A Proposal for a Cross-Layer, P+S Approach in Underwater Communication: The Underwater Data Exchange Protocol

Brama, Riccardo; Arvonen, Pertti; Andersen, Nicklas Sindlev; Giambarrasi, Denis

Published in:
Information & Security

DOI:
10.11610/isij.5538

Publication date:
2024

Document version:
Final published version

Information & Security

R. Brama et al.
vol. 55, no. 2 (2024): 165-182
https://doi.org/10.11610/isij.5538

Published since 1998

ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

A Proposal for a Cross-Layer, P+S Approach in Underwater Communication: The Underwater Data Exchange Protocol

Riccardo Brama [1] (✉), *Denis Giambarrasi* [2], *Nicklas Sindlev Andersen* [3], *and Pertti Arvonen* [4]

[1] *Independent, Carovigno, Italy*

[2] *Ratio Computers, Valpiana, Italy, https://www.ratio-computers.com/en*

[3] *Department of Mathematics and Computer Science, Odense, Denmark https://www.sdu.dk/en/om-sdu/institutter-centre/imada_matematik_og_datalogi*

[4] *UWIS Oy, Turku, Finland, https://uwis.fi/en*

ABSTRACT:

Communication in underwater environments is extremely challenging: bandwidth limitation, latency, and extreme packet losses are only the tip of the iceberg. Usually, just a few bytes can be conveyed, and due to the lack of common standards in the physical layer, the problem becomes even more complex. An accurate network architecture, aiming to maximize message portability in different networking scenarios while supporting significantly different transportation and physical layers, becomes mandatory. Thus, a cross-layer approach is proposed, moving some networking and transport details to the application layer while leveraging the publish-subscribe pattern to limit shared resource usage.

ARTICLE INFO:

RECEIVED: 13 SEP 2024
REVISED: 20 OCT 2024
ONLINE: 31 OCT 2024

KEYWORDS:

underwater communication, application protocol, communication protocol, security, publish-subscribe, military communication

✉ E-mail: r.brama@ieee.org

Introduction

As soon as a dive starts, the lack of ways to communicate with a teammate or with the surface becomes evident for a diver: the underwater environment is the only place humans know in which it is so hard to interact with others. Many methods have been explored to fulfill the need to share information, from hand signs to cables, from light pulses [1] to ultrasounds.

Despite being characterized by very limited bit rates, acoustic communication (AUC) is considered one of the most effective ways to convey, at long distances, information underwater: optical communication—despite being extremely efficient and characterized by the highest bitrate—is deeply affected by sediments, thermoclines and by visual impairments; electromagnetic communication [2] is viable only within the diver's personal space and, for military applications, it is not welcome due to its detectability.

In recent years, there has been an increased interest in underwater networking, not just at the PHY level,[1,3] but also at the MAC [4, 5] and network levels.[6, 7] However, the focus has largely been on individual environments, either underwater or on the surface.[8] Yet, given the inherent integration between these two realms, a holistic approach that minimizes the need for information trans-coding is not just desirable but necessary.

This paper introduces a proposal for a very compact cross-layer optimized communication protocol and is structured as follows: Section 2 describes the networking scenario in which the proposed Underwater Data eXchange Protocol (UDXP) has been developed, Section 3 reports the packet format together with an introduction to messages topics. In Section 5, multicast groups are suggested as a way to minimize the network traffic, especially when dealing with team communication. Section 6 describes a proposal to improve security in underwater communication. Finally, conclusions are drawn in Section 7.

The Network Scenario

The targeted networking scenario is shown in Figure 1. Here, every device exchanges information, either directly or by means of assisted relaying, overcoming each of the three identified different spatial boundaries:

1. Diver Personal Area Network (DPAN), encompassing devices that are geared on the diver, usually not farther than 1m;
2. Underwater Local Area Network (ULAN), connecting devices within the underwater environment up to 2km;
3. Surface Local Area Network (SLAN), linking devices on the surface, whether equipped on a vessel, flying nearby vessels or orbiting in space.

In such a complex scenario, a uniform application layer with common syntax and semantics is highly desirable to minimize the need for information re-encoding: a desirable feature until the surface is reached since, at depth, battery-powered devices are largely employed. In fact, at depth, strict energy constraints necessitate minimizing power consumption to maximize network lifetime.

