Carl von Ossietzky Universität Oldenburg

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

# Self-Organizing Ad-Hoc Routing for maritime Multi-Domain Systems

Von der Fakultät für Informatik, Wirtschafts- und Rechtswissenschaften der Carl von Ossietzky Universität Oldenburg zur Erlangung des Grades und Titels

**Doktor der Ingenieurwissenschaften (Dr.-Ing.)**

angenommene Dissertation

von Herrn Konstantin Johannes Brandl

geboren am 12.03.1997 in Weilburg (Deutschland)

Erstgutachter:

Prof. Dr.-Ing. Axel Hahn
Carl von Ossietzky Universität Oldenburg

Zweitgutachter:

Prof. Dr.-Ing. habil. Frank Sill Torres
Universität Bremen

Tag der Disputation: 20.02.2026

# Abstract

Protecting critical maritime infrastructure (CMI) in remote offshore regions poses a significant challenge due to limited communication coverage and harsh environmental conditions. Offshore wind farms and submarine cables are vulnerable to threats from above, on the sea surface, and below the water. Reliable protection requires the integration of Multi-Domain unmanned platforms that can coordinate across these environments via a Mobile Ad-Hoc Network (MANET), which must address the challenges arising from the transition between radio-based and acoustic communication domains by utilizing intermediate gateway nodes. This work extends the Ad-Hoc On-Demand Distance Vector (AODV) routing protocol to enable Multi-Domain Operations with unmanned systems operating in the air, on the sea surface, and underwater. The extended AODV protocol combines internal, geolocation-based logic with external, signal quality-based logic. Using additional message types, such as *Relay Request* and *Relay Reply*, nodes can dynamically identify and select Unmanned Surface Vehicles (USVs) as gateways. As USVs support radio communication above-water and acoustic communication underwater, they facilitate direct Cross-Domain data transmission and routing between the two environments. For evaluation, the classical AODV protocol is first implemented in a time-discrete simulation environment in MATLAB and adapted into a hybrid version that reflects the specific characteristics of radio and acoustic communication. Using the hybrid AODV as a reference baseline, the performance of the extended AODV is systematically compared within the simulation framework. Concurrently, a practical demonstrator is being developed using Raspberry Pis in conjunction with acoustic modems. The measurements obtained from the demonstrator include transmission range, message runtime, signal-to-noise ratio, Reception Drop Rate and received data volume. These measurements are used to parameterize the simulation's physical models.

The results show that the extended AODV protocol decreases latency, increases reliability, and decreases data traffic in the hydroacoustic channel. Using USVs as gateways significantly improves communication between domains, providing a reliable link between UAVs in the air and UUVs underwater. Therefore, this work advances the development of resilient, Cross-Domain Ad-Hoc communication systems and supports the protection of critical maritime infrastructure.

# Zusammenfassung

Der Schutz maritimer kritischer Infrastrukturen in abgelegenen Offshore-Regionen ist aufgrund begrenzter Kommunikationsabdeckung und rauer Umweltbedingungen besonders anspruchsvoll. Offshore-Windparks und Unterseekabel sind Bedrohungen aus der Luft, von der Oberfläche und aus dem Wasser ausgesetzt. Ein zuverlässiger Schutz erfordert die Koordination multidomainfähiger, unbemannter Plattformen über ein mobiles Ad-Hoc-Netzwerk (MANET). Dieses überbrückt mithilfe von Gateway-Knoten den Übergang zwischen funkbasierter und akustischer Kommunikation. In dieser Arbeit wird das Ad-Hoc On-Demand Distance Vector (AODV) Routing-Protokoll erweitert, um eine domänenübergreifende Vernetzung mit unbemannten Systemen zu ermöglichen. Das erweiterte AODV Protokoll kombiniert eine interne, geolokalisierungsbasierte Logik mit einer externen, signalqualitätsbasierten Logik. Mithilfe zusätzlicher Nachrichtentypen wie *Relay Request* und *Relay Reply* können Netzwerkknoten unbemannte Oberflächenfahrzeuge (USVs) dynamisch identifizieren und als Gateways auswählen. USVs fungieren als Gateways zwischen Funk- und akustischer Kommunikation und ermöglichen eine domänenübergreifende Datenübertragung. Zur Bewertung wurde zunächst das klassische AODV Protokoll in einer zeitdiskreten Simulationsumgebung in MATLAB implementiert und an eine Hybridversion angepasst, welche die spezifischen Eigenschaften der Funk- und Akustikkommunikation widerspiegelt. Unter Verwendung des hybriden AODV als Referenzbasis wird die Leistung des erweiterten AODV Protokolls innerhalb des Simulationsrahmens systematisch verglichen. Parallel dazu wird ein praktischer Demonstrator entwickelt. Mit dem Demonstrator werden Messungen zur Übertragungsreichweite, Nachrichtenlaufzeit, zum Signal-Rausch-Verhältnis, zur Empfangsausfallrate und zum empfangenen Datenvolumen durchgeführt. Diese Messungen werden zur Parametrisierung der physikalischen Modelle der Simulation verwendet.

Die Ergebnisse zeigen, dass das erweiterte AODV Protokoll die Latenz verringert, die Zuverlässigkeit erhöht und den Datenverkehr im hydroakustischen Kanal reduziert. Die Verwendung von USVs als Gateways verbessert die Kommunikation zwischen den Domänen erheblich und stellt eine zuverlässige Verbindung zwischen UAVs in der Luft und UUVs unter Wasser her. Somit trägt diese Arbeit zu der Entwicklung robuster, domänenübergreifender Ad-Hoc-Kommunikationssysteme bei und unterstützt damit den Schutz maritimer kritischer Infrastruktur.

# Acknowledgement

I would like to express my sincere gratitude to my PhD supervisor, Prof. Dr.-Ing. Axel Hahn, for his continuous guidance, constructive feedback and academic freedom throughout my studies. His expertise and trust significantly shaped the direction of this work and contributed decisively to its successful completion.

I am equally grateful to my second examiner, Prof. Dr.-Ing. Frank Sill Torres, for taking the time to review this work and for providing valuable feedback that has helped to enhance its quality.

My sincere thanks also go to Dr.-Ing. Jeronimo Dzaack and Dr.-Ing. Julien Hansen, who gave me the exceptional opportunity to undertake my PhD as one of the first candidates at the A-LAB. Their trust and support, which began during my master's thesis, were instrumental in enabling this work and supporting my academic development.

I would like to thank the entire A-LAB team for their professional collaboration, engaging technical discussions and supportive working environment throughout my doctoral studies. I would particularly like to thank the students I supervised for their initiative and independent approach, which repeatedly inspired new ideas and perspectives.

I am deeply grateful to my parents for their constant support and encouragement throughout my studies. Their belief in me has been essential in setting me on the right academic path and helping me to successfully complete this PhD.

Finally, I would like to thank my partner, Helene Dreißigacker, for her continuous support throughout my academic journey. Her encouragement and her ability to motivate me to believe in our shared goals have given me strength and confidence, especially during challenging periods.

# Table of Contents

# List of Abbreviations

| **Abbreviations** | |
|---|---|
| AI | Artificial Intelligence |
| AODV | Ad-Hoc On-Demand Distance Vector |
| BER | Bit Error Rate |
| BLOS | Beyond-Line-of-Sight |
| CMI | Critical Maritime Infrastructure |
| CPS | Cyber-Physical System |
| FR | Functional Requirement |
| IoT | Internet of Things |
| LoS | Line-of-Sight |
| MCACHE | Message Cache |
| MDO | Multi-Domain Operations |
| NFR | Non-Functional Requirement |
| OSI | Open System Interconection |
| RDR | Reception Drop Rate |
| RelRep | Relay Reply |
| RelReq | Relay Request |
| RET | Route-Establishment-Time |
| RREP | Route Reply |
| RREQ | Route Request |
| RQ | Research Question |
| RTABLE | Routing table |
| RTT | Round-Trip Time |
| SNR | Signal-to-noise-ratio |
| TTL | Time-To-Live |
| UAV | Unmanned Aerial Vehicle |
| UDP | User Datagram Protocol |
| UUV | Unmanned Underwater Vehicle |
| UxV | Unmanned Vehicle |

# List of Figures

# List of Tables

# 1 Introduction

Ensuring reliable communication across aerial, surface and underwater domains remains a major challenge in autonomous maritime operations. This project aims to develop a routing algorithm that will enable autonomous, Cross-Domain systems to interconnect and cooperate within dynamic maritime environments. This communication capability is essential for advanced applications, including safeguarding critical maritime infrastructure. The protection of the Borkum Riffgrund 2 offshore wind farm is used as a representative example to demonstrate the practical relevance of the proposed approach. Based on this scenario, communication objectives are derived that reveal existing capability gaps limiting Cross-Domain network interoperability. These findings provide the basis for deriving functional and non-functional requirements, revealing the key network-layer challenges that motivate the development of a dynamic Multi-Domain Ad-Hoc routing mechanism. From this, the central research question is formulated, highlighting the need for a Cross-Domain routing approach. Finally, the chapter outlines the scientific contribution of this work.

