

Received October 26, 2018, accepted November 19, 2018. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.2884090

Underwater Delay-Tolerant Routing via Probabilistic Spraying

PAOLO CASARI¹, (Senior Member, IEEE), SAIFUL AZAD², (Member, IEEE),
MARCO ZANFORLIN³, AND MICHELE ZORZI³, (Fellow, IEEE)

¹IMDEA Network Institute, 28918 Madrid, Spain.

²University Malaysia Pahang, Gambang, Kuantan 26300, Malaysia.

³Department of Information Engineering, University of Padova, 35122 Padova, Italy.

Corresponding author: Saiful Azad (e-mail: saifulazad@ump.edu.my).

This work has been supported in part by the FRGS project under Grant No. RDU160107 and by the RDU project under Grant No. RDU1703100.

ABSTRACT We propose underwater delay-tolerant routing via probabilistic spraying (UDTN-Prob), a routing protocol for underwater delay-tolerant networks based on the store-and-forward paradigm. Our protocol exploits limited statistical knowledge of the time between two subsequent contacts between pairs of network nodes in order to filter the packets injected into the network, so that only those with a sufficiently high chance of being delivered to their intended destination within a given deadline are actually transmitted. In addition, the foreseen duration of a contact is estimated via a preliminary packet exchange, so that the nodes get a fair share of the contact time to exchange their own data. The transmission is protected against channel-induced packet losses via an ARQ scheme modified to adapt itself to typical underwater transmission times and to the variation of round-trip times induced by node mobility. We simulate the protocol using the DESERT Underwater libraries, that make it possible to accurately reproduce the nodes' behavior and mobility patterns. Our results show that the proposed protocol achieves significantly better performance than Spray-and-Wait, which is currently the most typical choice among store-and-forward protocols. Moreover, we show that a 2-hop statistical knowledge of the node contact process yields marginally higher utility with respect to a simpler 1-hop knowledge, which is also much easier to collect or estimate.

INDEX TERMS Underwater networks, delay tolerant networking, simulation, WOSS, parameter optimization.

I. INTRODUCTION

OVER the last three decades, underwater acoustic communications witnessed a significant development, mainly due to the advancements in the design of underwater communication devices and of their corresponding networking protocols. As a consequence, a wide range of applications may be supported in the future, including, e.g., oceanographic expeditions, environmental monitoring, disaster prevention, assisted navigation, as well as coastal patrol and surveillance [1]–[4]. Different applications have different characteristics, and may require different network architectures. For instance, routing protocols designed for connected multi-hop networks are not suitable for networks of mobile nodes experiencing intermittent connectivity. When this is the case, the lack of a valid route from the packet source to its intended destination must be taken into account, and a Delay-

Tolerant Networking (DTN) approach may be preferable. In this paper, we propose a DTN routing protocol for mobile networks of Autonomous Underwater Vehicles (AUVs). Our target scenario assumes that the vehicles operate over large areas, and thereby experience intermittent connectivity and time-varying node-to-node propagation delays. This applies, for instance, to such applications as coastal patrol and surveillance [4], where we can envision a fleet of AUVs autonomously patrolling an area of interest, inspecting surface ships or underwater assets, and reporting to a shore- or ship-based control center when a contact with such center occurs. An important aspect of these scenarios is that data delivery is time-constrained, and thus it must be accomplished before a given deadline, after which the data loses value and can be discarded.

The routing protocols which establish complete end-to-

end routes before data transmissions [5]–[8] are not suited to DTNs due to the lack of connectivity; conversely, store-and-forward-based routing protocols are preferred. The latter approach prescribes that when a node receives a packet from its own application layer or from another node, it will store it until it gets an opportunity to forward it further on to other node(s), hoping that the relays will be able to forward the packet towards the destination. A common practice observed across several store-and-forward based routing protocols is to allow multiple copies of the same packet to circulate in the network, in order to maximize the chance of successful delivery. Such routing protocols are further classified as replication-based, and typically result in a higher chance that at least one packet replica is received within the delivery deadline. However, they also impose a higher replication overhead, which inconveniently wastes the already limited acoustic bandwidth, and increases the overall energy consumption of multiple nodes, which in turn may reduce the network lifetime. Some routing protocols take a different approach and do not replicate any packet. In this case, the nodes store packets locally and selectively forward them when opportunities arise. This kind of routing protocols are dubbed forwarding-based and tend to achieve a higher efficiency (due to the lack of replication), but also a lower packet delivery ratio, with respect to replication-based approaches (due to the higher chance that a packet is not delivered within its prescribed deadline). Ideally, a DTN routing protocol should be designed so that it will provide a high packet delivery ratio (as in replication-based approaches) with a limited overhead (as in forwarding-based approaches). Both in replication-based and in forwarding-based protocols, an important piece of information to be factored in routing decisions is the time that elapses between subsequent contacts among the same nodes. This is defined as the inter-contact time, and has been shown to have exponential distribution in a variety of practical cases [9]–[12]. Power-law distributions have also been observed in scenarios where human mobility is involved [13]–[15].

DTN protocols offer a number of advantages in several underwater scenarios. Consider the case of large-scale surveying, environmental monitoring and mapping over areas of several square kilometers, or the patrolling of safe underwater areas of similar extension. It is envisioned that such tasks will be carried out by groups of AUVs that will collaboratively execute parts of a common mission [16]–[18], while coordinating, exchanging relevant data, and most importantly, reporting periodically to a sink node. Some examples of these applications with a single AUV can be found in [19]–[24]. Due to the energy constraints typical of AUVs, it is unlikely that such devices will integrate bulky and energy-hungry long-range modems: they will more likely resort to shorter-range, lower-energy communications. In conjunction with the small number of AUVs usually deployed, and owing to their low movement speed (up to 4 m/s, most typically below 1 m/s), this determines a very erratically connected network. A DTN protocol is therefore

the only viable multihop communication solution in such a scenario. If properly designed, it can provide an effective data exchange scheme, whereby packets can reach the sink with sufficiently high probability, and without requiring an extra network node acting as data mule.

The design of DTN routing protocols for underwater networks is subject to a different set of constraints than terrestrial radio DTNs. Part of these constraints come from the characteristics of the underwater acoustic channel, part from the applications that build on top of the network connectivity. Examples of underwater channel characteristics that require special attention include *i*) a propagation speed on the order of 1500 m/s; *ii*) typical communication ranges from a few hundred meters to a few kilometers [25]–[28], *iii*) transmissions largely prone to errors, especially when operated over shallow-water acoustic channels [3], [29]; and *iv*) a transmit power roughly 10 to 100 times larger than idle listening power. The above features profoundly affect communications, and thus have a broad impact on underwater DTN protocol design. Items *i*) and *ii*) mean that typical packet communications incur long propagation delays, compared to the typical duration of packet transmissions: this results in ineffective channel sensing mechanisms [30], and in ineffective stop-and-wait automatic repeat query (ARQ) procedures, as the time between the transmission of a packet and the reception of its acknowledgment (ACK) would be significantly long. The significant errors that typically affect underwater transmissions (item *iii*) together with items *i*) and *ii*) require channel- and propagation delay-aware error control schemes, which turn the long propagation delay from a handicap into an opportunity to multiplex multiple transmissions, beyond common stop-and-wait mechanisms. Finally, item *iv*) demands that the DTN protocol keep the replication overhead limited, as every transmitted packet would increase the energy budget significantly. At the same time, there should exist a lightweight mechanism to coordinate transmissions and thereby avoid wasting energy with contending communications.

In this paper, we propose Underwater DTN with Probabilistic spraying (UDTN-Prob), a replication-based routing protocol for underwater DTNs which chooses which packets should be transmitted based not just on the duration of the contact between two nodes, but also on their chance to meet the final destination of a packet (the sink) in the future. More specifically, whenever an estimate of the sink meeting probability is available, at least in the form of the cumulative distribution function (CDF) of the inter-contact time between a node and the sink, UDTN-Prob can exploit it to single out the data packets with the greatest chance of being delivered in time. These packets are given priority during the data exchange that occurs upon a meeting. A distinctive result of our work is that knowing the statistics of the inter-contact time between any two nodes in the network only yields marginally better utility. Therefore, the nodes only need to store information about their own meeting statistics in order to take relaying decisions. A binary spraying technique

is employed in this phase [31] in order to reduce packet replication, and therefore the amount of overhead injected in the network.

A novel aspect of UDTN-Prob is that it seamlessly merges the two-way data packet exchange between two nodes with an error control scheme that employs time-division duplexing (TDD) to leverage the long propagation delays (compared to the packet duration) experienced in underwater acoustic channels, in line with the approaches presented in [32], [33]. Both probabilistic spraying and TDD-based error control contribute to the successful replication of data packets through multiple nodes, achieving a better chance of delivering packets within their prescribed deadline, compared to a typical choice for store-and-forward DTN routing, namely Spray-and-Wait [31]. UDTN-Prob is not restricted to a specific mobility pattern, and only requires rough statistical knowledge about the node encounters.

The remainder of the paper is organized as follows. In Section II, we outline some previous works on DTN routing protocols. We describe our proposed routing protocol in Section III. In Section IV, we describe the simulation scenarios and the results of our simulation campaign. Finally, we draw some concluding remarks in Section V.

II. RELATED WORK

As already mentioned in Section I, DTN routing protocols can be generally classified as replication-based or forwarding-based. Because the protocol presented in this paper is a replication-based protocol, we are going to discuss some relevant replication-based approaches for routing in terrestrial or underwater DTNs. Epidemic routing [34] is one such protocol, based on the massive replication of data packets to each newly discovered contact that does not already own a copy of the same packets. Therefore, similar to flooding, epidemic routing is likely to achieve the best packet delivery ratio among repetition-based approaches, albeit at the price of a very large replication overhead.

Other routing protocols have been proposed which limit the replication overhead. For instance, MaxProp [35] prioritizes the packets to be transmitted or dropped upon contact with a peer based on a number of parameters, such as the packet generation time, and the lists of previous encounters. It incorporates a replication-limiting mechanism, but still experiences a considerable overhead due to its flooding-based nature. In Spray-And-Wait (SAW) [31], the replication of each packet is restricted to a fixed number of copies. In the vanilla version of SAW, only the packet source can replicate, whereas the binary version allows intermediate relays to also replicate packets: the maximum number of replicas allowed is evenly split between the current and the next relay. A more recent version [36] considers that overdue contacts are more likely to happen in the future. Therefore, those nodes that did not encounter the destination for a longer time become preferred relays.

The Resource Allocation Protocol for Intentional DTN (RAPID) [37] also limits replication by computing the utility

of the packets in a node's buffer based on a global routing metric (such as the average delay or the number of missed deadlines) and by replicating only packets with the highest utility. Single replication is advocated in the Prediction Assisted Single-copy Routing (PASR) protocol [38], which outperforms multi-copy routing in some resource-constrained underwater network scenarios.

Node trajectories, when deterministically known, can be leveraged to make optimal routing decisions. For example, PROPHET [39] limits the replication of packets by forwarding replicas only to those neighbors that will meet the packet's destination in a short time. Similarly, [40], [41] suggest to compute optimal routes on event-driven graphs, that convey the sequence of the contacts over time and their duration.

In [42], two nodes in contact exchange packets if they move orthogonally, which has a better chance to distribute the packet to relays different than those encountered by either node in the past. The concept above is extended in [43] where nodes leverage the history of previous encounters to predict the area where each packet's destination node is likely to be located, and replicate the packet to nodes directed there. In [44], the authors characterize vehicle mobility patterns via high-order Markov chains, and use them to predict inter-contact times. Upon contact, packets are forwarded only if the encountered node provides an opportunity to reduce the delivery delay. The storage-friendly region-based (RENA) protocol [45], relieve the need to store the location history in order to predict future encounters by selecting relays based only on the expected movement time from the current region to the destination region. Similarly, [46] assumes that a node moves around a limited set of fixed landmarks, and that mobility behaviors can be at least roughly predicted based on mobility history. The resulting predict-and-relay (PER) protocol describes mobility as transitions between landmarks and exploits the resulting model to predict future contact times and choose relays.

