (1.1 km). In this scenario, we assume that each node of the network sends a periodic status message to the EMB ship node, that acts as a sink. This scenario mimics the situation where a ship node is used to gather data from a distributed underwater network.

3.2 Barrier Scenario

The barrier scenario implements the idea to surveil a harbor with bottom-mounted sensor sensor nodes as introduced in [22]. Ships or submarines leaving the harbor are detected and reported via acoustic communications to a cooperating fleet at sea. The network is organized in barriers as can be seen in Fig. 7. A barrier is a set of nodes arranged in a line topology: the largest barrier is placed in front of the harbor in order to ensure the largest sensing coverage along the coast. The nodes are placed 2 km apart within the same barrier, and subsequent barriers are 8 km apart. While proceeding towards the sea base, the number of nodes per barrier is reduced by one. Note that the scenario is scalable: the number of barriers (and therefore the number of nodes) can be increased in order to analyze the scalability of the network protocols. In this study we start from 5 barriers (including 21 nodes) and simulate up to 20 barriers (230 nodes). All nodes can sense movement as well as relay data towards the cooperating fleet at the sea base. However, the traffic is generated only by the sensor nodes that detect a nearby intruder.

4. NUMERICAL RESULTS

In this Section we consider the scenarios described in Section 3, and compare the forwarding schemes of Section 2.3 via numerical results for both the empirical physical model of 2.2.1 and the modulation-specific LUTs described in Section 2.2.2. All the results in this section are obtained by fixing the transmit power to 175 dB re μPa^2 for the empirical physical model, whereas when lookup tables are used, we set a transmit power of 170 dB re μPa^2.

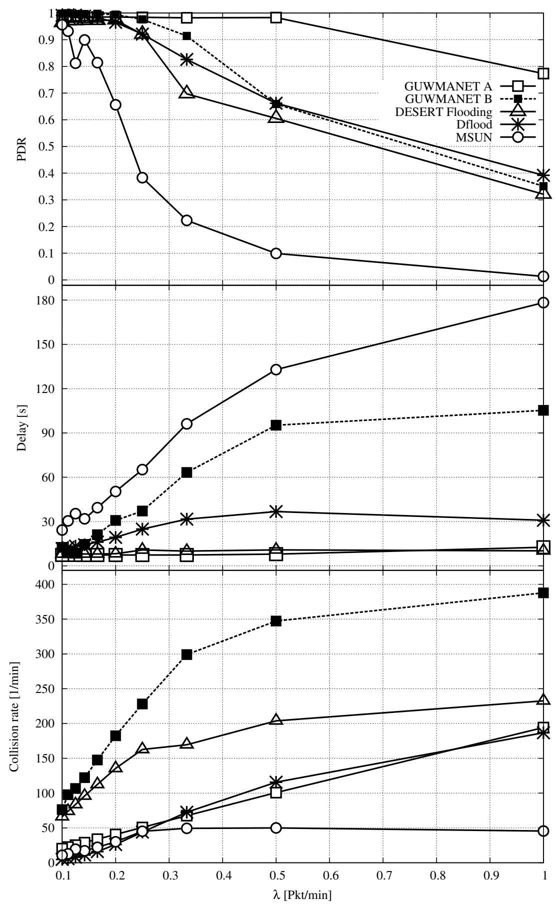

Figure 8: ST2 scenario, empirical physical layer model. PDR, end-to-end delay and collision rate as a function of the packet generation rate per node λ.

This choice has been made because, with these power levels, the two physical layer models have similar PER vs. distance performance (see Fig. 4).

We start by describing the results for the ST2 scenario, which is used as a reference to evaluate the performance of the different routing schemes under different network traffic loads. To this end, Fig. 8 shows the performance attained by the network as a function of the packet generation rate per node λ, when the empirical physical model is considered. The figure shows (top to bottom) the Packet Delivery Ratio (PDR), defined as the ratio of the number of data packets correctly received within 5 min from their generation epoch to the total number of generated packets; the average end-to-end delivery delay; the number of packet collisions per unit time. As expected, the PDR decreases for increasing λ while, at the same time, both the average end-to-end delay and the collision rate increase. This is because increasing the amount of traffic to be routed by the network makes it more likely for the packets to be discarded at the receivers due to interference or noise. We note that GUWMANET A performs best in terms of both PDR and delay, since it requires the transmission of a single fragment per packet, hence leading to the lowest amount of traffic. On the contrary, DESERT Flooding, Dflood and MSUN require the transmission of two fragments for each data packet, and therefore return worse performance. Moreover, GUWMANET B's retransmissions lead to several collisions, hence its performance is also worse than