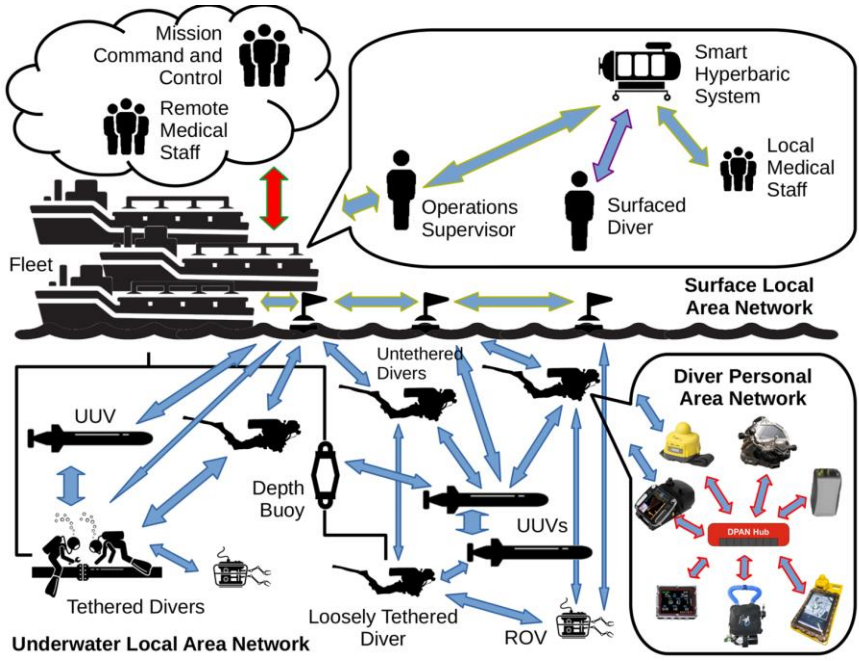

Figure 1: The heterogeneous networking scenario envisioned in the CUIIS project.[23]
The arrow's border color identifies different physical technologies, such as WiFi (in dark yellow), Satellite (in green), Bluetooth™(in purple), Acoustic (in blue), and Proximity RF (in red). Arrow's filling color identifies application protocols. In blue – the UDXP, while in red STANAGs based messaging is employed.

Since AUC is characterized[9] by extremely low bit-rate and exceptionally high packet losses – mainly due to multi-path, interferers, and jammers – a common communication protocol must be designed to maximize its flexibility while minimizing bandwidth requests.

To date, just one international NATO standard has been approved for military underwater communication (JANUS),[10] while many other proprietary solutions have been implemented and successfully deployed in the market).[11, 12, 13] Each implements different technologies and quite peculiar design choices, implying profoundly different specifications, as summarized in Table 1.

A common problem is undoubtedly represented by the constrained size of the payload. Although fragmentation may be a solution, it may also increase the risk of re-transmissions, thus limiting even further the network goodput), [14, 15, 16] in particularly harsh environments.

Another significant limitation in AUC is the latency: to increase the success rate of transmissions, it is quite usual that an aggressive guard time is used to minimize possible collisions.[17] By defining the network usage rate N_u as:

$$N_u = \frac{T_{frame}}{T_{guard} + T_{frame}}$$

Table 1. Summary of Acoustic Physical Layer Features.

	Unit	WSENSE [13]	JANUS[10]	UWIS[11]	RTSYS [12]
Bandwidth	bps	480	80	24	160
Modulation	-	C-QPSK	FH-BFSK	OOK	OFDM [18]
Medium Access	-	Various [21, 19, 20]	CSMA-CA	TDMA	TDMA
Minimum Latency	s	≤ 20	≥ 1.625	2	3
Payload Size	B	50	33	3	40 [20]

it is easy to obtain the slot time T_{slot} reserved for each networked device, which becomes:

$$T_{slot} = \frac{T_{guard}}{1 - N_u}$$

directly influencing λ which is function of the number of networked devices U:

$$\lambda(U) = U \cdot T_{slot}$$

As the network grows, it becomes more and more challenging to realize real-time communication unless T_{guard} can be minimized, i.e., the distance between devices is limited.[20] This suggests that multi-hop communication is of vital importance in underwater communication,[21] thus mandating the need for end-to-end encryption in case message confidentiality has to be guaranteed in a network of shared devices.

On the other hand, in DPAN communication, where wearable devices have limited size and energy storage, the power consumption of network devices is the tightest constraint. Minimization of message exchange is, in this case, the only viable option.

A Publish and Subscribe (P+S) approach, in which publication is performed only in the presence of a limited lifetime subscription, is helpful. To push the envelope of this approach, packet routing through different networks is only performed by means of gateway broker devices. These act as subscribers towards information published in a geographically more limited network and as publishers towards a more geographically extended network. In other words, they store and forward information on behalf of other subscribers. Again, the need for end-to-end security becomes unavoidable if brokers may not be trustworthy devices.

Table 2. The UDXP Packet Format.

	Bit Position							
Bytes	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	SYNCHB							
1	SYNCHN				CLEAR	CYPH ALGO		
1	CAT				TYP			
1	LEN/TNF/LFV							
1	FRAG TYPE		IS LOCAL	CRC PRESENT	SRC LONG	DEST LONG	ADDR	
1	PRIORITY					QOS		
¼	SRC ADDRESS							
0/1/4	DEST ADDRESS							
0/1	FID							
0/1	ECE-ICV							
0/1	EXT CAT							
0/1	EXT TYP							
0/4	SESS ID							
0/1	EVAL							
<256	Payload							
0/2	CRC-16							

The Packet Format

The proposed Underwater Data eXchange Protocol (UDXP) is based on a cross-network packet format acting as an application-level packet for SLAN and ULAN while acting as a cross-layer packet for DPAN. Its format is shown in Table 2. It can be seen that the minimum overhead is limited to 7 bytes while reaching up to 20 bytes for the most complex packet.

The first 1.5 byte encodes a synchronization byte, SYNCHB, that shall always equal $0x44h$. This is followed by a synchronization nibble, SYNCHN, that shall always be set equal to $0x05$.

The CLEAR bit indicates whether the packet is in clear (1) or is ciphered (0). In the latter case, the following byte reports the ciphering algorithm used, as indicated in Table 3.

The CAT nibble, together with the TYP, indicates the category and topic of the message. This allows the recipient to know the payload's content, thus enabling semantic filtering.