Several approaches considered in the above works are suited to terrestrial DTNs, but cannot be directly ported under water. As explained in the introduction, underwater DTNs have distinctive properties and constraints that require specific solutions. Instead, many routing protocols for underwater DTNs [38], [47], [48] are designed in a way that is fully or partly oblivious to the nature of underwater environments. In [47], the authors assign an application-dependent priority to packets, which depends on geographic considerations. Therefore, all nodes should be aware of the sink position, whereas our proposed protocol is independent of such constraints. The same issue applies to [48], that needs up-to-date location information to make relaying decisions. DTNs for underwater networks have been considered in the past [49], [50], but mainly by showcasing the system-level interconnections between the IP protocol and store-and-forward mechanisms, without the design of a specific DTN protocol. The policies presented in [12] also consider probabilistic forwarding, but do not consider transmission errors

and operate on a single packet at a time: as such they are not well suited to underwater scenarios. Similar arguments apply to the more recent work in [51], which assumes infinite forwarding bandwidth and storage, and requires a proactive distributed mechanism to make optimal decisions. Delay-tolerant communications are considered in [52], where the focus is on designing an optimal trajectory for a mobile data-collecting node. In contrast, we consider networks of fully mobile data-exchanging entities.

In the present work, we make further steps towards a practical routing protocol for underwater DTNs. Our proposed scheme, UDTN-Prob, estimates local movement patterns in order to allocate the available meeting time to the transmissions of the nodes in contact. Then, only those packets that show a sufficiently high chance of being received by the destination in due time are actually transmitted. This selection is performed based on statistical information about the inter-contact times with the final destination that can be collected on the fly. Negligible space is required to store such information at the nodes. A selective-repeat ARQ (SR-ARQ) scheme that takes explicit advantage of long propagation delays is seamlessly integrated with the protocol operations in order to protect transmissions against losses induced by harsh channels.

In the spirit of [31] and unlike [38], we allow nodes to transmit multiple replicas of the same packet, but choose which packet to replicate depending on the statistics of the future contacts between the relay and the sink. Unlike PROPHET [39], UDTN-Prob does not assume that the trajectories of the nodes are periodical and fully known a priori. Rather, we consider a random mobility model that reproduces smooth 3D trajectories that would be typical of mobile underwater vehicles as they carry out their mission and react to events. In turn, only the statistics of the time to the next contact with a given node are known, and UDTN-Prob leverages on these statistics to decide which packets should be transmitted. For the same reason, we cannot rely on the trajectory orthogonality considerations of [42], but rather decide to replicate packets based on the statistics of future encounters. As a result, the information to be stored by the nodes is very limited, making our approach more feasible than HVR [43] in this respect.

III. UNDERWATER DELAY-TOLERANT NETWORK ROUTING PROTOCOL WITH PROBABILISTIC SPRAYING (UDTN-PROB)

In this section, we introduce the various design aspects of our Underwater Delay-Tolerant Network routing protocol with Probabilistic Spraying (UDTN-Prob) protocol. In the taxonomy of DTN protocols, UDTN-Prob is a store-and-forward replication-based protocol. Unlike plain replication-based protocols, UDTN-Prob leverages on the knowledge of the statistics of the inter-contact time between network nodes, where contacts start and end due to node mobility. We describe UDTN-Prob through the following three subsections: the details on the statistics employed to predict inter-contact

times are described in Section III-A; the neighbor discovery and contact setup messages exchanged by UDTN-Prob are detailed in Section III-B; the data exchange phase is described in Section III-C, and some implementation details in Section III-D conclude the description. As a clearer reference for the employed notation, we include the key variable names and meanings in Table 1.

A. PREDICTION OF ONE-HOP AND TWO-HOP SINK θ -MEETING TIME

The overarching objective of UDTN is to allow the largest possible fraction of the packets generated in the network to reach the sink (which can be either static or mobile), within their assigned delivery deadline. To do so, UDTN relies on the measured statistics of the inter-contact time between the nodes. Let t_{ij} be a random variable representing the time elapsed since the last contact of nodes i and j , and call $F_{ij}(t)$ the cumulative distribution function (CDF) of t_{ij} . The basic design of UDTN assumes that each node i knows an estimate of $F_{iS}(t)$, where S denotes the destination node (or sink). Assuming that meetings take place erratically (which is typically the case in mobile underwater networks), the knowledge of $F_{iS}(t)$ provides node i with a measure of the probability of meeting the sink within a certain amount of time t . Such information can be collected in real time and progressively refined as a given mission is carried out, or otherwise pre-calculated by applying a communication link model to expected movement trajectories, in order to understand which nodes are within the coverage range of one another at any given time. We remark that UDTN-Prob works correctly regardless of the specific function $F_{iS}(t)$. What matters to the protocol is the sink θ -meeting time, defined as the quantile $q_\theta = F_{iS}^{-1}(\theta)$, $0 \leq \theta \leq 1$, where $F_{iS}^{-1}(\cdot)$ is the inverse of the CDF $F_{iS}(t)$. Since by definition we have $F_{iS}(q_\theta) = \mathbb{P}[t \leq q_\theta] = \theta$, the quantile q_θ conveys the maximum amount of time required for node i to meet the sink with probability θ . Note that the θ -meeting time is a non-decreasing function of θ . This derivation of θ -meeting times through the CDF of the inter-meeting time between a node and the sink is named one-hop sink θ -meeting time prediction.

Let us now turn to a more general scenario by considering that, in the presence of erratic contacts, it is not granted that shortest paths are also the quickest, and actually in general they are not. Therefore, being able to determine whether it is more efficient to send a packet to the sink through more than one intermediate hop may result in better packet delivery ratio and shorter end-to-end delays. The drawback in this case is that the amount of information required to perform this prediction is much larger than in the previous case. In particular, a generic node i needs to know not only the CDF of the time until the next contact between itself and the sink, but also the CDF of the time to the next contact with every other node $j \neq i$, as well as (at least an estimate of) the CDF of the remaining time, $F_{jS}(t)$.

For the moment, assume that this knowledge is available,

Table 1: Summary of the notation employed in this paper

Name	Meaning	Name	Meaning
t_{ij}	Inter-contact time of nodes i and j	P_{tgt}	Target probability of bit error
$F_{ij}(t)$	CDF of t_{ij}	γ_{tgt}	Target signal-to-noise-ratio
θ	Probability to meet the sink S	L	Data packet length
q_θ	Quantile of $F_{iS}(t)$, $q_\theta = F_{iS}^{-1}(\theta)$	f, ν	Transmission frequency and bandwidth
$\mathbf{x}_i, \mathbf{x}_j$	Position of nodes i and j	$a(f)$	Thorp absorption coefficient at frequency f
$\mathbf{v}_i, \mathbf{v}_j$	Velocities of nodes i and j	T_s, η	Share and fraction of the contact duration for the Response sender
$\zeta_{\mathbf{x}}, \zeta_{\mathbf{v}}$	Estimation errors for the relative position and velocity	$p_\ell^{\text{TX}}(m)$	Copies of packet m entrusted to the receiver after transmission ℓ
$\mathbf{x}_{ij}^{(r)}$	Relative position of i and j , $\mathbf{x}_{ij}^{(r)} = \mathbf{x}_i - \mathbf{x}_j + \zeta_{\mathbf{x}}$	τ_{ij}	Propagation delay between nodes i and j
$\mathbf{v}_{ij}^{(r)}$	Relative velocity of i and j , $\mathbf{v}_{ij}^{(r)} = \mathbf{v}_i - \mathbf{v}_j + \zeta_{\mathbf{v}}$	τ'_{ij}	Version of τ_{ij} corrected against reduced guard times
$d_{ij}(t)$	Distance between nodes i and j	v_{\max}	Maximum speed of a given node
T_{ij}^c	Estimate of the contact time between i and j	T_D, T_A	Data and ACK packet transmission time
d_{TX}	Transmission range	Δ	Guard time for the TDD SR-ARQ exchange
k	Tuner for the spacing of data and ACK packets	M'_{ij}, W'_{ij}	SR-ARQ window length, time between subsequent data packets

that the encounter processes are independent, and that each process is uncorrelated over time. We consider the case in which the packet reaches the sink in two hops, i.e., from i to j and then from j to S . Call $t_{ijS} = t_{ij} + t_{jS}$ the time when node j meets the sink after having met node i . The CDF of t_{ijS} , i.e., the probability that this sequence of meetings takes place in a time less than or equal to t , can be computed as

$$F_{ijS}(t) = \int_0^t f_{ij}(y) F_{jS}(t-y) dy, \quad (1)$$

where $f_{ij}(u) = dF_{ij}(u)/du$. Computing the quantile $F_{ijS}^{-1}(\theta)$, $0 \leq \theta \leq 1$ represents a two-hop sink θ -meeting time prediction.

We remark that a node i may not have access to the functions $F_{jS}(t-y)$ in (1) for $j \neq i$, as such functions relate to the inter-contact times between other nodes and the sink. However, these functions can be exchanged and updated over time upon contact between two nodes. In practice, a node i will store both $F_{iS}^{-1}(\theta)$ and $F_{ijS}^{-1}(\theta) \forall j$ in tabulated form for a limited number of values of θ . This way, upon contact with another node, i will be able to measure the shortest among the one-hop and the two-hop sink meeting times, $\min\{F_{iS}^{-1}(\theta), \min_j\{F_{ijS}^{-1}(\theta)\}\}$, and communicate it to the node as part of the messaging described in Section III-B. In turn, the node in contact with i will be able to discriminate which packets in its own buffer have a good chance to be actually delivered within their deadline, if sent to node i .

We conclude this subsection by noting that the two-hop estimation technique above can be actually extended to any number of hops. The obvious drawbacks are that the information a node needs to know increases (basically all nodes should know the inter-contact meeting statistics between themselves and every other node, as well as between any other two nodes); that (1) must be extended to multiple nested integrals; and that keeping the statistics of the inter-meeting times up to date may require a substantial amount of information exchange upon contact, and therefore a sig-

nificant waste of bandwidth.¹ However, we will show that extending beyond one-hop θ -meeting time predictions is at best marginally better and, in most cases, does not justify the additional complexity.

B. UDTN-PROB MESSAGING FOR NEIGHBOR DISCOVERY AND CONTACT SETUP

The messaging scheme of UDTN-Prob is illustrated in Fig. 1, and comprises three phases:

- 1) Contact discovery through the transmission of Beacon packets;
- 2) Analysis of contacts via the collection of Info packets from any neighboring nodes;
- 3) Selection of a node and contact establishment via the transmission of Response packets;
- 4) Error-controlled data transmission in turns, via a TDD-based SR-ARQ scheme.

We will now illustrate the details of steps 1 to 3 in this section. Step 4 will be detailed in Section III-C.

In order to discover contact opportunities, every node periodically broadcasts Beacon packets. In fact, in a generic DTN, the nodes may experience prolonged periods of isolation, and therefore cannot communicate with any other node until a contact occurs. After each Beacon transmission, the node waits for a given amount of time, predefined and known to all nodes. If no answer is received within the waiting period, i presumes that no node is located within its transmission range and retransmits a new Beacon after a random backoff.

If another node, say j , receives i 's Beacon message, it replies with a corresponding Info message. This message is used to convey the following information: the node's current position and velocity information (which allows the Beacon sender to estimate the relative velocity of the nodes, hence, the contact duration); the earliest deadline among all packets in j 's buffer; a subsampled version of the distribution

¹In the special case where meeting times are deterministically known, the approaches in [40], [41] provide a framework to optimize the DTN. In the present work, we focus on DTNs where only the statistics of the inter-contact time and of the contact duration are known.