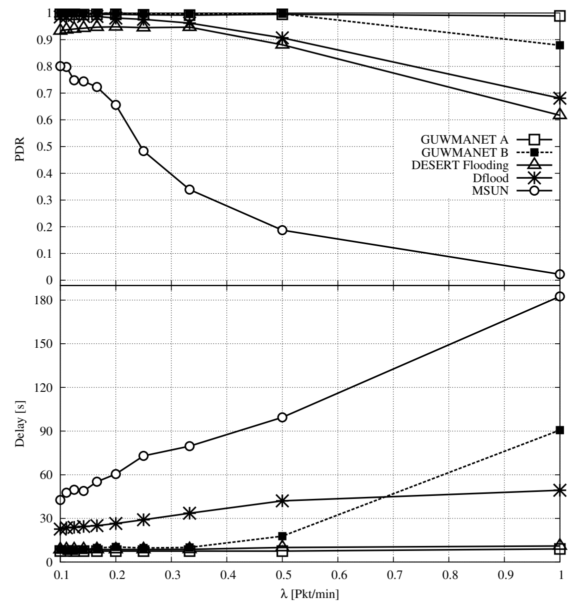

Figure 9: ST2 scenario with PER LUTs. PDR and end-to-end delay as a function of the packet generation rate per node λ.

GUWMANET A's (no retransmissions).

Similar considerations apply also to Fig. 9, where we show the PDR and the average delay as a function of λ, when the LUTs discussed in Section 2.2.2 are considered. We note that, in general, the performance of the different routing protocols is scaled up with respect to the case of the empirical physical model, thanks to a realistic model of collisions at the physical layer. For MSUN, this is also true for $\lambda = 0.5$ and $\lambda = 1$, however in general the protocol achieves the lowest PDR and the highest delay among the compared schemes. These results suggest that it is difficult for MSUN to find routes and use them reliably. The few collisions observed in Fig. 8 (the collision curve for the LUT case have been omitted because they are entirely analogous) suggest that, once routes have been established, MSUN transmits fewer packet replicas compared to flooding-based schemes (hence reducing the likelihood of collisions), but at the same time it cannot take advantage of redundant transmissions (hence its lower PDR).

Finally, we present some results for the barrier scenario. This scenario is used to analyze the performance of the routing protocols in larger networks. In Fig. 10, the PDR and the average end-to-end delay are shown as a function of the number of nodes in the network. Regarding the PDR, it can be seen that only GUWMANET shows good scalability, whereas the PDR of all other protocols decreases with increasing network size. This is mainly due to the higher number of collisions since, as mentioned before, all routing protocols need two consecutive successful fragment transmissions, whereas GUWMANET requires only one. Clearly, more transmissions increase the number of collisions and hence the packet error rate. In any event, the larger distances (with respect to those of the ST2 scenario) decrease the likelihood of collisions for GUWMANET B, which achieves the highest PDR.

We notice that the end-to-end delay increases when the number of nodes increases in Dflood and GUWMANET, due to the delivery of detection messages from nodes further away. In this scenario, the main contribution to the delay is yielded by the long propagation delays. Finally, we note that the delays of MSUN and DESERT Flooding are constant because most of the packets from barriers farther away are lost. The average delay of Dflood also becomes constant above 100 nodes. Detailed investigations show that this happens because packets are being lost due to a requirement of end-to-end delay being below 5 minutes in the simulations. As for the ST2 scenario, the results for the empirical physical model are highly correlated with the results obtained using the lookup tables and are therefore omitted due to lack of space.

5. CONCLUSIONS

In this paper, we discussed the performance of network protocols in scenarios of interest for RACUN, a 4-year multi-national research project aimed at demonstrating robust underwater networks in tactical and security scenarios. We described the complete RACUN simulation workflow, from channel measurements to network simulations. The latter employ packet error rate (PER) lookup tables (derived from channel measurements) for the modulation schemes in use, and then reuse the same protocol code employed in simulations for experimenting underwater networks at sea.

We showed that, due to the high PER experienced with the FMT modulation schemes considered in RACUN, flooding-based packet forwarding strategies yield better performance than other schemes requiring coordination via signaling messages. Among flooding-based schemes, GUWMANET is shown to yield the best PDR and delivery delay in the scenarios of interest, the main reason being that it requires the lowest number of fragments per data packet.

Acknowledgment

The work described in this publication was performed under a multinational four-year project with the name "Robust Acoustic Communications in Underwater Networks" (RACUN) under the EDA Project Arrangement No. B 0386 ESM1 GC.

GUWMANET and FMT are background information (provided by Germany) for the EDA RACUN project.

The authors gratefully acknowledge the collaboration of Arwid Komulainen to the development of the RACUN software, and the interaction with Loreto Pescosolido and Roberto Petroccia for the development of the NS-MIRACLE chunk interference model.

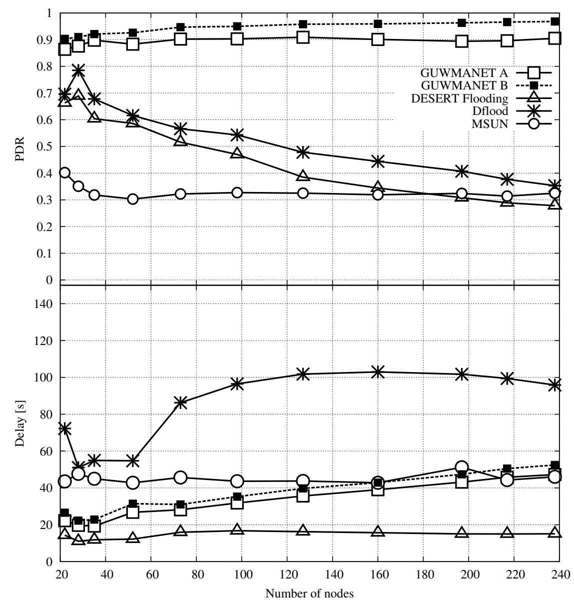

Figure 10: Barrier scenario with LUTs. PDR and end-to-end delay as a function of the number of nodes.