Table 3. Admissible Ciphering Algorithms.

Value	Mnemonic	Algorithm
000…011	- -	Reserved
100	AES128	AES-128
101	AES256	AES-256
110	TUBcipher	TUBcipher
111	ChaCha	ChaCha

The following byte can encode either the length of the UDXP packet payload (LEN), the total number of fragments (TNF), or the last fragment valid bits (LFV), depending on the packet type.

In the former case, LEN encodes the length of the payload instead of the length of the remaining packet bytes. UDXP allows payload lengths up to *aMaxPayloadSize* bytes in case of in-clear packets.

In the second case, TNF encodes the total number of fragments of equal length in which the original packet has been divided. The size of each fragment is equal to *aFragPayloadSize*, specified in Table 6. For AES ciphered packets (i.e., those having CYPH ALGO equal to 4 or 5), length is always set equal to 16B in order to convey exactly one single AES block.

In the latter case, this field encodes how many bits within the *aFragPayloadSize* payload are valid for the last fragment. Bits within the payload exceeding LFV shall be discarded at reception and padded with zeros at transmission.

The following field in the UDXP packet allows the specification of the fragment type. Admissible field values can be either $0x0h$, indicating a regular, not fragmented packet; $0x1h$, indicating a fragment; or $0x3h$, in the case of the last fragment.

The IS LOCAL field defines whether the packet can be forwarded to another network or not. This lets gateway brokers know if ULAN packets can make their way up to the SLAN or vice versa. If this field is set to 1, the packet shall not be forwarded to another network.

The CRC PRESENT field indicates whether the packet is closed by a CRC-CCITT calculated 16b long CRC code. When the CRC is present, and a packet is received with a not-matching CRC field value, it must be discarded.

The last two one-bit-long fields, SRC LONG and DEST LONG, specify the length of the SRC ADDRESS and DEST ADDRESS packet fields, respectively. If these flags are set to 1, the respective address field must be *aLongAddrLength* bytes long. This is used in case a short address is neither available nor viable.

The ADDR field specifies packet characteristics: if set to 0, the packet shall be considered a broadcast packet. In this case, no DEST ADDRESS shall be present in the packet. However, SRC ADDRESS must be specified. Should this field be set to 1, the packet shall be considered as unicast, and both source and

destination addresses must be present. A value equal to 2 marks a unicast acknowledgment. This packet is used to assure the sender about the proper reception of a transmission. In case of fragmented messaging, this packet may be sent back in order to monitor correctly the progress of message delivery. A unicast acknowledgment packet shall contain both source and destination addresses. Finally, a value equal to 3 indicates a reliable unicast packet. Such a packet requests an acknowledgment of correct reception.

The PRIORITY field is used to define message delivery scheduling, and it indicates the priority of the message within the same CAT and TYP. A value of 0 indicates the highest priority, and 31 is the lowest.

The QOS field allows to specify the information level conveyed by a message, thus enabling to perform PHY adaptation. This is a trans-codification into another type of message that shall be reverted back into a UDXP message on its reception from compatible PHYs.

The SRC ADDRESS field encodes either the short or the long address of the packet originator. Short addresses are recommended for local links and shall be leased by a local authority in either a static or dynamic way. The second is the preferred choice on local networks in which at least one broker is present. On the other hand, long addresses shall indicate a non-colliding *aLongAddrLength* bytes long identifier. A network-wide authority should grant this address's uniqueness.

The DEST ADDRESS field encodes the destination address, either short or long, to which the packet is directed. UDXP does not reserve any broadcast address: lack of destination address is considered broadcast. Multicast addressing is also allowed, as described in Section 5.

The FID field is useful in identifying a fragment within a fragmented transmission. Up to 256 fragments are allowed, and the FID value must always be limited by the TNF value, encoding the total number of frames. Should a packet be received with FID higher than TNF, it must be discarded.

The ECE-ICV field is present only in ciphered packets in which the used cipher is a block one and cipher chaining is applied.[22] It contains an index to a shared secret used as an initialization vector for a block cipher. Electronic Codebook Enhanced (ECE) Cipher Block Chaining (CBC) is discussed in more depth in Section 6.

Two coupled fields, if needed, may be present: the EXT CAT and EXT TYP. They are used only if both the CAT and TYP fields are set to PROTO ($0x0h$) and EXT_MSG_TYPE ($0xFh$), respectively. These are used to convey 256 message categories, each with 256 types.

In a secured packet exchange, the SESS ID field is used to keep track of the encrypted session and identify the messaging session. UDXP is a connectionless protocol, so there is no negotiation of the SESS ID identifier between the transmitter and receiver. A message exchange is initiated using the lowest SESS ID value, typically generated by the publisher, and continues until the final reply associated with that session is received. If messages do not need a reply, the session is considered to expire after a single message is received.

At every messaging session, a SESS ID value shall be generated by the message originator using a True Random Number Generator (TRNG). The SESS ID shall then be incremented by one at every message exchange by both the messaging session initiator and the receiver. Suppose an error is identified in the SESS ID sequence. In that case, the messaging session must be finished by sending a proper PROTO|SESSION_END packet whose encrypted payload shall contain the initial SESS ID of the session being concluded. In case a packet is received with a SESS ID having a distance less than aInvalidSessIdWindowWidth, it must be discarded by the device that has identified a SESS ID chain error.