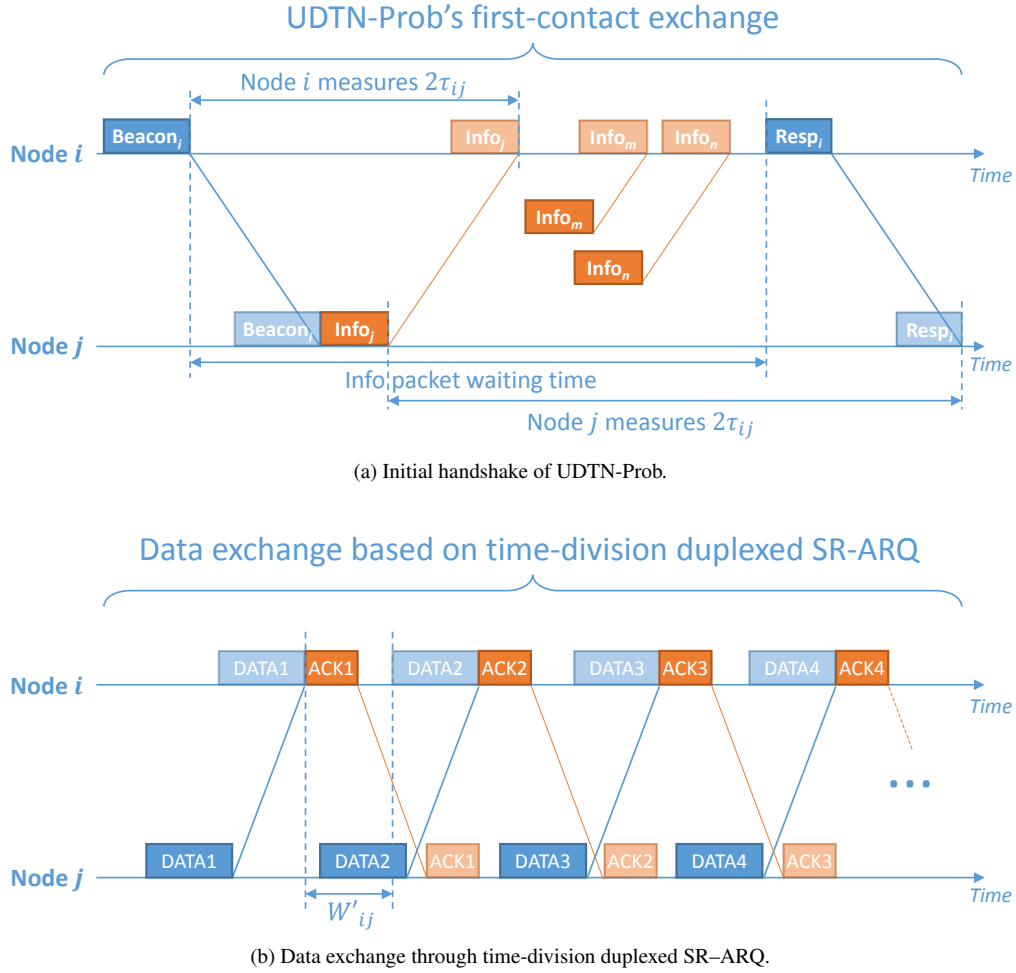


Figure 1: Example of control and data packet exchange in UDTN-Prob (a) and of the ensuing error-controlled data transmission. The communicating nodes i and j measure the round-trip time $2\tau_{ij}$ during UDTN-Prob's contact discovery sequence, and use this estimate to derive the timing of the time-division duplexed SR-ARQ exchange (only the transmission from i to j is shown).

function of the inter-contact time between node j and the destination (more implementation details are provided in Section III-D).² After sending the **Info** message, node j waits for the respective **Response** packet for up to a given maximum waiting time. If no **Response** message is received within this time, j goes back to the idle state.

When i receives the **Info** packet from j , it estimates the contact duration as follows. Call \mathbf{x}_i and \mathbf{x}_j the positions of nodes i and j , respectively, and let \mathbf{v}_i and \mathbf{v}_j be their speed vectors, whose magnitude is measured in m/s. Therefore, the relative position of i and j is $\mathbf{x}_{ij}^{(r)} = \mathbf{x}_i - \mathbf{x}_j + \zeta_x$ and the relative velocity of those nodes is $\mathbf{v}_{ij}^{(r)} = \mathbf{v}_i - \mathbf{v}_j + \zeta_v$, where ζ_x and ζ_v are random terms that represent the relative position

²In the rare case both nodes i and j should send their own **Beacon** almost at the same time, and the typically large propagation delays incurred under water should allow both to receive each other's **Beacon** correctly, the corresponding tie is broken by having the node with the lowest ID wait for **Info** messages, while the other will transmit its own **Info**. In any event, we remark that both nodes will be given a transmit opportunity once the contact is established.

estimation and velocity estimation errors, respectively. Given that the trajectories of the nodes remain the same or change negligibly within the short time required by UDTN-Prob's handshake, we assume that the average values of $\mathbf{x}_{ij}^{(r)}$ and $\mathbf{v}_{ij}^{(r)}$ are the same for both nodes, and that the random relative errors absorb possible discrepancies due to, e.g., position or velocity estimation errors. The instantaneous distance $d_{ij}(t)$ at time t is found as

$$d_{ij}(t) = \|\mathbf{x}_{ij}^{(r)} + \mathbf{v}_{ij}^{(r)}t\| = \sqrt{\|\mathbf{v}_{ij}^{(r)}\|^2 t^2 + 2(\mathbf{x}_{ij}^{(r)} \cdot \mathbf{v}_{ij}^{(r)})t + \|\mathbf{x}_{ij}^{(r)}\|^2}, \quad (2)$$

where \cdot denotes the inner product, and $\|\mathbf{a}\|$ is the 2-norm of vector \mathbf{a} . The approximate contact duration T_{ij}^c can be computed as the time required for the nodes to exit the communication range of each other. The contact time follows by equating $d_{ij}(t)$ to a nominal transmission range of the underwater acoustic communication system, d_{TX} . Here, we compute d_{TX} by assuming that the BPSK modulation scheme

is employed to transmit packets of length L and the transmission power level is set such that the receiver achieves a target packet error rate (PER), P_{tgt} . By assuming independent bit errors across a packet, we have

$$P_{\text{tgt}} = 1 - \left(1 - 0.5 \operatorname{erfc} \sqrt{\gamma_{\text{tgt}}/\psi}\right)^L, \quad (3)$$

where γ_{tgt} is the corresponding target Signal-to-Noise Ratio (SNR) at the receiver, and ψ is an SNR margin [53]. Inverting (3) yields

$$\gamma_{\text{tgt}} = \psi \left(\operatorname{erfc}^{-1}(2 - 2(1 - P_{\text{tgt}})^{1/L}) \right)^2. \quad (4)$$

Given the distance of the nodes, the carrier frequency of the acoustic signals f , the transmit source level P_{TX} and the noise power in the communications bandwidth ν , we can derive the SNR γ_{ij} of a given transmission between nodes i and j via the well-known link budget equations for an underwater acoustic transmission [54]:

$$\gamma_{ij} = \frac{P_{\text{TX}} d_{ij}(t)^{-b} a(f)^{-d_{ij}(t)}}{\nu}, \quad (5)$$

where b is the spreading factor, $a(f)$ is the linear-scale Thorp absorption coefficient computed for f in kHz, and ν is derived from the power spectral density of the noise, computed using standard equations such as those in [54]. The spreading factor b describes the geometry of the propagation, but can also be tuned to fit data obtained from experiments or from the simulation of channel realizations [55]. In the following, we will assume that $b = 1.75$ as in [55].

We define the transmission range d_{TX} as the distance where a receiver j receives a signal from transmitter i such that the SNR of this signal is $\gamma_{ij} = \gamma_{\text{tgt}}$. Therefore,

$$d_{\text{TX}} = \frac{b}{\log a(f)} \mathcal{W} \left(\frac{\log a(f)}{b} \left(\frac{\gamma_{\text{tgt}} N}{P_{\text{TX}}} \right)^{-1/b} \right), \quad (6)$$

where $\mathcal{W}(x)$ is the principal branch of the Lambert function, $x \geq -e^{-1}$, defined as the unique solution of the equation $ye^y = x$, $y \geq -1$ [56]. By equating the instantaneous distance in (2) to the transmission range in (6), we have

$$\|\mathbf{v}_{ij}^{(r)}\|^2 (T_{ij}^c)^2 + 2(\mathbf{x}_{ij}^{(r)} \cdot \mathbf{v}_{ij}^{(r)}) T_{ij}^c + \|\mathbf{x}_{ij}^{(r)}\|^2 = d_{\text{TX}}^2, \quad (7)$$

which yields the approximate contact duration as

$$T_{ij}^c = \frac{-(\mathbf{x}_{ij}^{(r)} \cdot \mathbf{v}_{ij}^{(r)})}{\|\mathbf{v}_{ij}^{(r)}\|^2} + \frac{\sqrt{(\mathbf{x}_{ij}^{(r)} \cdot \mathbf{v}_{ij}^{(r)})^2 - \|\mathbf{v}_{ij}^{(r)}\|^2 (\|\mathbf{x}_{ij}^{(r)}\|^2 - d_{\text{TX}}^2)}}{\|\mathbf{v}_{ij}^{(r)}\|^2}. \quad (8)$$

We recall that node i may receive several Info packets, each from a different neighbor. The procedure for computing T_{ij}^c is therefore repeated for every Info packet (hence for every node). The contacts that are estimated to be shorter than a user-defined minimum threshold T_{min}^c are discarded right

away.³ Among the remaining nodes, the neighbor with the highest estimated contact time, say node j , is chosen and sent a Response message.⁴ If no nodes offer an estimated contact time greater than or equal to T_{min}^c , node i goes back to the idle state and the Beacon-Info-Response signaling restarts from the beginning.

To continue with the description of the protocol, assume node i has chosen to make contact with node j . In the header of its Response packet to j , node i includes its own share of the contact duration, simply computed as $T_s = \eta T_{ij}^c$, implying that j will have an available time equal to $(1 - \eta) T_{ij}^c$ to transmit its own packets to node i . The factor η makes it possible to implement priority policies. If there is no specific priority to be accounted for in the communication between i and j , we set $\eta = 0.5$. If j is a sink, then $\eta = 1$ as the sink transmits nothing, and node i can employ the whole contact time to transmit its own packets. Finally, the Response packet from i to j incorporates the statistics of the inter-contact time between i and the destination, which enables node j to compute the sink θ -meeting times, and thereby decide on the priority of the packets that should be relayed to i . We remark that the Response is sent in unicast from i to j : therefore, if the Response packet is lost due to communication issues, node j times out and returns to the idle state.

When j receives the Response packet, it has all the necessary information to obtain the same estimate of T_{ij}^c computed by i , as well as the share of this time that it can use to transmit packets. Note that the reception of the Response packet marks the first instant when both nodes in contact become aware of the estimated contact duration. Therefore, to make the packet transmission phase mechanism robust against short contacts, we designed UDTN such that the receiver of the Response packet (node j in this example) is the first node to send data. On the contrary, the Response sender (node i) could start data transmission right after sending the Response message. However, in this case all contacts whose duration is shorter than a transmission window would lead to a loss of data. Since the node encounter process is stochastic, packets with a shorter deadline are taken into account if the length of the contact time share assigned to j so allows. Once the packets are selected, they are transmitted using a time-division duplexed error-control scheme, as discussed

³Here, we set $T_{\text{min}}^c = 2T_D + 2T_A + 2\Delta$, where T_D and T_A are the transmission times of a data packet and of an ACK packet, and Δ is a guard time that accounts for a short propagation delay among the nodes. This is the minimum contact time that would allow both i and j to send a packet to each other, and thus prevents that i and j start sending packets to one another when there is not enough time to do so.

⁴The only exception to this rule is when the sink is within range. In this case, the sink always gets priority when establishing contact with a node, however short.

in Section III-C.⁵ Whenever a data packet is transmitted, a binary spray technique [31] is applied: namely, a node is allowed to transmit each packet m up to a maximum number of times. When packet m is generated, this number is set to a predefined value $p_1^{\text{TX}}(m) = p_{\max}$, where the subscript 1 indicates that this limit applies to the first transmission. Upon each packet transmission confirmed by a corresponding acknowledgment (ACK) reception, this number is divided between the transmitter and the receiver, namely $p_2^{\text{TX}}(m) = \lceil p_1^{\text{TX}}(m)/2 \rceil$ and $p_2^{\text{RX}}(m) = p_1^{\text{TX}}(m) - p_2^{\text{TX}}(m)$. This means that when $p_\ell^{\text{TX}}(m)$ reaches a value of 1 for some ℓ , the transmitting node will hold the last copy of packet m for itself, and relay it only to the sink, or drop it once its deadline expires. After the packet transmissions performed by j are over (i.e., its share of the contact time has expired), node i starts transmitting its own packets according to the same procedure.