## 1.1 Motivation and Operational Context

The growing importance of critical maritime infrastructures (CMIs), including submarine cables, pipelines and offshore wind farms, is underlined by their essential role in energy supply, communications and economic stability. Geopolitical developments and several sabotage events in recent years have drawn increased attention to the security risks associated with these systems and demonstrated the necessity of more resilient protection measures (Bueger & Liebetrau, 2023). Indeed, the global communication infrastructure is dependent on a network of submarine cables spanning approximately 1.4 million kilometers (Clare, 2023). This network is responsible for the transmission of over 99% of internet traffic (Tréhu & Roberts, 2024). The disruption of this network would have a significant impact on the global internet infrastructure, potentially leading to its collapse (Ganz et al., 2024). Recent geopolitical incidents have highlighted the vulnerability of critical maritime infrastructure. The sabotage of the Nord Stream 1 and 2 gas pipelines in September 2022 showed that subsea infrastructure can be deliberately targeted to gain geopolitical leverage and cause economic disruption (Q. Zhang et al., 2024). This incident

also highlighted the importance of ensuring the security of Europe's underwater infrastructure (Bueger et al., 2022). A similar event occurred in October 2023, when the Balticconnector pipeline, which runs between Finland and Estonia, and nearby communication cables were severely damaged in an incident suspected to be the result of external influence (Cannon & Bhatt, 2024). These cases demonstrate that subsea assets, including energy pipelines, power cables, and offshore communication links, are vulnerable to both intentional attacks and accidental damage. Due to their remote location and limited real-time monitoring, maritime infrastructure has become a strategically sensitive target (Schaller, 2024), which reinforces the need for autonomous monitoring and resilient communication systems in this domain.

The primary challenge in protecting critical maritime infrastructures stems from their vast size and remote location. For instance, the Borkum Riffgrund 2 offshore wind farm covers an area of around 36 km² and is largely unmanned, which limits the possibility of continuous human supervision. Furthermore, offshore wind energy has become a key pillar of Germany’s energy supply, rendering such infrastructure strategically critical. In 2022, offshore wind farms contributed around 8.1 GW to the national grid, accounting for approximately 4% of Germany’s gross electricity production. National energy strategies aim to increase offshore capacity to at least 70 GW by 2045, highlighting the increasing importance of ensuring the security and resilience of offshore operations (Sill Torres, 2025). Unlike terrestrial infrastructures, which can rely on fixed communication infrastructure for security and monitoring, maritime environments require persistent situational awareness across both the surface and underwater domains. Unmanned Vehicles (UxVs), including Unmanned Aerial Vehicles (UAVs), Unmanned Surface Vehicles (USVs), and Unmanned Underwater Vehicles (UUVs), offer a flexible and scalable means of distributed monitoring. Combining different types of sensors, such as cameras on UAVs, radar on USVs and sonar on UUVs, allows UxVs to work together to improve situational awareness in difficult maritime environments, as illustrated in Figure 1. The combination of various sensors, including cameras on UAVs, radar on USVs and sonar on UUVs, enables UxVs to collaborate and enhance situational awareness in challenging maritime conditions. These platforms can be classified as cyber-physical systems (CPS) because they integrate sensing, actuation, software and communication technologies in order to interact with the physical environment in real time (Brinkmann & Hahn, 2017). While the CPS nature of UxVs enables autonomous decision-making, it also increases system complexity and vulnerability, particularly in distributed maritime operations.

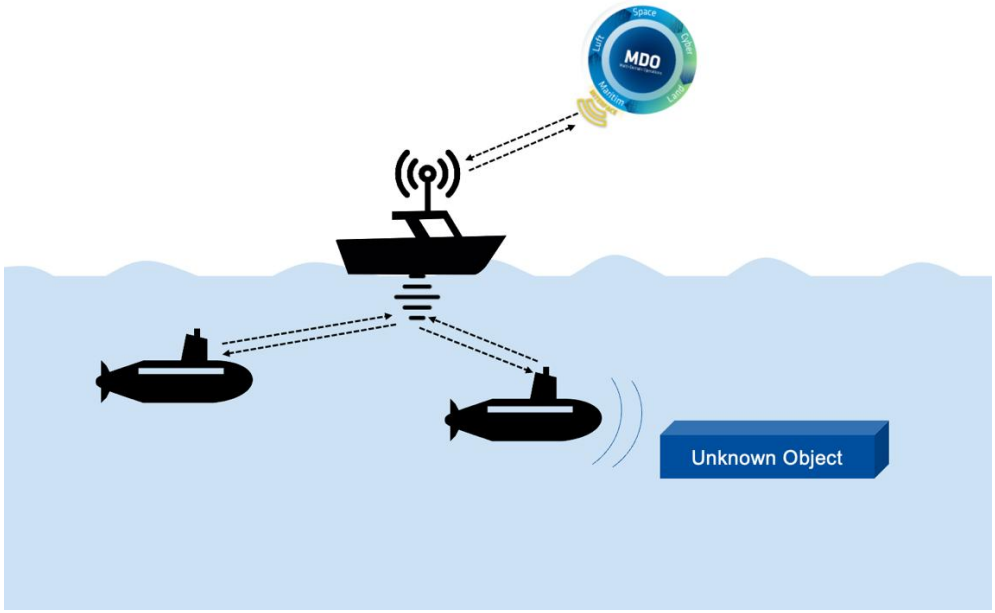


Figure 1: Example of an MDO scenario: Two UAVs are connected via a gateway (USV) to two flying drones and a central MDO cloud. In order to accomplish the search mission for a container accident, the assets must self-organize in a swarm.

The utilization of such vehicles provides operational flexibility, especially in areas where traditional communication infrastructures are limited (Orfanidis et al., 2019). The integration of UAVs, USVs, and UUVs in a coordinated manner has been shown to enhance the effectiveness of sensor capabilities (Ding et al., 2024). Integrating optical imaging from UAVs, radar data from USVs, and sonar data from UUVs can generate a comprehensive maritime situational picture across domains (Orfanidis et al., 2019). However, enabling coordinated Multi-Domain Operation (MDO) is essentially a networking issue. Communication between UxVs must function under highly dynamic, infrastructure-free conditions, since no fixed base stations, satellites or backbone networks can be relied upon to provide consistent connectivity in offshore regions. The network topology changes over time due to mobility, diving and surfacing maneuvers, and temporary communication losses between nodes. In addition, UxVs operate across two heterogeneous physical communication domains. The first is radio frequency (RF) communication in the air and on the surface. The second is hydroacoustic communication underwater. The bandwidth, propagation speed, range and error characteristics of these media differ significantly, leading to asymmetric link behavior and frequent link disruption. Traditional network architectures based on centralized control or static routing are infeasible under these conditions. Instead, communication must rely on autonomous, self-organizing networking mechanisms that can establish and maintain routes in real time, even when there

is mobility or intermittent connectivity. Distributed coordination and adaptive multi-hop forwarding are required to maintain end-to-end communication reliability without infrastructure support. Gateway nodes are necessary to enable interoperability across communication domains by bridging RF and hydroacoustic links while preserving addressing continuity and routing information. In network theory, such systems are categorized as infrastructure-less, self-organizing mobile networks characterized by dynamic topologies, heterogeneous link conditions, and an absence of centralized coordination (Tanenbaum & Wetherall, 2011). These characteristics introduce substantial routing complexity, particularly in hybrid communication environments, where conventional protocols are not designed to cope with heterogeneous propagation delays, limited underwater bandwidth, and high routing overhead. In such dynamic environments, point-to-point communication is insufficient, as communication links cannot be assumed to persist over time due to node mobility, intermittent connectivity, and heterogeneous transmission media. Therefore, routing is critical to enabling scalable, self-organizing communication. Any routing strategy must consider the significant differences in delay and reliability between radio and hydroacoustic links, in order to prevent route failures caused by timeouts or outdated control information. Furthermore, routing decisions must consider link quality, transmission range, and spatial node distribution to maintain robust connectivity over multiple hops. As surface vehicles often act as dynamic gateway nodes between aerial and underwater units, they must be automatically detected and integrated into routing procedures to ensure seamless Cross-Domain communication.