6. REFERENCES

[1] "UAN Project web site," retrieved: July 2013. [Online]. Available: www.ua-net.eu/Main_Page

[2] "CLAM Project web site," retrieved: July 2013. [Online]. Available: www.clam-project.eu

[3] R. K. Creber, J. A. Rice, P. A. Baxley, and C. L. Fletcher, "Performance of undersea acoustic networking using RTS/CTS handshaking and ARQ retransmission," in *Proc. MTS/IEEE Oceans*, Honolulu, HI , USA, Nov. 2011.

[4] B. Tomasi, J. Preisig, and M. Zorzi, "On the predictability of underwater acoustic communications performance: the KAM11 data set as a case study," in *Proc. ACM WUWNet*, Seattle, WA, 2011.

[5] ——, "On the spatial correlation in shallow water and its impact on networking protocols," in *Proc. MTS/IEEE Oceans*, Yeosu, Korea, 2012.

[6] R. Masiero, S. Azad, F. Favaro, M. Petrani, G. Toso, F. Guerra, P. Casari, and M. Zorzi, "DESERT Underwater: an NS–Miracle-based framework to DEsign, Simulate, Emulate and Realize Test-beds for underwater network protocols," in *Proc. MTS/IEEE Oceans*, Yeosu, Korea, May 2012.

[7] N. Baldo, M. Miozzo, F. Guerra, M. Rossi, and M. Zorzi, "MIRACLE: The Multi-Interface Cross-Layer Extension of ns2," *EURASIP Journal on Wireless Communications and Networking*, Jan. 2010. [Online]. Available: http://www.hindawi.com/journals/wcn/2010/761792/cta/

[8] M. Goetz and I. Nissen, "Generic Underwater Application Language (GUWAL) - Specification of Tactical Instant Messaging in Underwater Networks," FWG, Kiel, Germany, Tech. Rep. WTD71 - 0070/2012 FR, Dec. 2012.

[9] R. Urick, *Principles of Underwater Sound*. New York: McGraw-Hill, 1983.

[10] 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.

[11] D. E. Weston, "Intensity-range relations in oceanographic acoustics," *Journal of Sound and Vibration*, vol. 18, no. 2, pp. 271–287, 1971.

[12] M. A. Ainslie, "Shallow water propagation for linear and bilinear sound speed profiles," in *Proc. UDT*, Wembley, UK, Jul. 1994.

[13] M. Zuba, Z. Jiang, T. C. Yang, Y. Su, and J. Cui, "An Advanced Channel Framework for Improved Underwater Acoustic Network Simulations," in *Proc. MTS/IEEE OCEANS*, San Diego, CA, Sep. 2013.

[14] R. Otnes, P. A. van Walree, and T. Jenserud, "Validation of replay-based underwater acoustic communication channel simulation," *IEEE J. Oceanic Eng.*, 2013, accepted for publication.

[15] F. Traverso, G. Vernazza, and A. Trucco, "Simulation of non-white and non-Gaussian underwater ambient noise," in *Proc. MTS/IEEE OCEANS*, Yeosu, Korea, May 2012.

[16] I. Nissen, "Schlüsseltechnologie für das Unterwasser Info Management," in *Proc. of DWT Conference*, Kiel, Germany, 2012.

[17] T. Radtke, M. Motz, and F. Berning, "Unterwasser- kommunikation mit Spezialkräften," in *Proc. of DWT Conference*, Kiel, Germany, 2012.

[18] M. Goetz and I. Nissen, "GUWMANET - multicast routing in underwater acoustic networks," in *Proc. MCC*, vol. 2, Warsaw, Poland, 2012.

[19] R. Otnes and S. Haavik, "Duplicate reduction with adaptive backoff for a flooding-based underwater network protocol," in *Proc. MTS/IEEE OCEANS*, Bergen, Norway, Jun. 2013.

[20] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. ACM Mobicom*, Seattle, WA, Aug. 2009.

[21] G. Toso, R. Masiero, P. Casari, O. Kebkal, M. Komar, and M. Zorzi, "Field experiments for dynamic source routing: S2C EvoLogics modems run the SUN protocol using the DESERT Underwater libraries," in *Proc. MTS/IEEE Oceans*, Hampton Roads, VA, Sep. 2012.

[22] M. Goetz *et al.*, "Jamming-Resistant Multi-path Routing for Reliable Intruder Detection in Underwater Networks," in *Proc. ACM WUWNet*, Seattle, WA, Sep. 2011.