As a final detail, if a node has no packet to transmit, it signals this fact to the receiver through a short Proxy packet, so that the receiver can start its own transmissions in advance. In addition, the last packet to be transmitted by a node is always flagged. This makes it possible for the receiver to start sending packets earlier. Note that both Proxy and flagged data packets are protected from transmission errors by the SR-ARQ mechanism described in the next subsection, like all data packets exchanged during the contact. When the transmission and reception phases are over for both nodes, they move back to the idle state and resume the periodic transmission of Beacon packets after a random time.

C. ERROR-CONTROLLED DATA PACKET EXCHANGE

To make the best use of the erratic contacts that take place in an underwater DTN, it is important to protect data packet exchanges from transmission errors. Moreover, the error control mechanism should not imply long silence times, for example as prescribed by the stop-and-wait ARQ scheme, where the transmitter cannot send any new packets for one full round-trip time (RTT), before it receives an ACK packet from the receiver. It is specifically important to take these aspects into account in underwater DTNs, where the network area can be very large, the network could be composed of just a few nodes, and contacts may be erratic and short due to the movement speed of the mobile underwater vehicles. In order to make the best use of these relatively rare contacts, we employ a SR-ARQ scheme modified to leverage the long propagation delays experienced in underwater acoustic communications by interlacing packet transmissions and ACK receptions in a time-division duplexed (TDD) fashion, akin to [32], [33]. This technique enables a more efficient

utilization of the underwater channel, as data transmissions and ACK receptions occur more closely, and therefore the amount of time that the nodes remain in contact (estimated via Eq. (8) as explained in Section III) is exploited more effectively by transmitting a larger number of packets. By way of contrast, stop-and-wait policies would not achieve the same effectiveness, as they would require the transmitter to wait for the reception of an ACK packet before sending a new data packet. The main idea behind our TDD SR-ARQ scheme is to enable the transmission of multiple data packets without waiting for ACKs to be received by the transmitter; at the same time, data packet transmissions are spaced such that ACK packets would be received within subsequent data packet transmissions. This is made possible by estimating the RTT from the UDTN handshake to tune the spacing among the data and ACK packets, and by correcting the RTT estimation for the worst-case mobility scenario. The latter occurs when the transmitter and the receiver approach each other, which reduces the guard intervals preserving the interlacing of data and ACK packets. We note that the above mechanism is made specifically possible by the significant propagation delays experienced over underwater acoustic channels, as these delays are most typically larger than the duration of a data packet transmission. In the following, we include the main characteristics of the error control scheme in order to keep the presentation self-contained.

In line with Section III-B, call i a transmitter and j its receiver. The main ingredient to implement a TDD SR-ARQ scheme is to estimate the Round-Trip Time (RTT), $2\tau_{ij}$ between nodes i and j , where τ_{ij} is the one-way propagation delay between nodes i and j , defined as the time that elapses from the end of a given transmission to the end of the corresponding reception at the receiver.

Our SR-ARQ scheme bases the computation of its parameters on the RTT estimate $2\tau_{ij}$ that can be derived by both nodes i and j based on the timing of the packets of UDTN's preliminary handshake. Specifically, with reference to Fig. 1a, we note that this time can be independently measured by both nodes upon the establishment of a contact, where node i can already take the estimate at the arrival of the Info packet from node j , whereas j needs to wait until the arrival of the Response packet from i .

We remark that the estimation of τ_{ij} through UDTN's preliminary handshake yields an instantaneous value, which may change over time due to the movement of the nodes. In fact, the DTN nodes' mobility may yield unforeseen trajectory changes, that would both alter the effective interlacing of data and ACK packets, and change the expected contact duration computed in Section III-B. To make the protocol robust to mobility, we assume that the nodes move towards each other, i.e., that the RTT decreases progressively over time. This is a conservative assumption, which prevents that guard times reduce excessively and lead to collisions between data packets and ACKs when two nodes move towards each other. Specifically, assume that node i measured the propagation delay towards node j at time t_1 . Upon later transmissions

⁵We note that the use of an error control scheme can lead to retransmissions, which in turn may potentially make a node exceed its share of the contact duration. In this case, the node will drop the transmission of the packets that would exceed its allotted time and keep them in the buffer for later transmission. In any event, packets with a longer lifetime than the computed sink θ -meeting time are transmitted first (and retransmitted if required), therefore the packets dropped to avoid exceeding the transmission deadline are typically low-priority.

between i and j , say at time t_2 , node i can adjust the propagation delay estimate. A worst-case strategy for node i to avoid the unwanted reduction of guard times is to define a modified propagation delay τ'_{ij} as

$$\tau'_{ij} = \tau_{ij} - \frac{N_M(t_2 - t_1)v_{\max}}{c}, \quad (9)$$

where v_{\max} is the maximum velocity of a mobile node, c is the speed of sound in the water (approximated to a fixed value of 1500 m/s for this computation), and $N_M = 0$ if the sender and receiver are both static, $N_M = 1$ if one of the two moves (e.g., when a mobile node gets in contact with a fixed sink), or $N_M = 2$ if both nodes i and j move. Eq. (9) corresponds to taking the instantaneous estimate computed from the UDTN-Prob handshake τ_{ij} (see also Fig. 1) and by subtracting the distance that the nodes i and j would have covered between time epochs t_1 and t_2 when traveling towards each other, assuming that N_M of them actually move [4]. The modified propagation delay estimate is then used to compute the transmission window, defined as the maximum number of packets that can be sent before any ACK is received:

$$M'_{ij} = \min \left\{ M_F, \max \left(1, \left\lfloor \frac{k \tau'_{ij}}{T_D + T_A + \Delta} \right\rfloor \right) \right\}, \quad (10)$$

where T_D and T_A are the transmission time of a data packet and of an ACK packet, respectively, Δ is a short guard time allotted to every data/ACK exchange and $M_F \geq 1$ is a user-defined upper bound to the window length, which is also employed as a fallback in case UDTN fails to estimate the contact duration for any reason. The value of M_F can be set in order to approach the performance of optimal TDD: a value between 5 and 10 is typically a good choice [57]. The factor $0 \leq k \leq 2$ is a tunable parameter, which specifies the fraction of τ'_{ij} to be considered in the computation of M'_{ij} . Namely, if $k = 0$, it can be observed from (10) that $M'_{ij} = 1$, hence SR-ARQ falls back to a simple Stop-and-Wait (S&W) ARQ scheme; conversely, for $k = 2$ the whole RTT will be considered when computing the window size, and the interleaving between data packets and ACKs will be tightest.

When $M'_{ij} \geq 2$, the time interval between subsequent data transmissions is employed to receive ACKs. This is the case, e.g., in Fig. 1b. Call W'_{ij} the length of the time interval between subsequent data packets, where the prime denotes again that the modified RTT estimate in (9) is employed to compute all subsequent quantities. In order to maximize the separation with both the preceding and the subsequent data packets, ideally the ACK reception should be centered at $W'_{ij}/2$. With this constraint, W'_{ij} is derived as:

$$W'_{ij} = \frac{T_A + 4\tau'_{ij} - 2(M'_{ij} - 1)T_D}{2M'_{ij} - 1}. \quad (11)$$

We remark that the setting in (11) interlaces data packet transmissions and ACK receptions with maximum guard times. This makes the timing still valid even if the actual ACK

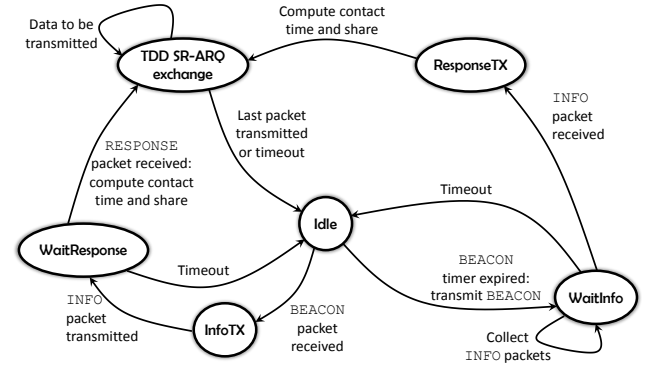


Figure 2: State transition diagram for UDTN.

reception is slightly anticipated or delayed, e.g., due to small changes in the propagation delay induced by changes in environmental conditions. Recall that the RTT estimate will be updated by node i upon the reception of every ACK corresponding to a previously transmitted data packet. This makes it possible to keep the value of τ'_{ij} , M'_{ij} and W'_{ij} up to date in face of RTT changes due to node mobility. Therefore, the interleaving of data packets and ACKs is frequently adapted to RTT changes. Once the window is known, UDTN-Prob will transmit according to a standard SR-ARQ technique, where no more than M'_{ij} packets will be transmitted before an ACK is received. We remark that as is common for DTN protocols [58], ACKs are not forwarded at a global route level, but only per link.

As per UDTN-Prob's rules, nodes i and j need to abide by the shares of T_{ij}^c assigned to them. Other details have been implemented to enforce this. For example, while the preliminary contact establishment of UDTN-Prob already provides a RTT estimate, the error-controlled transmission of a data packet does not start unless there is enough time to receive the corresponding ACK given the current estimate of the RTT. If many retransmissions should occur, or if the transmitter and the receiver accelerate out of range and thus make a contact shorter than initially expected, the above condition means that some packets in the queue will not be sent. Rather, they would be left for a future contact with another relay, if their deadline does not expire before. Additionally, the packets transmitted to the node currently in contact are flagged so that they are not transmitted to the same node again, should the same two nodes meet again before the packets' deadline.

D. SUMMARY AND IMPLEMENTATION DETAILS

To summarize the design of UDTN, we depict the state transition diagram in Fig. 2, and provide further details of the nodes' behavior in the pseudocodes of Algorithms 1 and 2, which refer to the case of a node initiating a connection and to that of a node accepting a connection, respectively. For better clarity, the TDD SR-ARQ data packet exchange discussed in Section III-C is summarized as a single state.

The computation of the quantiles described in Sec-

Algorithm 1: Behavior of a node initiating a connection

Input: Current node n , current state $cState$

```

1 switch  $cState$  do
2   case Idle do
3     if  $n$  is not a sink and has packets to transmit then
4       Set BEACON transmission timer
5       if Timer expires then
6         Send BEACON packet
7          $cState \leftarrow \text{WaitInfo}$ 
8   case WaitInfo do
9     Set INFO packet collection timer
10    while Timer not expired do
11      Collect INFO packets
12      foreach INFO packet received do
13        Update inter-contact time distribution for
14        INFO sender
15     $cState \leftarrow \text{ResponseTX}$ 
16  case ResponseTX do
17    Calculate contact duration for all INFO
18    transmitters
19    Choose node  $m$  yielding the highest contact
20    duration
21    Compute contact time share for  $n$  and  $m$ 
22    Send RESPONSE packet to  $m$ , including
23    subsampled distribution of the inter-contact
24    time with the sink
25  case TDD SR-ARQ exchange do
26    Compute transmission window  $M$  and inter-TX
27    time  $W$ 
28    Perform TDD SR-ARQ DATA/ACK exchange as
29    DATA RX; if Node  $m$  sent last packet or contact
30    share of  $n$  starts then
31      Perform TDD SR-ARQ DATA/ACK exchange
32      as DATA TX;
33    if Last packet or end of contact share then
34       $cState \leftarrow \text{Idle}$ 

```

Algorithm 2: Behavior of a node accepting a connection

Input: Current node n , current state $cState$

```

1 switch  $cState$  do
2   case Idle do
3     if BEACON packet received from node  $m$  then
4       Update inter-contact time distribution for
5       node  $m$ 
6        $cState \leftarrow \text{InfoTX}$ 
7   case InfoTX do
8     Collect position estimate  $x$  and velocity estimate
9      $v$  of  $n$ 
10    Retrieve earliest deadline for all packets in queue
11    Send INFO packet to node  $m$  with all above
12    information
13     $cState \leftarrow \text{WaitResponse}$ 
14  case WaitResponse do
15    if RESPONSE received before timeout period then
16      Read contact share and compute contact
17      duration
18       $cState \leftarrow \text{TDD SR-ARQ exchange}$ 
19    else
20       $cState \leftarrow \text{Idle}$ 
21  case TDD SR-ARQ exchange do
22    Compute transmission window  $M$  and inter-TX
23    time  $W$ 
24    Perform TDD SR-ARQ DATA/ACK exchange as
25    DATA TX
26    if Last packet or contact share of  $m$  starts then
27      Perform TDD SR-ARQ DATA/ACK exchange
28      as DATA RX;
29    if Last packet or end of contact share then
30       $cState \leftarrow \text{Idle}$ 

```

tion III-A requires a node j to store the distribution of the inter-contact time between itself and the sink, $F_{kS}(t)$. Additionally, for two-hop θ -meeting time prediction, node j needs to know the distribution of the inter-contact time between itself and any other node, $F_{jk}(t)$, as well as an estimate of the distribution $F_{kS}(t) \forall k \neq j$. We remark that node j 's own distributions can be estimated and stored down to an arbitrary level of precision using standard binning procedures. This also enables the real-time update of the distribution when new data becomes available, i.e., when a new meeting takes place between j and another node or the sink. However, the distribution of the inter-contact time between other nodes k

and the sink needs to be communicated using Info packets. This requires some level of compression in order to avoid excessive overhead.