Although routing concepts are developed in this work, they must first be validated in a controlled and reproducible environment. While physical testbeds provide valuable realism for maritime research (Hahn, 2014), their scalability is limited when it comes to evaluating large, dynamic, Multi-Domain Networks. Therefore, a dedicated network simulation framework is needed. This framework must enable the time-discrete modeling of hydroacoustic propagation, radio communication, and routing behavior across all vehicle types involved (S. Peng et al., 2020). It is unclear whether the current technological capabilities are adequate for the specific communication requirements necessary for effectively protecting maritime critical infrastructures with UxVs. In this context, this work aims to derive both operational and functional requirements for a suitable communication solution, based on a representative real-world scenario. Subsequently, the State of the Art in routing protocols for Mobile Ad-Hoc Networks will be examined in detail, with a particular focus on identifying current technological limitations in the context of maritime

Multi-Domain communication. This work uses the Borkum Riffgrund 2 offshore wind farm as a case study to examine how unmanned aerial, surface and underwater systems can be integrated to protect critical maritime infrastructure. The resulting communication requirements are analyzed and existing technologies are assessed. The need for enhanced routing mechanisms in Multi-Domain Networks is also identified. Five routing concepts are developed and evaluated against state-of-the-art approaches, as well as against the system requirements derived from the gap analysis.
To evaluate their performance, a time-discrete network simulation framework was implemented in MATLAB and parameterized using measurements from a physical demonstrator comprising Raspberry Pis, acoustic modems, and UAVs. The simulation reflects realistic link behavior and is applied to representative use cases derived from the case study. The routing mechanisms are evaluated based on performance metrics in order to analyze their behavior under different operational conditions.

## 1.2 Use Case: Protection of the Offshore Windpark Borkum Riffgrund 2

This work explores a technical solution for safeguarding critical offshore infrastructure, using the Borkum Riffgrund 2 wind farm as a case study. The communication objectives derived from this operational scenario identify the specific networking requirements necessary for reliable, domain-spanning coordination between unmanned platforms. The Borkum Riffgrund 2 offshore wind farm is located in the German sector of the North Sea, approximately 57 kilometers northwest of the island of Borkum and around 45 kilometers from the German mainland coast. The wind farm which is shown in Figure 2 is operated by the energy company Ørsted and is part of Germany’s renewable energy infrastructure, contributing significantly to the national energy supply. The site comprises 56 wind turbines, each with an 8-megawatt rated capacity, resulting in a total capacity of 450 megawatts. Spanning a surface area of approximately 36 square kilometers, the wind farm operates in water depths ranging from -25 to -30 m. Access to the site is limited to vessels and helicopters, which depend heavily on weather and sea conditions. (IWR, 2017)

Figure 2: Depiction of the Borkum Riffgrund 2 wind farm (Ørsted, image gallery, accessed 06.12.2025)

According to the *Bundesamt für Sicherheit der Informationstechnik* (BSI), offshore wind farms such as Borkum Riffgrund 2 are considered part of the critical infrastructure in the energy sector as they have a net installed capacity of over 104 MW. They are therefore subject to special security and monitoring requirements, legal obligations, including the implementation of adequate security measures, and regular audits (BSI-KritisV, 2016; Sill Torres, 2025). Due to its remote location and strategic importance, the Borkum Riffgrund 2 offshore wind farm is exposed to a number of potential risks. These include unauthorized intrusion by ships, aerial observation by drones and underwater activities that could target submarine cables or foundation structures. Continuous monitoring of all relevant air, water surface and underwater domains is required in order to be able to react to such hazards at an early stage. Traditional, manned surveillance concepts quickly reach their limits in this environment. A continuous presence of personnel on site is almost impossible to realize due to logistical, weather-related and economic restrictions. In addition, manned operations are limited in their range and reaction speed, especially in the case of parallel or difficult to recognize threat scenarios. Therefore, the aim of this study is to investigate the deployment of UxVs to provide persistent monitoring, rapid response capabilities, and domain-spanning coverage, regardless of human presence. UAVs enable the rapid capture of large-scale aerial images and early detection from a height. USVs such as those shown in Figure 3 perform patrol functions and can serve as Relay nodes for communication data.


Figure 3: Picture of the USV “Moritz” in summer 2024. The USV is a proprietary design with extensive sensor technology and communication (radio & hydroacoustic), (photograph by the author).

UUVs enable continuous monitoring below the water surface, especially along critical infrastructure such as submarine cables. Coordinated cooperation between these platforms can create a Cross-Domain situational picture, despite permanent human control.


Figure 4: Image of the SeaCat UUV. This is a proprietary design by TKMS Atlas and is equipped with extensive sensor technology such as radar and communication systems for above-water and below-water operation (TKMS ATLAS ELEKTRONIK GmbH, SeaCat MK1, accessed 06.12.2025)

This is also known as emergent self-organization of Multi-Domain Operations and describes a highly complex system of systems approach (Brandl et al., 2024). The development of a hybrid, Cross-Domain communication protocol represents the core of this work. To achieve this self-organizing behavior, participating systems must exchange operationally relevant data, such as geolocation, mission status, sensor observations, role

assignments and link quality indicators. Together, these data streams form the basis for coordinated decision-making within and across domains. However, this study focuses primarily on the networking layer, which is described in Chapter 3, and on establishing and maintaining the communication infrastructure required for such data exchange. The Borkum Riffgrund 2 wind farm covers an area of approximately 36 km². However, this work focuses on a selected 2×2 km section only. This is because the primary interest lies in the routing between systems in a hybrid network above and below the water surface. This includes the routing and communication of the systems.

### 1.2.1 Possible Sabotage Scenario at the Wind Farm

The protection of Borkum Riffgrund 2 is inherently multidimensional, as security must be ensured in several areas: the seabed (including export and data cables), the underwater domain, the sea surface, airspace, and the cyber domain. The unrestricted maritime access permitted by international law, coupled with the limited attribution of hostile activities, poses additional challenges, complicating detection and response (Sill Torres, 2025). To investigate how UxVs can be used to detect such attacks, a pragmatic scenario is illustrated in Figure 5.

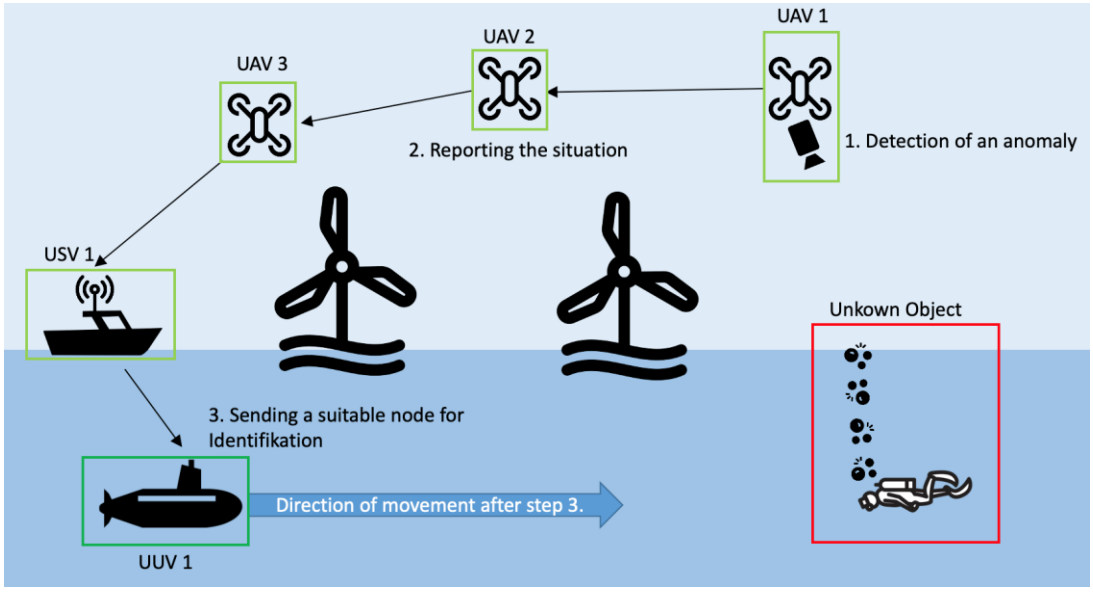


Figure 5: Practical example of a sabotage maneuver at an offshore wind farm. A diver attempts to damage underwater cables. A UAV detects this and uses a Cross-Domain Ad-Hoc routing algorithm to organize additional UxVs, such as a UUV, to create an underwater image.

The right-hand part of Figure 5 depicts an unknown object that is later identified as a hostile diver. The diver may have embarked from the shore or been deployed from a small boat, and is approaching the wind farm underwater. Air bubbles are periodically produced at the surface by the diver and rise to the sea's surface. In clear water, the diver may also be visible a few meters below the surface. These anomalies are detected by UAV 1 in Figure 5 while it is patrolling predefined transit routes. Equipped with a daylight camera, a thermal imaging sensor and a *Light Detection and Ranging* (LiDAR) unit, UAV 1 immediately reports the detected anomaly. Nevertheless, due to its sensing modality, UAV 1 cannot generate an underwater situational picture. However, it is known that a UUV equipped with forward- and side-looking sonar capabilities is operating within the same swarm and surrounding area. UAV 1 forwards the observation to UAV 2 and UAV 3, which act as Relay nodes, as UAV 1 has only a limited radio communication range. The relayed message is then transmitted to a USV that serves as a surface gateway. The USV then converts the radio message into a hydroacoustic transmission and reports it to the nearby UUV. The UUV then moves to the diver’s location and acquires an underwater situational picture. The UUV's presence and actions contribute to deterrence, and it simultaneously returns the situational data to the USV via a hydroacoustic link. The USV then informs headquarters via a satellite communication, who may initiate a response by the Federal Police or the Navy. Although the scenario illustrates a potential application, the underlying challenge is to ensure reliable, domain-spanning communication. The following sections therefore analyze the information types, communication requirements and technological constraints that influence the design of an effective routing mechanism.