An initiator beginning a new messaging session is recommended to select a SESS ID at least *aSessIdMinDistance* far from the initial SESS ID of the preceding messaging session.

Table 4. The Acknowledgement Packet Format.

	Bit Position							
Bytes	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	SYNCHB							
1	SYNCHN				CLEAR	CYPH ALGO		
1	CAT				TYP			
1	0/2							
1	0		0/1	1	0/1	0/1	2	
1	PRIORITY					QOS		
1/4	SRC ADDRESS							
1/4	DEST ADDRESS							
0/1	EXT CAT							
0/1	EXT TYP							
2	CRC-16							

The EVAL field is present in all UDXP packets ciphered with a block cipher, and it is part of their payloads. Since a ciphered packet must have a packet length equal exactly to one ciphering block, it may be necessary to add padding bits to a message that is shorter than the ciphering block. This field – ending up the UDXP packet header – allows, thus, to indicate how many bits in the encrypted payload are valid before they are sent to the upper layer.

Acknowledgment Packet

The UDXP protocol mandates reliable message delivery only in case of fragmented transmission due to the long latency usually experienced by underwater communication. Once a sender has published an acknowledgment needing a message on a given topic, it starts waiting for *aAckWindowLengthTime* milliseconds. Within this time window, sending any other message with the same topic is not allowed. Once that timeout elapses, the sender shall resend the same message for a maximum of *aAckRetxAttempts* times before giving up. Meanwhile, if the receiver receives the packet, it shall reply with an acknowledgment packet with the same topic and with LEN set either to 0 or 2 in case the topic was an extended one.

Table 5. The Fragment Acknowledgement Packet Format.

<table>
<tr><th></th><th colspan="8">Bit Position</th></tr>
<tr><th>Bytes</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>1</td><td colspan="8">SYNCHB</td></tr>
<tr><td>1</td><td colspan="4">SYNCHN</td><td>CLEAR</td><td colspan="3">CYPH ALGO</td></tr>
<tr><td>1</td><td colspan="4">CAT</td><td colspan="4">TYP</td></tr>
<tr><td>1</td><td colspan="8">0/2</td></tr>
<tr><td>1</td><td colspan="2">1/3</td><td>0/1</td><td>1</td><td>0/1</td><td>0/1</td><td colspan="2">2</td></tr>
<tr><td>1</td><td colspan="5">PRIORITY</td><td colspan="3">QOS</td></tr>
<tr><td>1/4</td><td colspan="8">SRC ADDRESS</td></tr>
<tr><td>1/4</td><td colspan="8">DEST ADDRESS</td></tr>
<tr><td>0/1</td><td colspan="8">EXT CAT</td></tr>
<tr><td>0/1</td><td colspan="8">EXT TYP</td></tr>
<tr><td>0/(TNF / 8)</td><td colspan="8">FRAG BITMAP</td></tr>
<tr><td>2</td><td colspan="8">CRC-16</td></tr>
</table>

Packet Fragmentation & Fragment Acknowledgement Packet Format

A fragment acknowledgment strategy has been designed to promote packet flooding with minimal acknowledgment reception. To this extent, each fragment's payload size has been set to either *aMaxFragmentPayloadSize* or *vCypheringBlockLength* bits for in-clear and ciphered packets, respectively. In case the payload is not an integer multiple of those lengths, the last fragment (FRAG TYPE field set equal to 3) is padded with random bits while the number of its valid bits is declared using either the LFV or the EVAL fields, again in case of in-clear and ciphered packets respectively. A sender proceeds to send

fragmented packets – in which instead of the LEN field, the TNF field is used, and where FID must be present – in a best-effort fashion until a fragment acknowledgment packet is received. This packet contains a fragment bitmap in which every bit identifies one of the possible TNF fragments. The total length of the bitmap, thus, is TNF/8 bytes, and it shall be padded, in case of need, with zeroes on the most significant bits.

The UDXP protocol does not mandate the need for fragment acknowledgment. This allows, in quite reliable physical layers, to avoid unnecessary traffic and to minimize power consumption. It should also be noted that for packets marked as fragments (i.e., FRAG TYPE set to 1), the EVAL field shall not be present, while it shall be present in all other ciphered packets.

Table 6. UDXP Constant Values.

Name	Unit	Value
aMaxPayloadSize	B	256
aFragPayloadSize	B	16
aLongAddrLength	B	4
aInvalidSessIdWindowWidth	-	10
aSessIdMinDistance	-	100
aAckWindowLengthTime	ms	1500
aAckRetxAttempts	-	3
aMaxFragmentPayloadSize	b	128
vCypheringBlockLength	b	128
aMulticastAnnouncePeriod	s	30

Categories and Topics

A UDXP network is formed by three types of devices: publishers, subscribers, and brokers. The former is the source of information and can be thought of as a server; the second is the sink of information acting, in other words, as a client; the latter works both as a subscriber towards the publisher it wants to offer re-transmission services to and as a publisher towards another network. Despite being similar to gateways, brokers offer more information aggregation capabilities, providing, e.g., not only store-and-forward services but also caching, aggregation, and traffic management. A future paper will address brokers-enabled services in more detail.

We use ENCAPSULATION—CUDXP to aggregate the same topics coming from different publishers. In this way, a single transmission, if possible, is used to forward multiple publishers' data. A Compressed-UDXP is a payload in which only an SADDR field and payload are included.