To communicate such distributions, we employ a limited set of values corresponding to the quantiles $F_{jS}^{-1}(\theta_\ell)$, where $\theta_\ell \in \Theta$, and the latter set contains predetermined meeting probability values. In our implementation, we communicate $|\Theta| = 20$ quantiles of the distribution, for $t_\ell = F_{jS}^{-1}(\theta_\ell)$ and $\Theta = \{\theta_\ell = 0.05\ell, 1 \leq \ell \leq 20\}$. Given a meeting probability threshold θ , the prediction of one-hop sink θ -meeting is then performed by finding

$$\hat{t}_1(j) = \min_{\theta_{jS} \geq \theta} F_{jS}^{-1}(\theta_{jS}). \quad (12)$$

Node j predicts its two-hop sink θ -meeting time by comput-

ing:

$$\begin{aligned} \hat{t}_2(j) = \min_{\theta_{jk}, \theta_{kS}} F_{jk}^{-1}(\theta_{jk}) + F_{kS}^{-1}(\theta_{kS}) \\ \text{s.t. } \theta_{jk}\theta_{kS} \geq \theta. \end{aligned} \quad (13)$$

IV. SIMULATION RESULTS

In this section we evaluate the performance of UDTN-Prob as a function of different network and protocol configuration parameters using the DESERT Underwater simulation framework [59], based on the well known network simulator ns2 [60] and its MIRACLE extensions [61]. We start by explaining the simulation scenario and default parameter setting in Section IV-A; we proceed with several sets of simulation results in Section IV-B and conclude in Section IV-C by comparing the performance of UDTN when using both one-hop and two-hop sink θ -meeting time predictions to assign priorities to the packets to be transmitted.

A. SIMULATION SCENARIO AND METRICS

We consider a network with one fixed sink and either 5 or 10 mobile AUVs. The network is deployed over an area of 9000 m \times 5000 m, whose north-west corner is located at coordinates (39.97°N, 11.82°E). The depth of the area is in line with the geographical bathymetry in this region, as retrieved from the GEBCO database [62]. The nodes communicate via a Binary Phase Shift Keying (BPSK) modulation technique at a bit rate of 4800 bps using a central frequency of 25 kHz and a bandwidth of 9 kHz. By setting a source level of $P_{TX} = 150$ dB re μ Pa relative to a distance of 1 m from the source, this leads to an estimated nominal transmission range $d_{TX} = 2000$ m in (6). The size of the Beacon packet is 10 Bytes, the Info, Response, and data packets are 125 Bytes, whereas Proxy packets (which notify that a node has no packets to transmit) and ACK packets have a size of 10 Bytes. At the beginning of a simulation run, all AUVs are deployed at random within the area. After that, they start moving freely. Their trajectories are simulated as random realizations of a three-dimensional Gauss-Markov process with fixed self-correlation parameter $\alpha = 0.8$ [63]. As illustrated in Fig. 3, this leads to random yet smooth trajectories, that reproduce sufficiently well the actual trajectories that autonomous underwater vehicles may follow during patrol, reconnaissance or survey missions. A sink is placed at one side of the network.

The nodes are provided a preliminary estimate of the CDF of the inter-contact time between themselves and all other nodes. This estimate is obtained by simulating the movement of the nodes off-line through a Gauss-Markov model with the same self-correlation for a total period of six simulated months.

Data packets are generated by each AUV according to a Poisson process of rate λ , ranging from 0.12 to 3 packets per minute per node. During a simulation run, the packet generation process is maintained active for 24 simulated hours and the simulation is concluded 4 hours later. Simulation results

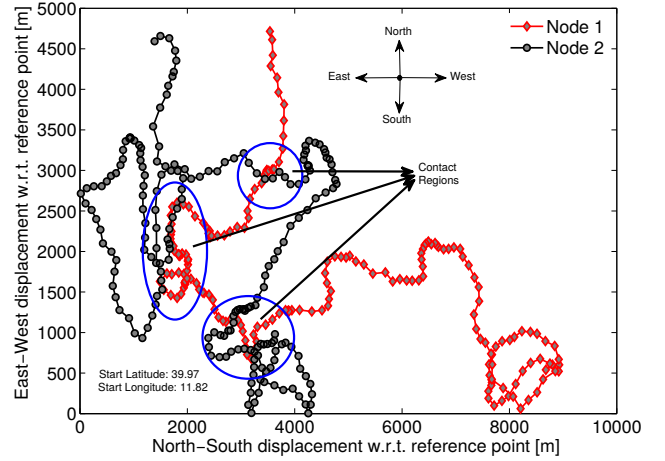


Figure 3: Example of Gauss-Markov mobility pattern for two AUVs, with self-correlation $\alpha = 0.8$ (view from above). The circles and diamonds correspond to the locations where each AUV randomly picks a new velocity vector. Three examples of areas where meeting occurred are contoured in blue.

are finally averaged over 50 runs to keep the confidence interval size within 5% of the displayed average values.

We focus on three protocol performance metrics: the packet delivery ratio (PDR), the overhead ratio, and the end-to-end delay. The PDR measures the fraction of packets generated by the nodes and correctly received by the sink, and is defined as

$$\text{PDR} = \frac{\sum_{i=1}^N R_i}{\sum_{i=1}^N G_i}, \quad (14)$$

where N is the number of nodes in the network, G_i is the number of unique packets generated by node i during a given simulation, and R_i is the number of unique packets correctly received by the sink before the deadline.

The overhead ratio is defined as the average number of additional copies received by the sink per each packet correctly received, and measures the efficiency of replication-based protocols. It is defined as

$$\text{OVH} = \frac{\sum_{j \in \mathcal{R}} C_j}{|\mathcal{R}|}, \quad (15)$$

where \mathcal{R} is the set of all unique packets received by the sink from all nodes, such that $|\mathcal{R}| = \sum_{i=1}^N R_i$, and C_j is the total number of copies received for packet j . Note that the minimum value achievable by the OVH metric is 1, meaning that the sink received exactly one copy for each unique packet collected.

Finally, the end-to-end delay is defined as:

$$\text{E2E} = \frac{\sum_{j \in \mathcal{R}} t_{j,R} - t_{j,G}}{|\mathcal{R}|}, \quad (16)$$

where $t_{j,R}$ is the time packet $j \in \mathcal{R}$ is received by the sink, and $t_{j,G}$ is the time packet j was generated.

We compared the performance of UDTN-Prob against a version of the SAW protocol [31] and with the more advanced SAW version described in [36], where the probability that a

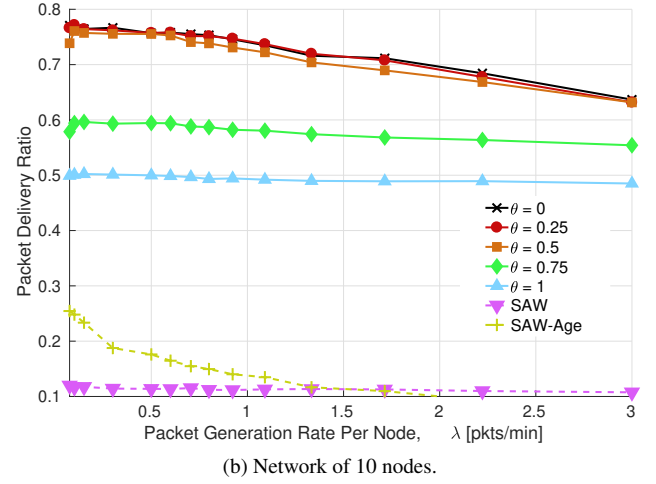
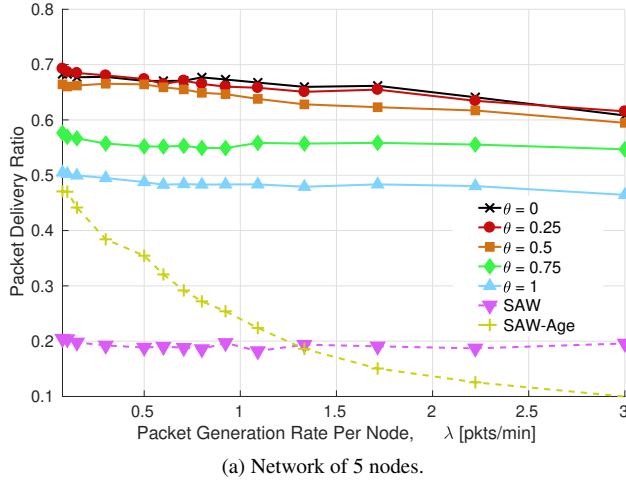


Figure 4: PDR vs. packet generation rate per node for different network sizes. UDTN is able to keep the PDR relatively stable for all values of θ . The SAW protocol, albeit enhanced with an ARQ scheme, achieves consistently worse performance.

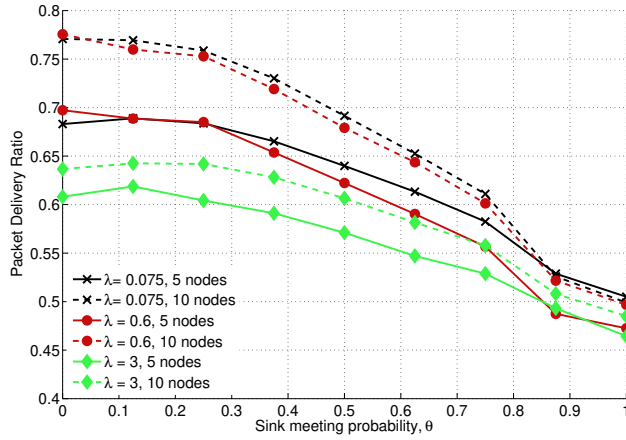


Figure 5: PDR vs. θ for different values of the packet generation rate and the number of nodes. In all cases, the preferred values of θ range between 0 and 0.5.

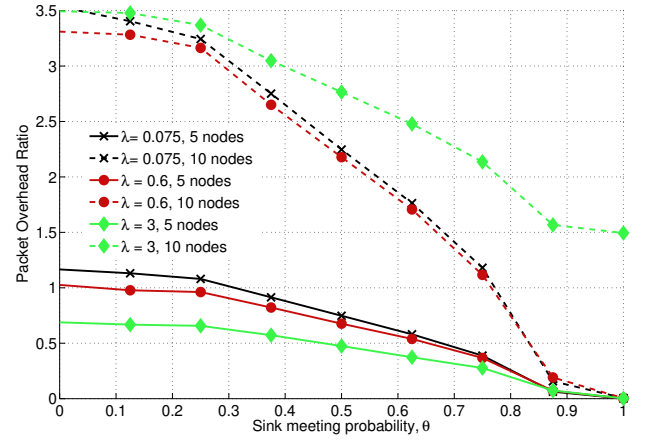


Figure 6: Overhead vs. θ for different values of the packet generation rate and the number of nodes. Increasing θ constrains the nodes to relay packet only if the relay has increasingly better changes to meet the sink, thus reducing the replication overhead.

node receives a copy of a packet increases if the encounter between this node and the destination has been overdue for a longer time. The latter has been dubbed “SAW-Age” in the graphs. We enhanced both versions of SAW with a stop-and-wait ARQ error control technique. This makes the comparison with UDTN more fair, given that UDTN also compensates for channel errors through an ARQ scheme. The comparison is detailed in the following section.