### 1.2.2 Communication Objectives

This section defines the key communication objectives that determine the design principles for reliable information exchange in Multi-Domain Operations, in order to better understand the underlying networking requirements. The effective coordination between aerial, surface and underwater platforms hinges on the reliable and adaptive exchange of information. These objectives outline the functional and performance-related criteria that a Cross-Domain Network must fulfil to ensure mission continuity and responsiveness. Based on the use case and related work (Z. Li et al., 2024), (Bauer et al., 2025), it becomes evident that maritime scenarios involve several distinct types of communication. These include radio-to-radio communication between surface and aerial platforms; acoustic

communication between underwater vehicles; and Cross-Domain exchanges between UAVs and UUVs, for which a gateway node is required to translate between radio and acoustic signals. Data transmitted across these links may comprise raw sensor measurements, pre-processed information, commands, control messages or coordination data. Efficient routing mechanisms are required to adapt dynamically to changing topologies and link conditions. The limited bandwidth, particularly in the underwater acoustic domain, necessitates strict control of protocol overhead and intelligent data prioritization.

The communication range varies from several kilometers above-water to just a few hundred meters underwater, making multi-hop communication indispensable. As unmanned systems operate on limited energy budgets, energy-efficient transmission is a critical design factor (Al-Absi et al., 2021). Furthermore, the cooperative behavior of Unmanned Vehicles, which continuously exchange sensor data, control commands and coordination messages, leads to high network load and requires congestion-aware routing. The dynamic topology caused by vehicle mobility requires constant route maintenance, and scalability and security are essential for robustness in mission-critical environments. The corresponding communication relationships and technical constraints are summarized systematically in Table 1. This table distinguishes between the physical and operational domains. This distinction is necessary because, although aerial and surface platforms may use the same physical communication medium (radio), their mobility and operational dynamics differ significantly. For example, UAVs usually have much higher speeds and movement patterns than surface vehicles. The colors in the matrix indicate communication channels types: UAV-UAV, UAV-USV, UAV-UUV, USV-USV, USV-UUV, UUV-UUV. Each colored field represents a bidirectional link, meaning that communication is possible in both directions between the two involved platforms. The matrix in Table 1 shows that communication in maritime Multi-Domain Operations uses different physical media and ever-changing topologies. Each communication path has different constraints regarding range, latency and reliability, creating a highly complex networking environment. These characteristics highlight the need for adaptive routing mechanisms that can maintain end-to-end connectivity in continuously changing conditions. Currently, there is no known comprehensive networking solution that enables seamless communication between these domains.

Table 1: Communication objectives between UxVs and their information flow.

| From \ To | UAV | USV | UUV |
|---|---|---|---|
| **UAV** | Intra-Domain (radio). Coordination and information exchange among aerial nodes. Constraints: high mobility, LoS dependency. | Air-Surface (radio). Command and coordination exchange between aerial and surface nodes. Constraints: dynamic topology, link instability (waves), interference. | Air-Underwater (radio to acoustic via Gateway). Control data sent to underwater nodes. Constraints: gateway dependency, delay, bandwidth mismatch. |
| **USV** | Surface-Air (radio). Alerts and status reports transmitted upward. Constraints: intermittent connectivity (waves), antenna alignment. | Intra-Domain (radio). Coordination and Relay among surface nodes. Constraints: scalability, channel contention. | Surface-Underwater (radio to acoustic). Relay between domains. Constraints: long acoustic delay, reduced reliability, energy demand. |
| **UUV** | Underwater-Air (acoustic to radio via Gateway). Sensor or alert data forwarded upward. Constraints: long propagation delay, packet loss, gateway dependency. | Underwater-Surface (acoustic-radio). Relay between domains. Constraints: synchronization, energy demand, high propagation delay. | Intra-Domain (acoustic). Coordination and data sharing among underwater nodes, including Relay forwarding within the domain. Constraints: low data rate, long RTT, multipath propagation. |

The following section therefore critically assesses whether existing technologies and state-of-the-art approaches in maritime and Cross-Domain communication are sufficient to meet the outlined operational and technical requirements.

### 1.2.3 Limitations of Conventional Solutions and the Role of Ad-Hoc Networking

Various communication technologies are employed in maritime environments to enable information exchange between platforms, illustrated in Figure 6. These can be broadly categorized as Infrastructure-based, Point-to-Point, Mesh, Ad-Hoc, Satellite-based, and hybrid communication systems (Tanenbaum & Wetherall, 2011). While each of these approaches offers specific advantages, they also exhibit distinct limitations when applied to dynamic scenarios.

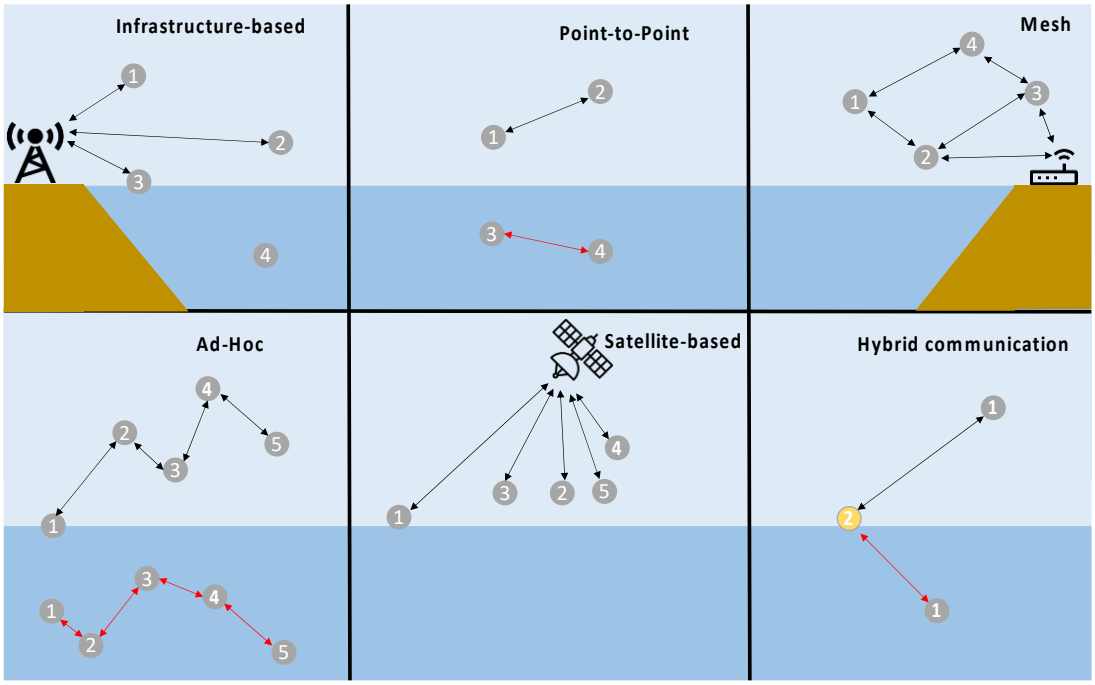


Figure 6: Overview of common communication technologies.