To avoid multiple re-transmissions of the same data, publishers usually employ broadcast addressing, but to keep unnecessary traffic inspection under control, they declare the packet's content. The UDXP protocol has been designed for underwater military communication within the Comprehensive Underwater Intervention Intervention System (CUIIS) project [23] context. Its messages categories, thus, encompass the following: protocol PROTO, life support LS, environmental sensors ES, diver monitor DM, dive computer DC, underwater unmanned vehicle UUV, smart hyperbaric system SHS, positioning LOC, communication COMM, hand-held devices HHD, other protocols encapsulation ENCAPSULATION.

Each category defines a set of devices or a given information group. Within each category, a set of 16 different topics are allowed, enabling sensor readings, remote actuation, remote control of unmanned devices, reporting, remote safety and health monitoring, and so on. The following subsections give a very brief description of each category.

Protocol

This category is dedicated to UDXP protocol messaging. It comprises topics allowing the setup of the communication protocol (e.g., assigning to devices network addresses, encryption keys, and more), announcing available topics, managing subscriptions, and soliciting replies.

Life Support

This category is used by those devices that provide life support to human beings. Among them, we can find smart rebreathers, smart life masks, smart Built-In Breathing Systems (BIBS), and more. The available topics here include oxygen, inert, or toxic gas levels, humidity readings, alteration and monitoring of set points, and more.

Environmental Sensors

This category comprises whatever sensor is used to assess the environmental conditions surrounding a human being, e.g., hazardous gases or conditions (temperature, humidity, pressure), presence of inert gases, and more. Due to its flexibility, this category also comprises topics that describe the sensor and provide diagnostics.

Diver Monitor

This category allows the encoding of messages about diver health status. It comprises information such as blood pressure, breathing frequency, heart rate, skin temperature, inhaled oxygen and inert gases, partial pressures, carbon dioxide level, and more.

Dive Computer

Due to the relatively large amount of different information conveyed by a dive computer and the actuation (i.e., settings) allowed, a category has been

dedicated to this vital safety instrument. It allows for the reading of ambient pressure, depth, water temperature, information about oxygen levels, ascent and descent speed, inhaled mix, and oxygen partial pressure. It also encodes alerts and alarms, and it allows the modification of decompression calculation parameters.

Unmanned Underwater Vehicle

This category allows retrieving information and maneuvering UUVs from the surface and other underwater devices, such as the dive computer or the hand-held device. UUV conditions, heading, speed, way-points, and more can be monitored in real-time by means of these messages, while, in the same way, complex orders can be sent to a UUV. Due to bandwidth limitations, it is, in fact, not possible to remote control a UUV in real time. Still, it is possible to interact with it by issuing commands ordering it to perform complex maneuvers.

Smart Hyperbaric System

This message category conveys critical information about hyperbaric chambers used to perform surface decompression and treat injured divers. Using these messages, it is possible to monitor the SHS conditions, such as oxygen levels, temperature, and humidity, and to set proper treatment or ascent tables.

Positioning

This category allows to inform devices, usually in the DPAN, about their position both on the surface and underwater. Positioning is usually calculated using information from both GNSS/GPS and acoustic buoys, thus allowing the three-dimensional positioning of divers and UUVs. Messages conveyed by this category encompass GPS coordinates, depth, speed, heading, UTC time, and alarms triggered by geofencing or by teammates' positions.

Communication

This category includes messages devoted to communication between command, control, and divers or between divers. Due to the difficulty of inputting data in some of the divers' usually equipped devices (e.g., dive computers) and the need to allow a certain degree of confidentiality in messages, two categories of messages have been envisioned: preset and text-based messages. The former are usually indexes of a common small dictionary, allowing minimal but efficient communication. Messages of this type could encode "OK", "STOP", "SURFACE", and so on. Textual messages, on the other hand, are input by the user and are usually length-limited. Alarms can also be conveyed.

Hand-Held Devices

This message category encompasses peculiar messages that can be published only by HHDs. Among them are orientation, bitmaps, alerts, actuation commands, and more.

Encapsulation

This category allows the sending of messages whose content is other messages. It becomes particularly interesting when brokers need to aggregate information coming from different publishers. In this case, either compressed UDXP or UDXP packets can be encapsulated. Other protocols, such as UDP or IP, can also be embedded within a UDXP packet. This may allow easier coexistence in heterogeneous protocol networks.

Multicast Groups

Aiming to limit the number of transmissions to different devices within a diving team, UDXP allows the definition of multicast groups by using multicast addresses. These are long addresses within the range starting from $0x9B000000h$ up to $0x9BFFFFFF$. Devices are free to join and leave, on demand, multicast groups.

Publishers may either be instructed or discover the presence of multicast groups through PROTO messages. A device willing to join or create a multicast group will periodically start sending out a JOIN_MCAST command broadcasted to the network. If the device does not receive any reply, it will try again later, after an *aMulticastAnnouncePeriod* long time interval. Otherwise, if it receives a reply, the group will be considered active, and the publisher will consider itself allowed to send out packets addressed to that multicast address. Every device participating in a multicast group is responsible for keeping track of the number of joined devices in that group. As soon as enough LEAVE_MCAST messages have been collected, the multicast group is to be considered closed, thus denying the chance to use its address as a destination for any other UDXP packet. In administered networks, it is, of course, also possible to instruct UDXP devices to use a multicast group in a static way, i.e., without active announcement and discovery via PROTO messages.