B. UDTN-PROB PERFORMANCE ANALYSIS

In the following performance evaluation we focus on three main metrics, namely the packet delivery ratio (PDR), the end-to-end delivery delay, and the overhead represented by the excess replicas injected in the network. We analyze these metrics as a function of network parameters such as the packet generation rate per node, the number of nodes and their speed, as well as protocol-specific parameters such as

the threshold θ on the sink meeting probability.

We start by considering the PDR, defined as the fraction of packets correctly received by the sink within their deadline, as a function of the packet generation rate per node. We recall that several events can lead to a failed end-to-end delivery, such as repeated communication failures, a packet exceeding the delivery deadline, or being still in the queue of some node at the end of a simulation run. We plot the PDR for different values of θ in Figs. 4a and 4b, for different numbers of network nodes, namely 4 or 9 nodes plus 1 sink, for a total of 5 and 10 nodes, respectively. The packet delivery ratio achieved by SAW is also reported.

We first note that UDTN-Prob outperforms SAW in all cases. Interestingly, when $\theta = 0$, UDTN-Prob employs the same transmission strategy as SAW, namely binary spraying, but in addition *i*) it allows both nodes in contact to exchange data packets by sharing the contact time, and *ii*) it employs an

efficient ARQ scheme to correct transmission errors. Despite the initial handshake, which effectively invests part of the contact duration to coordinate the nodes, the performance improvement enabled by contact time sharing and ARQ is significant, and can be observed by comparing the SAW PDR curve against the UDTN-Prob PDR curve for $\theta = 0$.

A value of $\theta > 0$ corresponds to enabling the prediction of inter-meeting times with the sink, and tends to make nodes avoid the transmission of packets that are likely to exceed their delivery deadline before the receiver meets the sink. For this reason, the PDR decreases slightly until $\theta \leq 0.5$, whereas a more substantial decrease is observed for $\theta = 0.75$ and $\theta = 1$. Note that this is expected, as from a statistical point of view a node would meet the sink with very high probability typically over long periods of time. In turn, this means that a sender will avoid the transmission of all packets whose deadline would expire before such time, and more packets will be dropped for this reason.

By comparing Figs. 4a and 4b, we observe that the PDR achieved by UDTN-Prob is higher than SAW's for all values of θ . In fact, with more nodes in the same network area, the contact opportunities are more frequent, and the possibility to deliver packets to the sink correspondingly increases. It is worth noting that this does not apply to SAW: the reason is again that in SAW only one transmits to the other during a contact, and moreover no choice over the packet to be transmitted is performed as a function of the packet deadline. Despite SAW-Age being an advancement over SAW, its working assumption that overdue contact are more likely does not always yield better performance in practice. Due to the rules of SAW-Age, if the source of a packet has not met the destination for a long time interval, its probability to meet the destination is estimated to be higher than that of any encountered node. In turn, the source will stop forwarding packets to encountered nodes. For high values of the packet generation rate, this means that most packets will be held by their generating node until this node meets the sink. However, the encounter with the sink is also limited in time, and the node typically has no time to forward its packets. For the highest packet generation rates, this tends to reduce the packet delivery ratio to very low values, which become worse than the PDR achieved by the normal version of SAW in some cases. Because of the limited performance of SAW, in the remainder of this section we will focus mostly on the effect of UDTN-Prob parameters on the PDR and end-to-end delay achieved by the protocol.

Figs. 5 and 6 show, respectively, the PDR and the overhead (defined as the ratio between the number of additional replicas received by the sink to the number of unique packets received) as a function of θ , for different values of the number of nodes and of the packet generation rate per node. As a general observation, lower values of θ lead to a higher PDR, because the nodes tend to exchange more packets upon contact. As a consequence, the overhead is also quite high, and more so for a greater number of nodes in the network and for a higher packet generation rate per node, as confirmed

by Fig. 6. The curves in Fig. 5 confirm that the PDR does not decrease substantially, as long as $\theta \leq 0.5$. A more pronounced decrease is observed in the case of a 10-node network, but in any event the decrease is still limited. In this range of values of θ , the overhead decreases as well. The decrease is more apparent in denser networks with higher packet generation rates. From this evaluation, we conclude that setting θ in the range from 0.4 to 0.6 achieves a good tradeoff between PDR and overhead. The exact value that should be set remains a design choice, and depends on the requirements of the application to be supported by the network.

As in terrestrial radio networks, in underwater DTNs the movement speed of the nodes also affects the frequency of contact occurrence and the statistics of the contact duration. In particular, the speed of the nodes is normally low with respect to the areas covered by typical missions, and as a result contacts tend to occur more rarely. If the AUVs can afford a higher movement speed, contacts may be shorter but occur more frequently, and the performance of a DTN protocol is expected to improve. This is confirmed by Fig. 7, where we show the PDR vs. θ for a node speed of 2 and 4 m/s in a network of 5 nodes. We observe that a higher node speed yields an increase of the PDR: this increase takes place for all values of the packet generation rate λ , and is larger for higher values of θ . This is because more frequent contacts increase the probability that the sink is met within a sufficiently short time: as a consequence, UDTN-Prob allows the nodes to exchange more packets per contact event.

The length of the inter-contact time is a key parameter for UDTN-Prob, and is leveraged already during the handshake phase in order to select the relay that yields the longest contact time. This option makes it possible for two nodes to exchange packets over a more stable connection. To show that this choice yields better performance than typical alternatives, in Fig. 9 we compare the PDR performance of UDTN-Prob against λ under the relay selection policy described in Section III-B (whereby contact is established by sending a Response packet to the node providing the longest contact time among those that sent the Info message) and under a different policy where the Response is sent to the node that yields the shortest expected time to meet the sink for a given value of the sink meeting probability θ . From Fig. 9 we observe that our policy always yields better PDR values. This is the case both for $\theta = 0.25$ (black crosses vs. red circles) and for $\theta = 0.5$ (orange stars vs. green diamonds). This confirms that a relay providing an earlier sink meeting time may be associated with a short contact, which in turn would reduce the capability of the nodes to transfer data packets.

Another design choice of UDTN-Prob is to employ the conservative estimate τ'_{ij} in (9) instead of the actual estimate τ_{ij} measured from the initial UDTN-Prob exchange, as schematically shown in Fig. 1. Fig. 10 shows the PDR against λ for the standard UDTN-Prob design (conservative τ'_{ij} estimate, black crosses) against the use of the non-

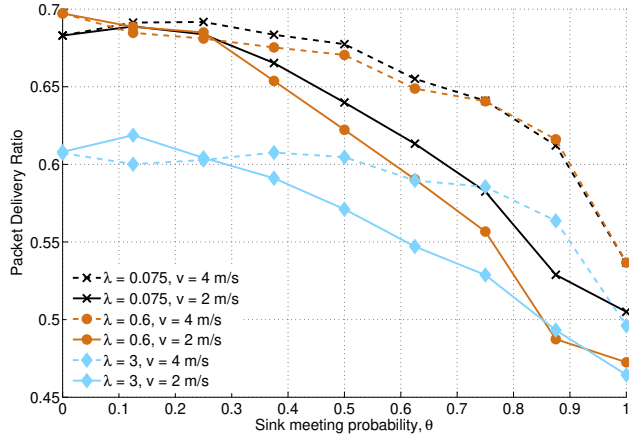


Figure 7: PDR vs. θ for different values of the packet generation rate and the node speed. A higher speed promotes more frequent contacts, which in turn improves the PDR for higher values of θ .

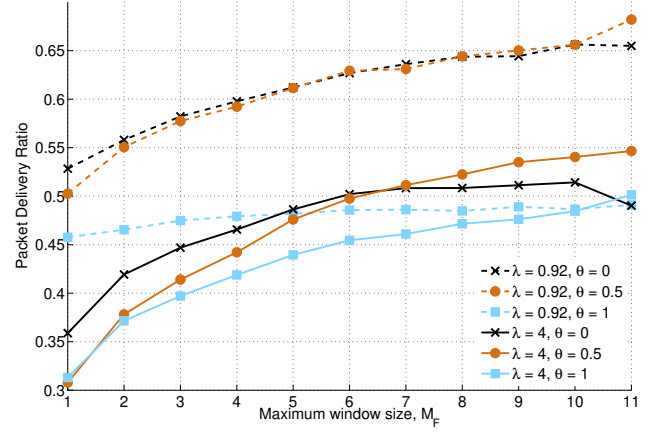


Figure 8: PDR vs. M_F for different values of θ and of the packet generation rate. An exceedingly low value of M_F forces shorter packet exchanges, and tends to reduce the PDR as a result of the less efficient use of the contact time.

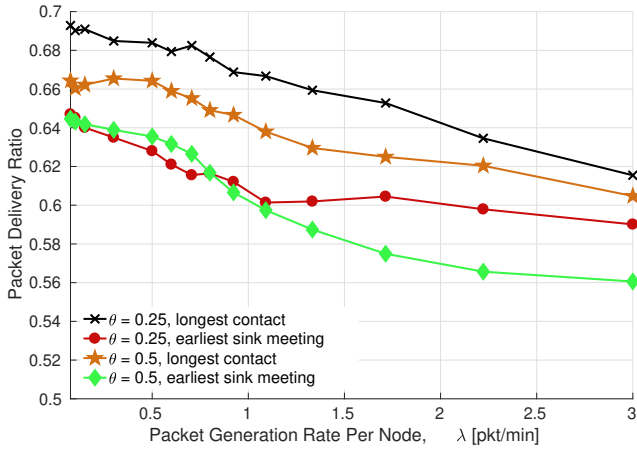


Figure 9: PDR vs. λ for different contact node selection policies. Selecting the node yielding the longest contact time leads to a higher PDR than choosing based on the earliest expected sink meeting time.

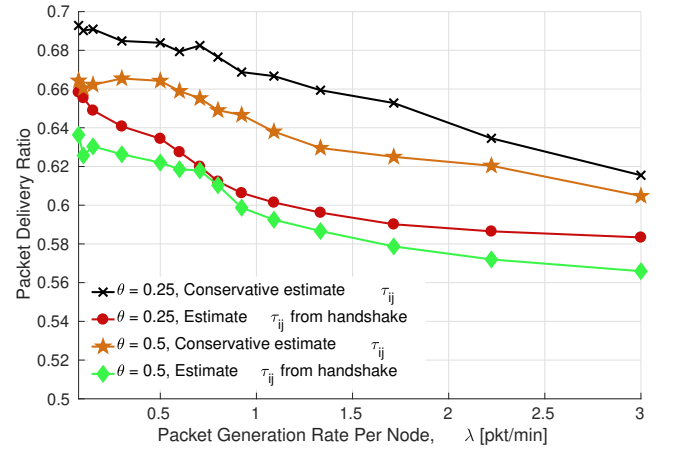


Figure 10: PDR vs. λ achieved by setting the timing of the TDD SR-ARQ packet exchange according to the conservative estimate of τ_{ij}^c from (9) (red crosses). Note that the black and orange curves are the same as in Fig. 9.

corrected τ_{ij} estimate of the round-trip time (red circles). As in Fig. 9, we show both cases of $\theta = 0.25$ and $\theta = 0.5$. Our conservative approach offers up to about 14% better PDR with respect to the direct use of τ_{ij} .

The last result we show in this set refers to the use of a fixed window size M_F , an option discussed in Section III to operate UDTN-Prob in those cases where the contact time estimation fails for any reason. Fig. 8 shows the PDR vs. M_F for different values of the packet generation rate and of the number of nodes. We remark that the actual transmit window used during a contact is computed as in Eq. (10). We observe that forcing the window size to an unnecessarily low value (e.g., $M_F \leq 4$) adversely affects the PDR, as it limits the number of packets exchanged per contact. Should the nodes be still within each other's transmission range when the packet exchange finishes, they would have to perform a fresh handshake in order to transmit additional packets, wasting time that could be otherwise used to transmit data.