Infrastructure-based systems, such as LTE and VDES, rely on fixed terrestrial or coastal infrastructure to establish stable communication links. While they enable high data rates within coverage areas, they cannot be extended to remote offshore or underwater environments (Alqurashi et al., 2023). Point-to-point communication establishes a direct link between two nodes and is typically used for short-range control, data transfer or Relay operations. While this approach is simple and energy-efficient, it is not scalable in dynamic topologies. When a single node must sequentially establish connections with multiple peers, the resulting scheduling task becomes combinatorially complex, comparable to the Travelling Salesman Problem (TSP) (Xu & Che, 2019). Mesh networks, such as those based on LoRa or the Automatic Identification System (AIS), establish peer-to-peer connections between nodes and enable data transmission across multiple Hops within a single communication domain. They usually depend on pre-defined or semi-static routing structures, where the connectivity pattern is either known or changes slowly over time. While this allows for efficient operation in stable environments, it limits adaptability in dynamic scenarios. Consequently, while mesh networks are well-suited to low-mobility applications, they struggle to maintain connectivity when nodes move rapidly or when Cross-Domain communication is required (Elgharbi et al., 2025). Ad-Hoc Networks operate without fixed infrastructure, dynamically forming and maintaining communication

paths as required. They are self-organizing by nature, robust against topology changes, and well-suited to autonomous operations. Such networks can be established above and below the water surface, enabling communication within these domains. However, the transition from radio-based to acoustic communication, and the seamless integration between these domains, remain open challenges that have yet to be comprehensively addressed in the current state of research. Their performance therefore hinges on efficient routing strategies that can handle node mobility, limited bandwidth, and heterogeneous communication media (Shang, et al., 2025). Satellite-based communication provides near-global coverage and forms the backbone of long-range maritime connectivity. Systems such as Inmarsat and Iridium enable reliable data exchange over thousands of kilometers, but they have limitations such as high latency, limited bandwidth, and an inability to penetrate underwater environments. New constellations in low and very low Earth orbit (LEO and VLEO) have emerged in recent years, offering reduced propagation delay, higher data rates and global accessibility, particularly within the context of emerging 6G networks (X. Luo et al., 2024). Hybrid communication architectures combine various transmission media, such as radio, acoustic and optical links, to enable communication between nodes in the air, on the surface and underwater. While this integration allows for information exchange across domains, it also introduces challenges in terms of synchronization, latency compensation and protocol compatibility. Recent studies have demonstrated that utilizing various transmission paths can enhance reliability and range in maritime environments (Padhan et al., 2025). However, the heterogeneity of the physical media and the complexity this creates at the networking layer make it difficult to achieve a unified, adaptive communication system.

Table 2 provides a comparative overview of the presented communication architectures, summarizing their main characteristics and suitability for maritime operations.

Table 2: Overview of common communication technologies and their applicability for the use case.

| **Network Type** | **Infrastructure Dependency** | **Self-Organization** | **Typical Examples** | **Characteristics and Limitations** |
|---|---|---|---|---|
| **Infrastructure-based** | High- relies on fixed terrestrial or coastal installations | None | LTE,VDES | High data-rate communication is provided within coverage areas, but this cannot be extended |

| | | | | |
|---|---|---|---|---|
| | | | | to offshore or underwater regions. |
| **Point-to-Point** | Low – direct connection between two nodes | None | Control or Relay links between USV–UAV or ship–buoy | Simple and energy-efficient but scales poorly; becomes combinatorically complex when multiple peers are involved (TSP problem). |
| **Mesh** | Medium – local coordination within a single domain | Partial | AIS, LoRa, ZigBee | Multi-hop data transfer with predefined or semi-static routes is efficient in stable environments, but has limited adaptability to fast topology changes. |
| **Ad-Hoc** | None – operates without infrastructure | Full (within a single domain) | MANET, AODV-based networks | Both route discovery and maintenance are dynamic, and the system is resilient to topology changes. However, routing overhead and the presence of heterogeneous links (radio and acoustic) present challenges. The domain change is not yet resolved. |
| **Satellite-based** | Very high – dependent on orbital constellations | None | Inmarsat, Iridium, Starlink (LEO/VLEO) | It enables global coverage, but is limited by latency and bandwidth. It cannot penetrate underwater environments. |
| **Hybrid communication** | Variable – combines multiple technologies | Partial | Cross-Domain Relay systems (e.g., radio–acoustic or radio–optical links) | Integrates different transmission media across domains, increasing reliability |

| | | | | |
|---|---|---|---|---|
| | | | | and range, but requiring synchronization and protocol adaptation. |

As highlighted in Table 2, none of the existing communication architectures can meet the requirements of dynamic, Cross-Domain maritime operations independently. Infrastructure-based and satellite systems rely on fixed installations or orbital constellations, limiting their applicability in mobile, infrastructure-deprived environments. Although point-to-point communication is simple, it cannot scale when multiple UxVs must coordinate simultaneously. Mesh networks provide local, multi-hop connectivity, but they depend on pre-defined routes and are therefore not suitable for highly dynamic topologies. In contrast, Ad-Hoc Networks offer the flexibility, scalability and autonomy required for distributed UxV coordination. They enable nodes to organize themselves and maintain communication links in situations of mobility and intermittent connectivity. However, they are currently restricted to operating within single communication domains. Meanwhile, hybrid communication architectures have shown potential for extending connectivity across domains by combining different transmission media, such as radio, acoustic or optical links. Nevertheless, these systems lack a unifying networking logic that can dynamically adapt routes and handle heterogeneous link characteristics. However, these systems lack a unifying networking logic that can dynamically adapt routes and handle heterogeneous link characteristics. This logic will be investigated in this work in order to enable an adequate solution for future Cross-Domain networked operations of UxVs.

## 1.3 Requirement Overview and Challenge in Cross-Domain Networks

To enable autonomous cooperation between heterogeneous platforms in a Cross-Domain setting, the underlying communication system must fulfil certain functional and non-functional criteria. The following section provides an overview of both functional and non-functional requirements, followed by a discussion of the key challenges inherent in Cross-Domain networking. Functional requirements specify the behavior and capabilities that the system must achieve, while non-functional requirements define quality attributes such as performance, scalability and robustness (Jarzębowicz & Weichbroth, 2021).

These requirements stem from an understanding of network technology, professional discussions within the TKMS ATLAS ELEKTRONIK GmbH.

### 1.3.1 Functional Requirements

- FR 1: Each UxV must be able to discover its neighbors independently with a fixed *Time-To-Live* (TTL).
- FR 2: The system must establish routes between different types of UxVs, such as UAVs, USVs and UUVs.
- FR 3: USVs must act as communication gateways between the surface and underwater domains.
- FR4: UxVs must be able to transmit mission-relevant status information (e.g. position and sensor data) in order to send a payload.
- FR 5: The extended routing protocol must fulfil all of the above functions at least as reliably as the terrestrial domain reference protocol (e.g. the standard AODV protocol).

### 1.3.2 Non-Functional Requirements

- NFR1: The extended protocol is expected to reduce the number of redundant control messages by a significant amount compared to a reference protocol. Redundant messages are a common problem in Ad-Hoc Networks and should be minimized from the beginning.
- NFR2: Under comparable network conditions, route discovery time should be at least 30% faster than that of the reference protocol. This is to ensure that the route is built up as far as possible in the radio medium, which has significantly faster transmission properties.
- NFR3: The underwater communication load must be minimized to avoid hydroacoustic interference and conserve bandwidth.
- NFR4: The routing protocol must be able to operate without centralized control and tolerate dynamic changes in node availability.

It is not possible to quantify these requirements precisely at this stage, as Multi-Domain Ad-Hoc routing has not yet been implemented or systematically studied in this form. Therefore, the requirements have been formulated qualitatively and will serve as a conceptual foundation for the design and subsequent evaluation of the protocol.

## 1.4 Objectives and Research Questions

In order to provide an uninterrupted situational awareness of the surface and underwater environments for the protection of CMIs, this work involves the development, implementation and evaluation of an enhanced Ad-Hoc routing protocol. The aim is to provide a robust self-organizing networking foundation that supports Cross-Domain communication between UAVs, USVs and UUVs. The proposed protocol is aimed at outperforming classical approaches, particularly with regard to improving the efficiency of underwater communication, reducing route discovery time and lowering control message overhead. The following research questions are derived from these objectives and the challenges outlined in the previous sections:

- RQ1: What are the specific communication and routing limitations that arise when conventional Ad-Hoc routing protocols are applied in Cross-Domain environments?
- RQ2: How can classical Ad-Hoc routing protocols be extended to support reliable communication across air, surface and underwater domains in a heterogeneous maritime network?
- RQ3: How can a physical demonstrator be used to parameterize and validate a simulation framework for realistic evaluation of routing behavior?
- RQ4: To what extent does the proposed protocol reduce the load on underwater communication, the time taken for route discovery, and the number of control messages compared to a reference protocol?

The subsequent chapters will explore these research questions, beginning with an overview of the State of the Art. This will be followed by the development of the extended routing protocol, the implementation of a physical demonstrator and the design and execution of a simulation framework.