Electronic Codebook Enhanced CBC

The CBC enhancement proposed in this work allows to avoid as much as possible repetitions of the same ciphertext when the plaintext is known. UDXP packets can be extremely short and predictable: e.g., the oxygen setpoint of most divers will be between 1.1 and 1.3 with 0.1 increments. Together with the payload content declaration and due to the extremely high probability of packet loss – preventing the chance to aggressively use long sessions in which chaining itself helps to avoid a known plaintext attack – this could lead, at least in simplified AES versions, to an increased risk of cryptoanalysis.[24] In UDXP a shared secret is used as Initialization Vector (IV) for supported block ciphers. Each device is instructed about the Shared Secret (SS) via a secured channel (i.e., not over the UDXP network) right after initialization. As soon as the SS is known, it is used by both the initiator and receiver to start a messaging session whose entropy is increased by adding a 4-byte-long random session identifier (SESS-ID) to the packet payload.

The ECE-CBC approach is shown in Figure 2. Here, the initiator starts selecting a random value for IV 1 and uses that value as a seed to generate a SESS-ID. The UDXP packet (shown in its encrypted version in Table 7) is then sent to the

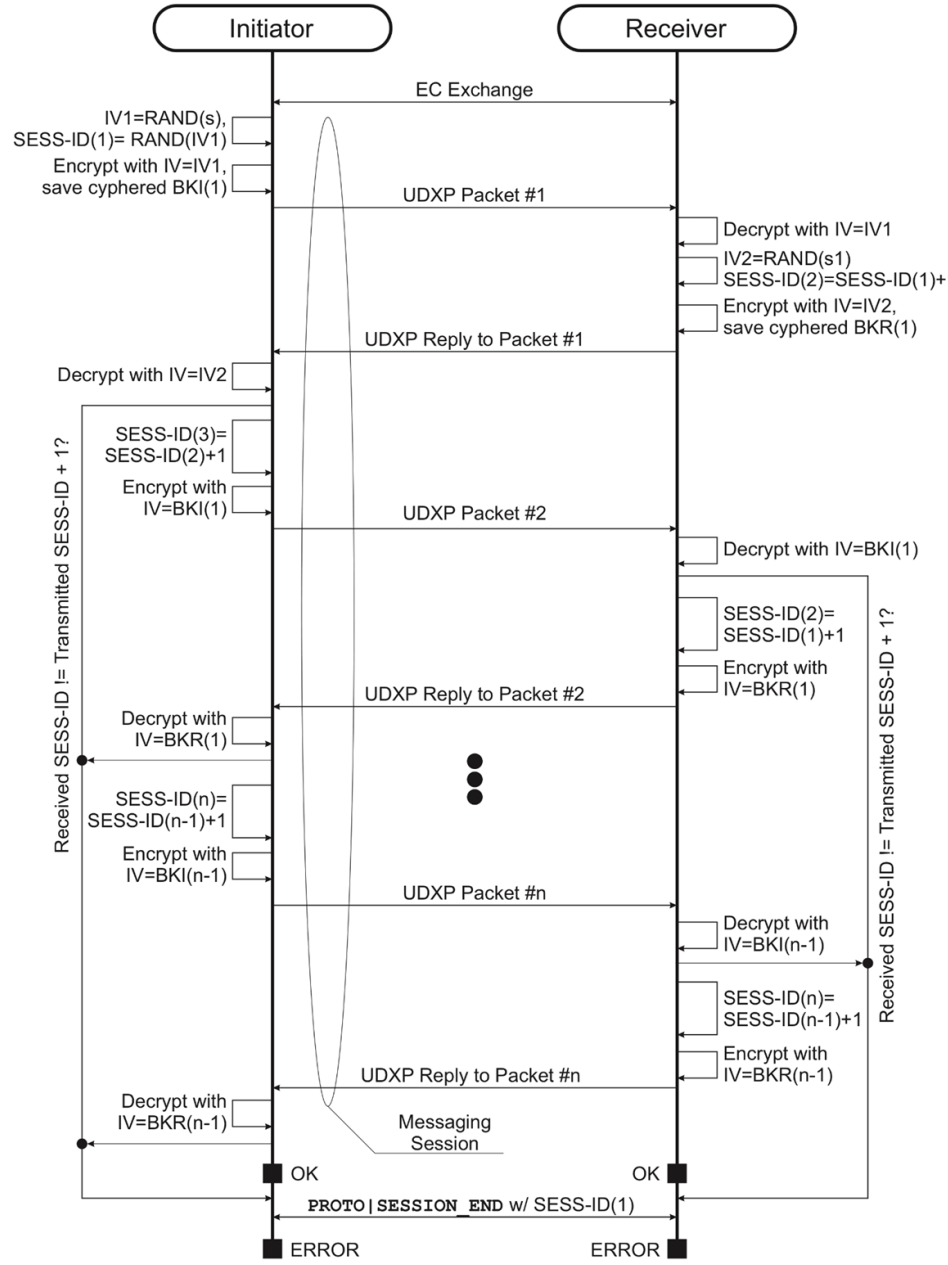

Figure 2: ECE-CBC session.

receiver. This one will access the SS at IV 1 offset and then retrieve the initialization vector to decipher the packet correctly.