Conversely, a sufficiently high value (e.g., $M_F \geq 5$) exploits the contact time more. This distributes more packets to different nodes, and ultimately achieves a higher PDR.

In the context of DTN protocols, it is also of interest to evaluate the end-to-end delivery delay performance, defined as the time that elapses from when a packet is generated by a node to when it is delivered to the sink for the first time. Fig. 11 shows how such delay varies with θ in networks of 5 and 10 nodes. Increasing θ makes senders give the highest priority to newer packets, which have a longer residual lifetime: as a result, the average delivery delay decreases. We also observe that the presence of 10 nodes and the more frequent contacts that take place tend to increase the average delay: this is an expected consequence of the higher PDR obtained in the presence of 10 nodes (see also Fig. 4b). By way of contrast, the delay achieved by the two versions of SAW considered in Fig. 4 is at between 2.5 and 20 times higher, showing that UDTN achieves a more effective

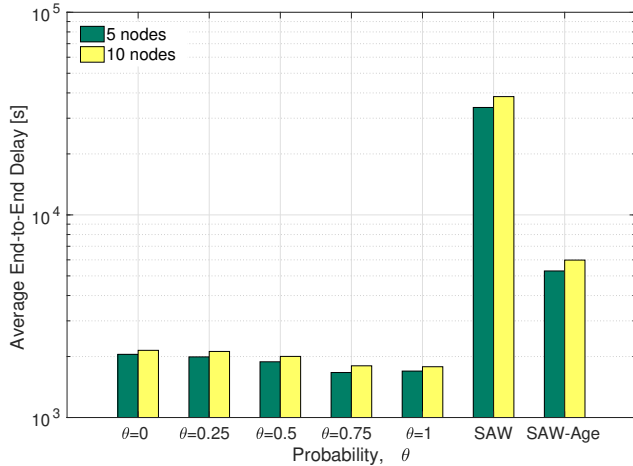


Figure 11: End-to-end delay vs. θ in a network of 5 and 10 nodes. Larger values of θ imply that only the packets with a longer time-to-live are delivered: the delay decreases as a result. The delay of both SAW versions is significantly higher.

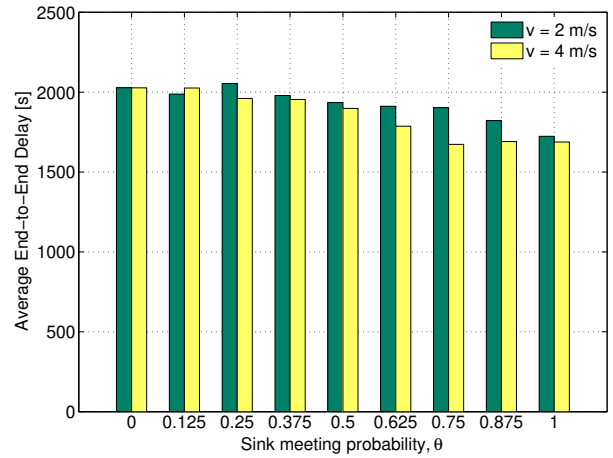


Figure 12: End-to-end delay vs. θ for different node speeds in a network of 5 nodes. A higher movement speed provides more frequent contacts and decreases the delivery delay.

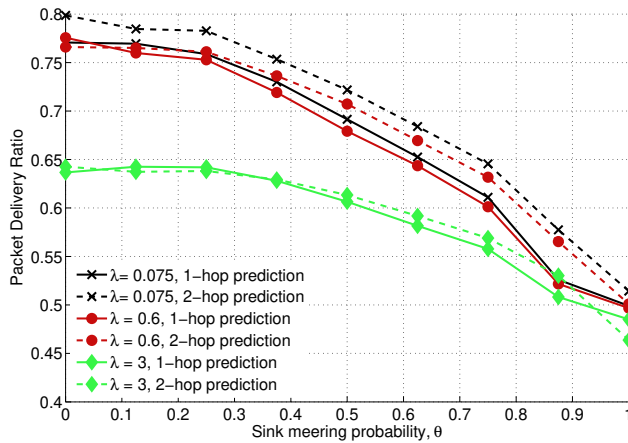


Figure 13: PDR vs. θ in a network of 10 nodes for different values of the packet generation rate. Two-hop sink θ -meeting time prediction offers negligible improvement over one-hop prediction.

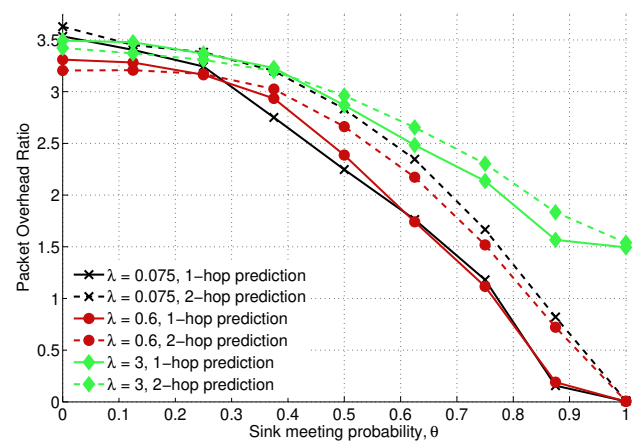


Figure 14: Overhead vs. θ in a network of 10 nodes for different values of the packet generation rate. Two-hop sink θ -meeting time prediction requires higher overhead.

redistribution of the packets upon the occurrence of contacts. For a higher node speed of 4 m/s, Fig. 12 shows that the end-to-end delivery delay becomes lower, especially for $\theta > 0.5$.

C. ONE- AND TWO-HOP SINK MEETING PREDICTION

All the results discussed so far have been obtained by employing one-hop sink θ -meeting time predictions (see Section III) to decide on the priority of the packets to be transmitted. In this last section, we discuss a comparison between the one-hop and the two-hop sink meeting prediction schemes. Fig. 13 and Fig. 14 respectively show the PDR and the overhead achieved by UDTN-Prob via either scheme in a network of 10 nodes and for different values of the packet generation rate. The two-hop technique provides only a marginally higher PDR. This improvement is due to the prediction of meetings with the sink through two intermediate hops, which generally results in a shorter end-to-end inter-

meeting time, and in the exchange of packets whose residual time-to-live is also short. This effect is observed especially for values of θ approaching 1, which limit transmissions substantially when the one-hop prediction technique is used. The direct consequence of the higher PDR is an increase of the overhead, as seen in Fig. 14. In fact, the better effectiveness of the two-hop technique comes at the price of a larger amount of information that needs to be stored as well as exchanged by the nodes upon contact. Therefore, given the small PDR improvement offered by the two-hop prediction technique with respect to the one-hop technique, we argue that extending beyond one-hop θ -meeting time predictions is at best marginally better. Considering the additional complexity of the two-hop prediction scheme, we conclude that the one-hop technique is adequate to achieve good performance in underwater DTNs.

V. CONCLUSIONS

We proposed a replication-based routing protocol named Underwater Delay-Tolerant Networking with Probabilistic spraying (UDTN-Prob). The protocol estimates the contact duration based on information exchanged when two nodes meet, and decides on the order of priority of the packets to be transmitted based on the time required by the relay to meet the sink in the future with a prescribed probability. The statistics of future meeting times are inferred initially from synthetic mobility models that well approximate the behavior of actual nodes, and are updated as the network runs. This allows to make the best use of the infrequent contacts among the nodes. Through the knowledge of the statistics of inter-contact times with the sink, the nodes can transmit only those packets that have a sufficiently high chance of being delivered to the sink before their time-to-live expires. The actual exchange of data packets is error controlled through a selective repeat ARQ scheme that interleaves data packet transmissions and ACK reception using time division duplexing. We show that UDTN-Prob achieves satisfactory PDR and limited overhead in the presence of different packet generation rates and node speed, and outperforms the well known Spray-and-Wait routing protocol.

We consider also the more complex case where the statistics of the inter-meeting times between any other node and the sink are known to the nodes. This allows the nodes to make two-hop sink meeting time predictions, which marginally improves the PDR, but implies higher overhead. Therefore, as a byproduct of our work, we conclude that two-hop sink meeting time predictions are not worth the additional effort required to collect the statistics of node meetings for the whole network and to keep them up to date.

ACKNOWLEDGMENTS

A preliminary version of this work has been presented at the MTS/IEEE OCEANS 2012 Conference, Yeosu, South Korea [4]. The authors would like to thank Prof. Simin Nadjm-Tehrani (Linköping University, Sweden) for sharing an ns2 version of the Spray-and-Wait protocol.

References

- [1] I. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," Elsevier's Ad Hoc Networks, vol. 3, no. 3, May 2005.
- [2] E. M. Sozer, M. Stojanovic, and J. G. Proakis, "Underwater acoustic networks," IEEE J. Ocean. Eng., vol. 25, no. 1, pp. 72–83, Jan. 2000.
- [3] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," Marine Tech. Soc. Journal, vol. 42, no. 1, pp. 103–116, spring 2008.
- [4] S. Azad, P. Casari, and M. Zorzi, "Coastal patrol and surveillance networks using AUVs and delay-tolerant networking," in Proc. of MTS/IEEE OCEANS, Yeosu, South Korea, May 2012.
- [5] C. E. Perkins and E. M. Royer, "Ad Hoc on-demand distance vector routing," in IEEE WMCSA, New Orleans, LA, US, Feb. 1999.
- [6] P. Xie, J.-H. Cui, and L. Lao, "VBF: vector-based forwarding protocol for underwater sensor networks," UCONN CSE, Tech. Rep. UbiNet-TR05-03 (BECAT/CSE-TR-05-6), Feb. 2005.
- [7] M. Goetz, S. Azad, P. Casari, I. Nissen, and M. Zorzi, "Jamming-resistant multi-path routing for reliable intruder detection in underwater networks," in Proc. of ACM WUWNet, Seattle, Washington, USA, Dec. 2011.
- [8] 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. of MTS/IEEE OCEANS, Virginia, USA, Oct. 2012.
- [9] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni, "Recognizing exponential inter-contact time in VANETs," in Proc. IEEE INFOCOM Mini-Conference, San Diego, CA, Mar. 2010.
- [10] H. Zhu, M. Li, L. Fu, G. Xue, Y. Zhu, and L. M. Ni, "Impact of traffic influxes: Revealing exponential intercontact time in urban VANETs," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 8, pp. 1258–1266, Aug. 2011.
- [11] Y. Hu, H. Wang, C. Xia, W. Li, and Y. Yang, "On the distribution of inter contact time for DTNs," in Proc. IEEE LCN, Denver, CO, Oct. 2012.
- [12] Y. Li, Y. Jiang, D. Jin, L. Su, L. Zeng, and D. Wu, "Energy-efficient optimal opportunistic forwarding for delay-tolerant networks," IEEE Trans. Veh. Technol., vol. 59, no. 9, pp. 4500–4512, Nov. 2010.
- [13] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," IEEE Trans. Mobile Comput., vol. 6, no. 6, pp. 606–620, Jun. 2007.
- [14] T. Karagiannis, J. Y. L. Boudec, and M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," IEEE Trans. Mobile Comput., vol. 9, no. 10, pp. 1377–1390, Oct. 2010.
- [15] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-walk nature of human mobility," IEEE/ACM Trans. Netw., vol. 19, no. 3, pp. 630–643, Jun. 2011.
- [16] H. Xu and G. Li, "Multi-AUVs formation control with acoustic communication constraints," in Proc. MTS/IEEE OCEANS, San Diego, CA, Sep. 2013.
- [17] N. Tsiogkas, G. Frost, N. Monni, and D. Lane, "Facilitating multi-AUV collaboration for marine archaeology," in Proc. MTS/IEEE OCEANS, Genova, Italy, May 2015.
- [18] N. Li, B. Cürüklü, J. Bastos, V. Sucasas, J. A. S. Fernandez, and J. Rodriguez, "A probabilistic and highly efficient topology control algorithm for underwater cooperating AUV networks," MDPI Sensors, vol. 17, no. 5, May 2017.
- [19] C. Barngrover, R. Kastner, T. Denewiler, and G. Mills, "The stingray AUV: A small and cost-effective solution for ecological monitoring," in Proc. MTS/IEEE OCEANS, Kona, HI, Sep. 2011.
- [20] B. Y. Queste, K. J. Heywood, J. Kaiser, G. A. Lee, A. Matthews, S. Schmidtke, C. Walker-Brown, and S. W. Woodward, "Deployments in extreme conditions: Pushing the boundaries of seaglider capabilities," in Proc. IEEE/OES AUV, Southampton, UK, Sep. 2012.
- [21] M. E. Furlong, D. Paxton, P. Stevenson, M. Pebody, S. D. McPhail, and J. Perrett, "Autosub long range: A long range deep diving auv for ocean monitoring," in Proc. IEEE/OES AUV, Southampton, UK, Sep. 2012.
- [22] M. Eichhorn, R. Taubert, C. Ament, M. Jacobi, and T. Pfuetzenreuter, "Modular AUV system for sea water quality monitoring and management," in Proc. MTS/IEEE OCEANS, Bergen, Norway, Jun. 2013.
- [23] A. Bodenmann, B. Thornton, and T. Ura, "Visual 3D mapping to measure hydrothermal deposit growth rates at a man-made deep sea vent," in Proc. IEEE/OES AUV, Oxford, MI, Oct. 2014.
- [24] B. Bayat, A. Crespi, and A. Ijspeert, "Envirobot: A bio-inspired environmental monitoring platform," in Proc. IEEE/OES AUV, Tokyo, Japan, Nov. 2016.
- [25] "Aquacomm: Underwater wireless modem." [Online]. Available: http://www.dspcomm.com/products_aquacomm.html
- [26] "Link-quest modem," <http://www.link-quest.com/html/models1.htm>.
- [27] "S2C R 48/78 underwater acoustic modem." [Online]. Available: http://www.evologics.de/en/products/acoustics/s2cr_48_78.html
- [28] H. S. Dol, P. Casari, T. van der Zwan, and R. Otnes, "Software-defined underwater acoustic modems: Historical review and the NILUS approach," IEEE J. Ocean. Eng., Jul. 2017.
- [29] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," IEEE Wireless Commun., vol. 47, no. 1, pp. 84–89, Jan. 2009.
- [30] P. Casari and M. Zorzi, "Protocol design issues in underwater acoustic networks," Elsevier Computer Commun., vol. 34, no. 17, pp. 2013–2025, Nov. 2011.
- [31] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: an efficient routing scheme for intermittently connected mobile networks," in Proc. of ACM WDTN, Philadelphia, USA, Aug. 2005, pp. 252–259.
- [32] M. Chitre and W. S. Soh, "Reliable point-to-point underwater acoustic data transfer: To juggle or not to juggle?" IEEE J. Ocean. Eng., vol. 40, no. 1, pp. 93–103, Jan. 2015.