## 1.5 Scientific Contribution and Structure of this Work

This thesis addresses the challenge of enabling communication in heterogeneous, infrastructure-less maritime environments. It proposes an extended Ad-Hoc routing protocol tailored for Multi-Domain Networks involving UAVs, USVs, and UUVs. The scientific contribution lies in analyzing existing Ad-Hoc routing protocols and identifying a suitable protocol to be extended to support Cross-Domain Communication in such environments. The focus is on minimizing underwater traffic, reducing route discovery time and lowering control message overhead. USVs are introduced as dynamic Relay gateways between radio- and hydroacoustically connected nodes, enabling efficient data transfer without the need to establish a full route in the underwater domain. In order to evaluate the protocol under realistic conditions, a physical demonstrator is implemented using Raspberry Pis, acoustic modems and UAVs to increase the range as a Relay node. This demonstrator is used to parameterize a custom MATLAB-based simulation framework that reflects the domain-specific constraints of underwater communication, such as low bandwidth and high propagation delay. This simulation environment is then used to analyze several use cases derived from protecting the Borkum Riffgrund 2 offshore wind farm. By answering the research questions formulated in Section 1.4 this work contributes to the design and evaluation of a novel Multi-Domain routing protocol that enables UxVs to collaboratively establish a shared situational picture. This represents a step towards autonomous, infrastructure-independent communication systems for protecting maritime critical infrastructure. Chapter 2 provides an overview of the current state of research, including existing routing protocols, their fundamental mechanisms and limitations in Cross-Domain environments. Chapter 3 introduces the logical architecture of the proposed solution, first by describing the hybrid AODV reference protocol, and then by outlining the design of the extended AODV versions. Chapter 4 describes the simulation framework, including its parametrization based on the physical demonstrator and the implementation of representative use cases. Chapter 5 presents the evaluation results and provides a critical analysis of the protocol's performance. Potential for future extensions is considered and evaluated in Chapter 6 alongside the results. Chapter 7 concludes this work by summarizing the main findings. The developments presented in this work were largely carried out at the test site of TKMS ATLAS ELEKTRONIK GmbH. Several sections of the research have already been published in the form of conference papers and patent applications.

# 2 State of the Art and Referenced Systems

Developing Multi-Domain Ad-Hoc Networks for maritime applications requires a thorough understanding of existing communication protocols and available simulation frameworks. The focus of this chapter is on providing an overview of current research in Ad-Hoc Networking, with a particular emphasis on the surface and underwater domains. It also outlines the limitations of these protocols when applied to hybrid environments, identifying key gaps in existing simulation approaches that motivated the development of a dedicated framework in this study.

The *Open Systems Interconnection* (OSI) reference model, as defined by the International Telecommunication Union (ITU-T, 1994), offers a systematic approach to understanding communication systems. It divides the networking process into seven hierarchical layers, each of which is responsible for a distinct set of functions related to data exchange. In Figure 7 this layered architecture is illustrated and highlights the layers most relevant to this work.

The physical layer is responsible for transmitting raw bits via a communication medium. In the context of this work, this is crucial since communication conditions vary substantially between domains. Underwater communication relies on hydroacoustic transmission, which is characterised by long propagation delays and limited bandwidth. Additional effects such as delay spread, Doppler shift, wave motion and varying signal-to-noise ratios further influence transmission reliability in the physical layer of underwater communication (Junejo et al., 2023). The data link layer ensures reliable transmission across a single link by handling framing, error detection, and medium access control (MAC). While this is not the main focus of this work, the effects of the data link layer are included in the simulation to capture realistic communication behavior. The characteristics of the medium have a particular effect on this layer in underwater communication. These characteristics are low bandwidth, long propagation delays and a high probability of collisions due to the shared acoustic channel. These characteristics make designing efficient MAC protocols especially challenging, as collisions can significantly reduce throughput and increase energy consumption. Recent studies have proposed time-division and clustering-based scheduling approaches to mitigate interference and improve link reliability in multi-hop underwater acoustic networks (Yang et al., 2023).

The network layer is responsible for managing logical addressing, packet forwarding and route maintenance in dynamic topologies. It ensures end-to-end connectivity, even in the absence of fixed infrastructure. In mobile Ad-Hoc and underwater Networks, however, this layer faces specific challenges, including high mobility, long propagation delays, limited bandwidth and high error rates. These conditions make reliable route discovery and maintenance difficult, since conventional terrestrial protocols often depend on stable, high-capacity links (Tanenbaum & Wetherall, 2011). In underwater communication, the strong influence of environmental conditions means that routing mechanisms must adapt to changes in link quality and topology. As demonstrated in (Busacca et al., 2024), adaptive and predictive routing strategies can minimize retransmissions, reduce latency and extend node lifespan. Reactive methods create routes on demand, whereas proactive approaches continuously maintain routing tables to shorten discovery times. Both must be adapted to the high latency and limited energy resources of acoustic networks. Additional functions of the network layer include security, synchronization, power management and localization (Gola et al., 2023). While these aspects are important for complete underwater networks, they are not the main focus of this work.

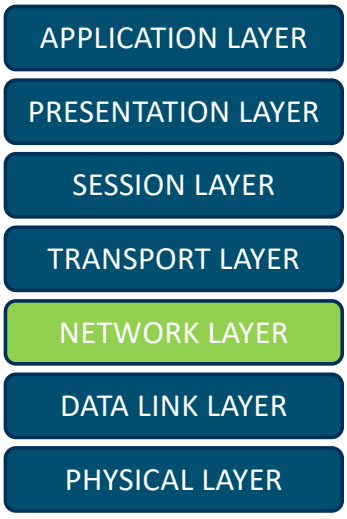


Figure 7: ISO/OSI model with individual layers. The network layer is marked because it is the core of this work. The layers are named in reference to (ITU-T, 1994).

The transport layer is essential for ensuring end-to-end data delivery, flow control and reliability. In maritime and underwater environments, however, its mechanisms are often simplified due to long propagation delays, high error rates and asymmetric links. Conventional acknowledgement-based schemes perform poorly in such conditions as

retransmissions can increase latency and energy consumption further. However, recent work has shown that optimized retransmission strategies and adaptive timeout mechanisms can improve throughput and reliability by mitigating the effects of long propagation times and intermittent connectivity in underwater channels (Su et al., 2019). The session layer establishes, manages and terminates logical communication sessions between applications. In maritime and Ad-Hoc Networks, however, these functions are usually incorporated into higher-level software or middleware, meaning a dedicated session layer is rarely implemented. The presentation layer ensures that transmitted data is correctly formatted, encoded and interpreted between systems. It handles tasks such as data compression, encryption, and translation. However, in most autonomous systems, these processes are defined within the application logic rather than within a distinct protocol layer. The application layer acts as an interface between the communication network and the mission software. In maritime systems, it facilitates the exchange of sensor data, status messages and control commands between unmanned platforms. Rather than using standardized internet protocols, middleware-based frameworks are typically employed to ensure reliable and interoperable communication. A widely adopted approach is the Data Distribution Service (DDS), which is a data-centric middleware that follows a publish–subscribe model. Operating above the network layer, DDS provides configurable Quality of Service (QoS) parameters for latency and reliability, while abstracting lower-level transmission details. DDS has proven effective in UAV and maritime networks for cooperative perception and mission coordination, offering resilience against intermittent connectivity and dynamic network conditions (Ribeiro et al., 2017).

## 2.1 Technical Limitations in Multi-Domain Communication

Multi-Domain Communication involves integrating radio- and hydroacoustic-based transmission domains, which are typically encountered in scenarios involving aerial/surface (UAV/USV) and underwater (UUV) vehicles. Although traditional MANET routing protocols are designed for radio-based communication, they cannot be applied directly to underwater domains. These limitations can be grouped into two central challenges. In Multi-Domain Networks involving underwater nodes, communication is based on hydroacoustic signals rather than radio frequencies. This is because electromagnetic waves attenuate rapidly in water, particularly at high frequencies, rendering radio-based transmission impractical for medium or long-range underwater communication (Akyildiz et al.,

2005). However, hydroacoustic waves can propagate over much greater distances in water, despite travelling more slowly, having reduced bandwidth and being more susceptible to environmental interference. The main challenges underwater can be categorized into three groups: Propagation delay, speed of hydroacoustic waves and underwater noise.

(a) Propagation Delay

One of the main reasons why underwater communication differs significantly from radio-based transmission is that acoustic absorption is highly frequency-dependent. The total attenuation of an underwater acoustic signal increases with both frequency and propagation distance, and can be composed of two main components: geometric spreading and frequency-dependent absorption (Khan et al., 2018). In underwater acoustics, the transmission loss $A(d, f)$ in dB is commonly expressed as the sum of spreading loss and absorption loss:

$$A(d, f) = 10 \cdot k \cdot \log_{10}(d) + \alpha(f) \cdot d. \tag{2.1}$$

Here, $d$ is the transmission distance (in kilometers or meters, depending on the definition of $\alpha(f)$ ), $k$ is the spreading factor, and $\alpha(f)$ is the frequency-dependent absorption coefficient expressed in dB per unit distance. Typical values for $k$ are 1 for cylindrical spreading, 2 for spherical spreading, and 1.5 in realistic underwater environments (Khan et al., 2018). The absorption coefficient $\alpha(f)$ is commonly obtained from the Thorp approximation, which is valid for frequencies between 100 Hz and 1 MHz. For a frequency $f$ expressed in kHz, the Thorp formula is given by:

$$\alpha(f) = 0.11 \cdot \frac{f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2.75 \cdot 10^{-4} \cdot f^2 + 0.003. \tag{2.2}$$

The Thorp formula models the various mechanisms of hydroacoustic absorption in seawater, including viscosity and ionic relaxation (Aydin & Onur, 2020). As a result, higher frequencies experience stronger attenuation and have shorter ranges. Underwater communication is therefore typically restricted to frequencies below 10 kHz, leading to low available bandwidth. For instance, the Evologics S2C R 18/34 modem operates between 18 and 34 kHz, providing a maximum bit rate of 13.9 kbps and communication ranges of up to 3.500 m under favorable conditions (EvoLogics, 2014). Routing protocols that rely on frequent control messages or broadcast-based route discovery can quickly overload such low-bandwidth channels, which degrades overall network performance and

increases packet loss. In practice, many commercial underwater modems operate within this low-frequency range.