Table 7. The secured UDXP packet format. In the table encrypted fields are indicated with underlined font, while CBL = *vCypheringBlockLength*.

	Bit Position							
Bytes	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	SYNCHB							
1	SYNCHN				0	CYPH ALGO		
1	CAT				TYP			
1	LEN/TNF/LFV							
1	FRAG TYPE		IS LOCAL	CRC PRESENT	SRC LONG	DEST LONG	ADDR	
1	PRIORITY					QOS		
1/4	SRC ADDRESS							
1/4	DEST ADDRESS							
1	ECE-ICV							
0/1	EXT CAT							
0/1	EXT TYP							
4	SESS ID							
1	EVAL							
CBL	Payload							
0/2	CRC-16							

At this time, the receiver will select a random IV 2, and the SESS-ID will be incremented by one to proceed with the messaging session. The reply will be sent to the initiator, ciphering it with the initialization vector value and obtaining access to the SS at the IV 2 offset.

The communication will then go on in CBC mode, using the previous block as IV for the next one on both ends of the communication. As soon as the session concludes, either due to a lack of further messages or an error, a PROTO message is sent to the other party indicating SESSION_END.

The situation becomes way easier when no reply is needed. This is shown in Figure 3. In this case, the initiator always selects a new IV X value and a new

SESS-ID after every UDXP packet sent to the receiver. The message receiver has to use the value obtained from the ECE-ICV field to enter the SS to obtain the IV value.

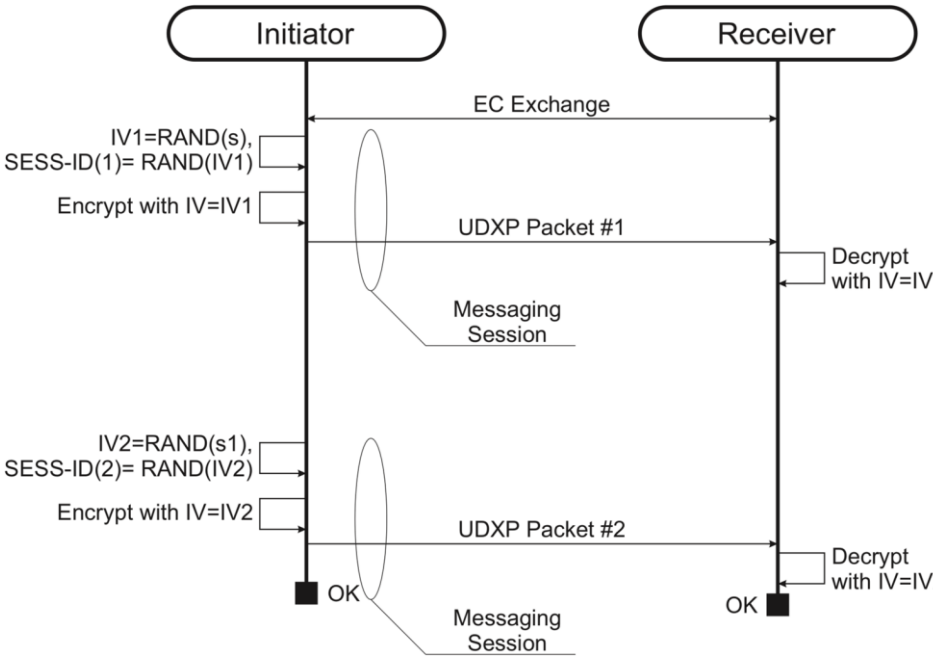

Figure 3: A single message ECE-CBC session.

Conclusions

This paper introduced UDXP, a cross-layer underwater application-driven communication protocol aiming to ease interoperability between different systems in harmonized underwater personal and local area networks. The UDXP protocol can be encapsulated within a UDP/IP packet and used as an application-level message also on surface networks until a more suitable encoding is applied to conform with existing military and civil standards, such as IEEE P1451.0[25] or STANAGs.

Despite this being preliminary work, we believe UDXP is a promising protocol that brings advanced networking concepts to a complex and complete system architecture for successful underwater communication. With this protocol, secure and trustful communication is enabled while ensuring enough flexibility to be implemented in the most resource- and bandwidth-constrained devices and networks. Military-grade secure communication can thus be guaranteed from the surface down into the depths of oceans.

Acknowledgement

This work has been developed within a project that has received funding from the European Defence Industrial Development Programme (EDIDP) under grant agreement No "EDIDP-UCCRS-EDD-2020-059 — CUIIS." This paper reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information this paper contains.

References

[1] Robert Frank Codd-Downey and Michael Jenkin, "Lightbyte: Communicating Wirelessly with an Underwater Robot Using Light," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics,* vol. 2: ICINCO, INSTICC, SciTePress, 2018, pp. 299–306.

[2] Xianhui Che, Ian Wells, Gordon Dickers, and Paul Kear, "TDMA Frame Design for a Prototype Underwater RF Communication Network," *Ad Hoc Networks* 10, no. 3 (2012): 317–327, https://www.sciencedirect.com/science/article/pii/S1570870511001521.

[3] Ishtiaq Ahmad and Kyung Hi Chang, "Downlink Power Allocation Strategy for Next-Generation Underwater Acoustic Communications Networks," *Electronics* 8, no. 11, (2019), https://www.mdpi.com/2079-9292/8/11/1297.