- [33] S. Azad, P. Casari, and M. Zorzi, "The Underwater Selective Repeat error control protocol for multiuser acoustic networks: Design and parameter optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4866–4877, Oct. 2013.
- [34] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Department of Computer Science, Duke University, Tech. Rep. CS-2000-06, Apr. 2000.
- [35] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.
- [36] D. Pan, M. Lin, L. Chen, and J. Sun, "An improved spray and wait with probability choice routing for opportunistic networks," *Journal of Networks*, vol. 7, no. 9, pp. 1486–1492, Sep. 2012.
- [37] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. of ACM SIGCOMM*, Kyoto, Japan, Aug. 2007, pp. 373–384.
- [38] Z. Guo, B. Wang, and J.-H. Cui, "Generic prediction assisted single-copy routing in underwater delay tolerant sensor networks," *Elsevier Ad Hoc Networks*, vol. 11, no. 3, pp. 1136–1149, May 2013.
- [39] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proc. of ACM MobiHoc*, Jul. 2003.
- [40] D. Hay and P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks," in *Proc. IEEE WONS 2009*, Snowbird, UT, Feb. 2009.
- [41] G. Neglia, X. Zhang, J. Kurose, D. Towsley, and H. Wang, "On optimal packet routing in deterministic DTNs," in *Proc. IEEE VTC*, Dresden, Germany, Jun. 2013.
- [42] H. Kang and K. Dongkyun, "Vector routing for delay tolerant networks," in *Proc. IEEE VTC-Fall*, Calgary, Canada, Sep. 2008.
- [43] —, "HVR: history-based vector routing for delay tolerant networks," in *Proc. ICCCN*, San Francisco, CA, Aug. 2009.
- [44] H. Zhu, S. Chang, M. Li, K. Naik, and S. Shen, "Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011.
- [45] H. Wen, F. Ren, J. Liu, C. Lin, P. Li, and Y. Fang, "A storage-friendly routing scheme in intermittently connected mobile network," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1138–1149, Jan. 2011.
- [46] Q. Yuan, I. Cardei, and J. Wu, "An efficient prediction-based routing in disruption-tolerant networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 19–31, May 2012.
- [47] Z. Guo, G. Colombi, B. Wang, J.-H. Cui, D. Maggiorini, and G. P. Rossi, "Adaptive routing underwater delay/disruption tolerant sensor networks," in *Proc. WONS*, Garmisch-Partenkirchen, Germany, Jan. 2008.
- [48] S. Tolba, M. Hakami, A. Mithdir, Y. Zhu, S. Le, and J. H. Cui, "Underwater delay tolerant routing in action," in *Proc. MTS/IEEE OCEANS*, Hampton Roads, VA, Oct. 2012.
- [49] D. Merani, A. Berni, J. Potter, and R. Martins, "An underwater convergence layer for disruption tolerant networking," in *Proc. BCFIC*, Riga, Latvia, Feb. 2011.
- [50] A. Berni, D. Merani, J. Potter, and R. Been, "Heterogeneous system framework for underwater networking," in *Proc. IEEE MILCOM*, Baltimore, MD, Nov. 2011.
- [51] T. Le, P. Du, and M. Gerla, "A relay selection strategy based on power-law and exponentially distributed contacts in DTNs," in *Proc. IEEE MASS*, Orlando, FL, Oct. 2017.
- [52] P. Gjanci, C. Petrioli, S. Basagni, C. A. Phillips, L. Bölöni, and D. Turgut, "Path finding for maximum value of information in multi-modal underwater wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 404–418, Feb. 2018.
- [53] M. Zorzi, P. Casari, N. Baldo, and A. F. Harris III, "Energy-efficient routing schemes for underwater acoustic networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 9, pp. 1754–1766, Dec. 2008.
- [54] 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.
- [55] K. Stamatiou, P. Casari, and M. Zorzi, "The throughput of underwater networks: Analysis and validation using a ray tracing simulator," *IEEE Trans. Wireless Commun.*, vol. 12, no. 3, pp. 1108–1117, Mar. 2013.
- [56] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W function," *Advances in Computational Mathematics*, vol. 5, pp. 329–359, Dec. 1996.
- [57] D. E. Lucani, M. Stojanovic, and M. Medard, "Random linear network coding for time division duplexing: When to stop talking and start listening," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brasil, Apr. 2009.
- [58] J. P. Rohrer and A. N. Mauldin, "Implementation of epidemic routing with ip convergence layer in ns-3," in *Proc. WNS3*, Surathkal, India, Jun. 2018.
- [59] P. Casari, C. Tapparello, I. Calabrese, F. Favaro, G. Toso, S. Azad, R. Masiero, and M. Zorzi, "Open-source suites for the underwater networking community: WOSS and DESERT Underwater," *IEEE Network S.I. Open Source for Networking: Development and Experimentation*, vol. 28, no. 5, pp. 38–46, Sep. 2014.
- [60] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 1st ed. Springer, 2008.
- [61] N. Baldo, M. Miozzo, F. Guerra, M. Rossi, and M. Zorzi, "Miracle: The multi-interface cross-layer extension of ns2," *EURASIP J. on Wireless Commun. and Networking*, vol. 2010, no. 1, pp. 761–792, Mar. 2010.
- [62] "General bathymetric chart of the oceans." [Online]. Available: www.gebco.net
- [63] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *Proc. of IEEE INFOCOM*, New York, NY, Mar. 1999, pp. 1377–1384.



PAOLO CASARI (S'05–M'08–SM'13) received the Ph.D. degree in information engineering from the University of Padova, Italy, in 2008. He was on leave from the Massachusetts Institute of Technology in 2007. In 2015, he joined the IMDEA Networks Institute, Madrid, Spain, where he currently leads the Ubiquitous Wireless Networks Group. He collaborated to several funded projects including CLAM (FP7), RACUN (EDA), and several U.S. ARO, ONR, and NSF initiatives. He is a PI of the NATO SPS project ThreatDetect and a Scientific Coordinator of the EU H2020 RECAP and SYMBIOSIS projects. He was the Technical Manager of the NAUTILUS and WISEWAI projects. His research progressively focused on underwater communications and networks. His research interests include many aspects of underwater communications, such as channel modeling and simulation, network protocol design and evaluation, localization, and field experiments. He was a co-recipient of the IEEE WPNC 2017 Best Paper Award. He serves on the Editorial Board of the *IEEE Transactions on Mobile Computing* and of the *IEEE Transactions on Wireless Communications*, and collaborated with the Organizing Committee of several conferences. He was a Guest Editor of a Special Issue on Underwater Communications and Networking of the *Hindawi Journal of Electrical and Computer Engineering* and of a Special Issue of *IEEE Access* on Underwater Wireless Communications and Networking.



SAIFUL AZAD (M'15) received the B.Sc. degree in computer and information technology from IUT, Bangladesh, the M.Sc. degree in computer and information engineering from IIUM, Malaysia, and the Ph.D. degree in information engineering from the University of Padova, Italy, in 2013. He joined the Department of Computer Science, American International University-Bangladesh, as a Faculty Member. He is currently with the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Malaysia. He is one of the developers of the DESERT underwater simulator. He has authored many scientific papers published in international renowned peer-reviewed journals and conferences. His research interests also lie in data mining, machine learning, information processing, designing and implementing communication protocols for any network architecture, QoS issues, network security, and simulator design. He is a fellow of IBM CoE, Malaysia. He also serves as a reviewer and a technical program committee member for many renowned peer-reviewed journals and conferences.



MARCO ZANFORLIN is currently a Research Collaborator with the Department of Information Engineering, University of Padova, Italy. His expertise lies in the simulation of different aspects of wireless networks and services. In the context of the EU FP7 research project MEDIEVAL, he collaborated to the implementation of algorithms to transfer video contents from content delivery networks to consumers through mobile networks.

Within the SIGNET Group, Department of Information Engineering, he also collaborated to the early recognition of video contents in cognitive networks, to the implementation of different mobility models for underwater acoustic networks, and to the simulation of underwater delay tolerant routing protocols.



MICHELE ZORZI (S'89–M'95–SM'98–F'07) received the Laurea and Ph.D. degrees in electrical engineering from the University of Padova in 1990 and 1994, respectively. During 1992–1993, he was on leave at the University of California at San Diego (UCSD). In 1993, he joined the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, as a Faculty Member. He was with the Center for Wireless Communications, UCSD, for three years. In 1998, he joined the School of

Engineering, University of Ferrara, Italy, where he became a Professor in 2000. Since 2003, he has been a Faculty Member with the Information Engineering Department, University of Padova. His current research interests include performance evaluation in mobile communications systems, WSN, Internet-of-Things, cognitive communications and networking, 5G mm-wave cellular systems, and underwater communications and networks. He is currently an Editor-in-Chief of the *IEEE Transactions on Cognitive Communications and Networking*. He was an Editor-In-Chief of *IEEE Wireless Communications* from 2003 to 2005 and of the *IEEE Transactions on Communications* from 2008 to 2011. He served as a Member-at-Large for the Board of Governors of the IEEE Communications Society from 2009 to 2011 and its Director of Education from 2014 to 2015.

...