(b) Speed of Hydroacoustic Waves

In a hydroacoustic medium, the overall latency of transmitting a data packet is composed of transmission delay and propagation delays. The propagation delay in hydroacoustic communication depends on the ratio of the distance and the speed of sound of the water (Kartha & Jacob, 2015):

$$T_{\text{latency}} = T_{\text{tx}} + T_{\text{prop}}. \tag{2.3}$$

However, in radio communication, only the transmission delay is considered, as the propagation delay is negligible due to the high speed of electromagnetic wave propagation. The transmission delay depends exclusively on the packet size, the bit rate and the coding method used. The formula of the transmission delay is composed as outlined below:

$$T_{\text{tx}} = \frac{L \cdot 8 \cdot 1000}{R} \cdot \frac{1}{c}. \tag{2.4}$$

The transmission delay $T_{\text{tx}}$ is calculated from the packet size $L$ (in bytes), multiplied by 8 to convert to bits, and multiplied by 1000 to convert to ms. This value is divided by the bit rate $R$ (in bits per second) and then divided by the coding rate $c$ to take into account any redundancies due to error correction. A packet size of 48 bytes, a bit rate of 10 kbit/s and a coding rate of 1/2 should be taken as an example value. This corresponds to relative values, 48 bytes would be the size of a Route Request according to RFC AODV (Perkins et al., 2003). Under these assumptions, the effective bit rate is reduced to 5 kbit/s, resulting in a transmission time of 76.8 ms.

In underwater hydroacoustic communication networks, propagation delay ($T_{\text{prop}}$) plays a pivotal role in determining routing efficiency, protocol behavior, and network synchronization. Due to the low speed of hydroacoustic signals compared to electromagnetic waves, the delay can significantly affect the performance of the network, especially over greater distances. Therefore, accurate modeling of propagation delay is essential to simulate realistic communication behavior. The propagation delay $T_{\text{prop}}$ between two nodes $j$ and $k$ is calculated between their distances and the sound of speed in water. The $distance_{jk}$ is the Euclidean distance between the nodes. The variable $c(T,S,D)$ represents the speed of sound in water, depending on the temperature $T$, the salinity $S$, and the average depth $D$. The formula for the calculation can be described as:

$$T_{\text{prop}} = \frac{distance_{jk}}{c(T,S,D)}. \tag{2.5}$$

To accurately model the propagation delay between two nodes, it is essential to calculate the speed of sound *c(T,S,D)* in the underwater environment. The speed of sound in water depends on the physical parameters temperature *T* (in degrees Celsius), the salinity *S* (in practical salinity units, PSU), and the depth *D* (in kilometers). In this work, the empirical model provided by the UK National Physical Laboratory (NPL) (National Physical Laboratory, 2000), based on the formulation originally proposed by Coppens (1981), is used to estimate the sound velocity. In the original NPL and Coppens formulations, water temperature is not used directly as *T* in degrees Celsius. Instead, the model introduces a scaled temperature variable, defined as *T*/10. The implementation consistently uses *T* divided by 10 because all coefficients in the equation are based on this scaled temperature. This model takes into account linear and non-linear dependencies on temperature, salinity and depth. The complete Equation is given as:

$$c(T,S,D) = 1449.05 + 45.7T - 5.21T^2 + 0.23T^3 + (1.333 - 0.126T + 0.009T^2)(S-35) + D(16.23 + 0.253T) + (0.213 - 0.1T)D^2 + [0.016 + 0.0002(S-35)](S-35)TD. \tag{2.6}$$

This formulation reflects the predominant influence of temperature on the speed of sound, while salinity introduces a smaller modification. The depth dependence includes a linear term, a quadratic term and a cross term involving temperature and salinity.

(c) Underwater Noise

In underwater hydroacoustic communication, ambient noise is a critical factor that affects the quality of signals and, consequently, the reliability of routing protocols. The literature on ocean hydroacoustic distinguishes four main sources of noise, each of which dominates a specific frequency range (Stojanovic, 2007):

- Turbulence noise: dominant below 10 Hz.
- Shipping noise: dominant between 10 Hz and 100 Hz.
- Wave noise: dominant between 100 Hz and 100 kHz.
- Thermal noise: dominant above 100 kHz.

The noise levels, expressed in decibels relative to 1 µPa per Hz, are frequency dependent and can be modeled as follows (Stojanovic, 2007):

Turbulence noise: $N_t(f) = 17 - 30 \cdot \log_{10}(f).$ (2.7)

Shipping noise: $N_s(f) = 40 + 20(s - 0.5) + 26 \cdot \log_{10}(f) - 60 \cdot \log_{10}(f + 0.03).$ (2.8)

Wind/wave noise: $N_w(f) = 50 + 7.5 \cdot w + 20 \cdot \log_{10}(f) - 40 \cdot \log_{10}(f + 0.4).$ (2.9)

Thermal noise: $N_{\text{th}}(f) = -15 + 20 \cdot \log_{10}(f).$ (2.10)

Total noise: $N_{\text{total}}(f) = 10 \cdot \log_{10}\left(10^{0.1 \cdot N_t(f)} + 10^{0.1 \cdot N_s(f)} + 10^{0.1 \cdot N_w(f)} + 10^{0.1 \cdot N_{\text{th}}(f)}\right).$ (2.11)

Here, *s* denotes the shipping activity (ranging from 0 to 1 for high activity), and *w* represents the wind speed in meters per second (m/s). The frequency $f$ is expressed in kHz. These physical challenges are crucial for understanding Ad-Hoc routing protocols in the underwater domain. There are specially adapted strategies for this purpose. These are presented in the following section.

## 2.2 Maritime Ad-Hoc Networks for Surface and Underwater Communication

MANETs are decentralized systems that enable flexible communication in dynamic environments. Specialized MANETs, such as VANETs for vehicular communication and FANETs for aerial systems, address domain-specific challenges (Kaur & Singh, 2018). Their applications include disaster response, infrastructure monitoring, military operations and maritime scenarios involving the coordination of UAVs, USVs, and UUVs. When considering MANETs for maritime applications, it is evident that there are routing protocols specialized for terrestrial and Underwater Acoustic Sensor Networks (UASNs). Terrestrial MANET routing protocols are primarily categorized as proactive, reactive, or hybrid based on their route discovery mechanisms. Proactive protocols maintain up-to-date routing tables, which offers low latency but requires more control overhead. Reactive protocols discover routes on demand, which minimizes overhead but introduces initial delay (Stepanov et al., 2023). In contrast, underwater routing protocols are more commonly classified as energy-based, data-centric, or geographic-based due to the distinct physical constraints of the underwater hydroacoustic environment (J. Luo et al., 2021).

Table 3: Overview of routing protocols in surface and underwater domains. These are reviewed in terms of their suitability for the MDO scenario.

| **Domain** | **Protocol & Type** | **Advantage** | **Disadvantages** | **Suitability for MDO-Routing** |
|---|---|---|---|---|
| **Surface (Terrestrial)** | AODV (Ad-Hoc On-Demand Distance Vector)<br><br>Reactive protocol. Discovers routes on demand using control messages (RREQ and RRREP) and stores them in a routing table. It forwards packets based on next-hop information. (D. Ramphull et al., 2021), (Arega et al., 2020) | Routes are established on demand with low control overhead and are scalable in dynamic networks. Sequence numbers avoid routing loops. | The route discovery process is initially delayed, and performance experiences a notable decline when nodes are highly mobile or the network encounters frequent link failures. | **High**: Supports dynamic discovery and can be extended with translation logic. |
| **Surface (Terrestrial)** | DSR (Dynamic Source Routing)<br><br>Reactive protocol. Routes are established on demand (RREQ and RREP). DSR protocol has the entire route included in each packet header, thereby eliminating the need for routing tables. (Zafar et al., 2016), (Sureshbhai et al., 2018), (Nayyar, 2018) | Periodic control messages are not required, so bandwidth usage is reduced. Discovery is less frequent because of route caching, and static networks have good throughput. | Source routing results in high routing overhead and limited scalability. Outdated cached routes can also reduce throughput in mobile or large-scale networks. | **Limited**: Source routing increases header size and lacks modular handling of heterogeneous links. |
| **Surface (Terrestrial)** | OLSR (Optimized Link State Routing)<br><br>Proactive protocol. A link-state routing mchanism that uses *Hello* messages to periodically discover neighbors and disseminates topology information via selected Relay nodes to compute | Routing information is always available. Multipoint Relays (MPRs) reduce the number of control messages. | Periodic updates result in high overhead for control messages, making it unsuitable for energy-constrained or mobile nodes. | **Low**: Relies on consistent link-state propagation. |