[4] Bo-Min Seo, Junho Cho, and Ho-Shin Cho, "A Signaling-free Underwater Code Division Multiple Access Scheme," *Electronic* 8, no. 8 (2019), https://www.mdpi.com/2079-9292/8/8/880.

[5] Jun Zhang, Zhi Hu, Yan Xiong, and Gengxin Ning, "A Collision-free Hybrid Mac Protocol Based on Pipeline Parallel Transmission for Distributed Multi-channel Underwater Acoustic Networks," *Electronics* 9, no. 4 (2020), https://www.mdpi.com/2079-9292/9/4/679.

[6] Shahzad Ashraf, Zeeshan Aslam, Adnan Yahya, and Adnan Tahir, "Underwater Routing Protocols: Analysis of Link Selection Challenges," *AIMS Electronics and Electrical Engineering* 4, no. 3 (2020): 234-2484.

[7] Taj Rahman, Irfan Ahmad, Asim Zeb, Inayat Khan, Gauhar Ali, and Mohammed ElAffendi, "Performance Evaluation of Routing Protocols for Underwater Wireless Sensor Networks," *Journal of Marine Science and Engineering* 11, no. 1, (2023), https://www.mdpi.com/2077-1312/11/1/38.

[8] Mandar Chitre, Shiraz Shahabudeen, Lee Freitag, and Milica Stojanovic, "Recent Advances in Underwater Acoustic Communications and Networking," C*onference: OCEANS 2008,* Volume: 2008-Supplement, October 2008, pp. 1–10.

[9] Dario Pompili and Ian F. Akyildiz, "A Multimedia Cross-layer Protocol for Underwater Acoustic Sensor Networks," *IEEE Transactions on Wireless Communications* 9, no. 9, (2010): 2924–2933.

[10] John Robert Potter, Joao Alves, Dale Green, Giovanni Zappa, Ivor Nissen, and Kim McCoy, "The Janus Underwater Communications Standard," in *2014 Underwater Communications and Networking (UComms) Conference*, *Sestri Levante*, 2014, pp. 1–4.

[11] UWIS, "Underwater Navigation, Communications and Surveillance for Divers," 2023, https://uwis.fi/en/underwater-navigation-system-underwater-communications-system.

[12] RTSYS, "Enhancing Trust Underwater," 2023, https://rtsys.eu.

[13] WSENSE, "Internet of Underwater Things, W-node," 2023, https://wsense.it/wsense-iout-platform/wnode.

[14] Rolando Herrero, "Hybrid Error Correction in Fragmented IOT Media Streams," *Transactions on Emerging Telecommunications Technologies* 33, no. 11 (2022): e4601, https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4601.

[15] Ioana Suciu, Xavier Vilajosana, and Ferran Adelantado, "An Analysis of Packet Fragmentation Impact in LPWAN," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[16] Ioana Suciu, Xavier Vilajosana, and Ferran Adelantado, "Aggressive Fragmentation Strategy for Enhanced Network Performance in Dense LPWANs," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE*, September 2018, https://doi.org/10.1109/pimrc.2018.8581051.

[17] Camila M. G. Gussen, Christophe Laot, François-Xavier Socheleau, Benoît Zerr, Thomas Le Mézo, Raphaël Bourdon, and Céline Le Berre, "Optimization of Acoustic Communication Links for a Swarm of AUVs: The COMET and NEMOSENS Examples," *Applied Sciences* 11, no. 17 (2021), https://www.mdpi.com/2076-3417/11/17/8200.

[18] Arnaud Bourre, Said Lmai, Christophe Laot, and Sébastien Houcke, "A robust OFDM Modem for Underwater Acoustic Communications," in *Oceans 2013: MTS/IEEE conference, Bergen, Norway*, Jun. 2013, pp. 1 – 5, https://hal.science/hal-00935199.

[19] Roberto Petroccia, "Underwater Wireless Sensor Networks," 2013, https://twiki.di.uniroma1.it/pub/Wireless/WebHome/UWSNs_lezione2013.pdf.

[20] Nitthita Chirdchoo, Wee-Seng Soh, and Kee Chaing Chua, "Aloha-Based MAC Protocols with Collision Avoidance for Underwater Acoustic Networks," in *Proceedings of the IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 2271–2275.

[21] Nirvana Meratnia, Paul J.M. Havinga, Paolo Casari, Chiara Petrioli, Knut Grythe, Thor Husoy, and Michele Zorzi, "CLAM — Collaborative Embedded Networks for Submarine Surveillance: An Overview," in *OCEANS 2011 IEEE - Spain*, 2011, pp. 1–4.

[22] Morris Dworkin, "Recommendation for block cipher modes of operation: Methods and techniques," NIST, SP 800-38A, December 2001, https://csrc.nist.gov/publications/detail/sp/800-38a/final.

[23] CUIIS Consortium, "CUIIS Project Grant Agreement EDIDP-UCCRS-EDD-2020-059-CUIIS," EU, 2020, https://cuiis.eu.

[24] Charles Bouillaguet, Patrick Derbez, Orr Dunkelman, Nathan Keller, Vincent Rijmen, and Pierre-Alain Fouque, "Low-data Complexity Attacks on AES," *IEEE Transactions on Information Theory* 58, no. 11 (2012): 7002–7017.

[25] IEEE, "ISO/IEC/IEEE Information technology – Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats," IEEE P1450.0.D4:2023, March 2023.