| | | | | |
|---|---|---|---|---|
| | full routes. (Gangopadhyay & Jain, 2023), (Dong, 2016), (Kurniawan et al., 2020) | | | |
| **Surface (Terrestrial)** | BATMAN (Better Approach To Mobile Ad-Hoc Networking)<br><br>Proactive protocol. A hop-by-hop routing mechanism that selects the best next Hop without calculating full paths by using periodic Originator Messages (OGMs). (Liu et al., 2018), (Kiran et al., 2018), (Baharuddin et al., 2020) | Simple and distributed: Each node only knows the best next Hop. Robust in dynamic topologies. | There is no end-to-end path awareness, and routing is based on local decisions only. There is constant control traffic. | **Limited**: Its proactive nature causes continuous overhead and is unsuitable for low-bandwidth domains, such as underwater links. |
| **Underwater (Hydroacoustic)** | UWIA-AODV (Improved Ad-Hoc On-Demand Distance Vector Routing Protocol for Underwater Acoustic Sensor Networks)<br><br>Reactive protocol. An AODV-based routing mechanism for underwater hydroacoustic networks. Implicit acknowledgements (ACK) and residual energy-based routing are used for energy balancing. (X. Wang et al., 2023) | A higher PDR, better energy distribution, and real-world validation can be found in the publication. | Additional buffer due to ACK required, the delay is due to the waiting period during route discovery, with control traffic remaining significant. | **High:** It is similar to AODV, but due to the high number of implicit acknowledgments, it is not ideal for minimizing underwater routing traffic. It is designed for reliability, not minimalism. |
| **Underwater (Hydroacoustic)** | DBR (Depth-Based Routing)<br><br>Geographic forwarding protocol. A localization-free underwater routing mechanism that uses node | Simple, stateless, and no GPS required, it is well-suited for sparse and mobile | Redundant transmissions may occur if nodes fail to communicate properly. DBR requires a depth sensor and has | **Low:** It cannot address gateways or domains explicitly. Forwarding is solely based on depth. |

| | | | | |
|---|---|---|---|---|
| | depth to forward packets, with shallower nodes transmitting earlier and deeper nodes suppressing transmission through overhearing. (Shaf et al., 2018), (Arega et al., 2020) | underwater environments. | no awareness of node conditions or functional roles. | |
| **Underwater (Hydroacoustic)** | VBF (Vector-Bases-Forwarding)<br><br>Geographic forwarding protocol. The forwarding of packets is only done by nodes that are located within a specific forwarding zone, which is defined as the area surrounding the routing vector from the source to the destination. (Hakim et al., 2018), (Ibrahim et al., 2015) | VBF is energy-efficient, can be scaled up in sparse networks, reduces redundant transmissions, and saves bandwidth. | The accuracy of the position information is a key factor, and it has low robustness in environments that are either dense or dynamic. | **Low:** Requires accurate position data of the source and destination. It is unsuitable if the positions of the gateway nodes (e.g., USVs) are unknown or change frequently. |
| **Underwater (Hydroacoustic)** | GUWMANET (Gossiping in Underwater Mobile Ad-Hoc Networks)<br><br>Flooding-based protocol. The flooding mechanism uses gossip logic, in which nodes forward only unseen messages with full addressing. (Goetz & Nissen, 2012), (Goetz & Nissen, 2015) | Robust to mobility and topology changes, does not require localization, and is simple to implement. | High energy consumption due to redundant rebroadcasting and large packets because of full addressing. | **Low**: Flooding overhead and large packets make MDO inefficient and slow Cross-Domain communication. |

Table 3 provides an overview that categorizes established routing protocols for terrestrial and underwater MANETs. For terrestrial environments, it lists well-known protocols such

as AODV, DSR, OLSR, and BATMAN, which represent reactive, proactive, and hybrid approaches. These protocols are optimized for radio-based communication, dynamic topologies, and minimal delay. In contrast, the underwater domain relies on protocols such as UWIA-AODV, DBR, VBF, and GUWMANET. These protocols are adapted to the hydroacoustic communication medium and its constraints, such as high latency, limited bandwidth, and energy efficiency. Underwater protocols are usually classified as data-centric, flooding-based, or geographic-based (Singh & Tagore, 2019). While these protocols are well-suited to their respective domains, none natively support routing across media boundaries, such as from hydroacoustic to radio-frequency networks. Therefore, dedicated Cross-Domain routing mechanisms are necessary for MDO in hybrid scenarios, in which UAVs, USVs, and UUVs collaborate in a shared mission environment. Although recent studies have addressed this issue, there are still significant gaps in terms of practical application.

The first approach is a Cross-Domain routing method based on *Improved multiobjective genetic algorithm* (IMOGA), which formulates the communication problem as a multi-objective optimization task (S. Zhang et al., 2024). It selects a set of Relay nodes, or gateways, to connect underwater and aerial platforms, with the aim of creating a stable network topology. Although this approach shows promise in optimizing link quality and energy consumption, it assumes static node positions and relies on centralized path computation. As such, it does not provide distributed route discovery or dynamic gateway detection capabilities, which are a central challenge for our application, as seen in FR1, FR2 and NFR4 (S. Zhang et al., 2024).

The second approach is known as Bubble Routing. This uses a geographic forwarding principle based on vertical (z-axis) progression (Aman et al., 2023). In this mechanism, packets are forwarded hop-by-hop towards nodes at a higher altitude, ultimately reaching airborne UAVs that can communicate long-range with a ground station. While this strategy facilitates the upward movement of data across domain boundaries, it does not establish routes in the traditional sense and lacks mechanisms to manage reliability across multiple Hops within and across domains. This is, however, a fundamental requirement for our routing to FR1, NFR3 and FR3.

A third recent approach is the *Cross-Media Routing and Clustering Algorithm* (CMRCA) (Ding et al., 2021). This approach combines clustering-based topology control with hierarchical data forwarding across domains. AUVs form local clusters, where cluster heads

act as coordinators and forward data to Relay nodes, which then pass the data towards UAVs. This method relies on the known positions of nodes and their communication ranges to establish efficient links (Ding et al., 2021). Although CMRCA establishes a structured data path from the underwater to the aerial domain, it does not specify how data is routed back from UAVs to AUVs. Furthermore, it does not implement distributed route discovery or dynamic Relay selection. The requirement for node positions to be known globally represents a further limitation, particularly in dynamic, infrastructure-less maritime scenarios where such information is typically unavailable.

Another approach is the *radio-acoustic opportunistic hybrid protocol* (RAOH), which uses the AODV protocol for underwater-to-surface routing. RAOH introduces the concept of opportunistic communication between underwater and surface nodes early on, leveraging a surface routing mechanism once data reaches the radio domain (Jiang et al., 2023). However, RAOH does not present a holistic end-to-end routing strategy. Specifically, it lacks a defined mechanism for routing from the surface back into the underwater domain and it does not support multi-hop underwater-to-underwater routing scenarios. The underwater routing logic is limited to a maximum of two Hops, which restricts its scalability for wider underwater networks. Although RAOH provides an initial concept for integrating underwater nodes with the surface network, the gateway is currently selected based solely on the first *Hello* message received. This approach does not account for the actual quality or distance of the underwater path to the destination, which is important given the significant differences in propagation delay caused by varying underwater distances. RAOH also lacks the ability to incorporate geographic information or assess signal quality and has no gateway selection logic beyond immediate neighbor availability. The lack of practical tests or real-world evaluations further limits its applicability hybrid routing scenarios (Jiang et al., 2023).

The *improved grey optimization algorithm* (IGWO)-based Cross-Domain routing approach (Shang, Yu, et al., 2025) uses a clustering structure in which nodes are organized into clusters and cluster heads manage inter-cluster communication. The IGWO algorithm is used to select the optimal cluster heads and Relay, thereby minimizing delay and energy consumption. However, this method relies on prior knowledge of node positions, static clustering and centralized path computation. Consequently, it is not suitable for highly dynamic, distributed maritime scenarios involving frequent topology changes and real-time constraints.

These recent publications demonstrate the growing scientific interest in the topic of Cross-Domain communication. However, existing approaches usually require prior knowledge of the positions of all nodes and depend on static or centrally computed structures. On the other hand, opportunistic approaches rely on continuously broadcasting control messages and often select the first node to respond without evaluating its suitability in terms of link quality or proximity to the final destination. Consequently, these approaches are not well suited to our highly dynamic maritime environments. The high mobility, topological changes and real-time constraints of UxVs necessitate adaptive, distributed routing mechanisms.

## 2.3 Analysis of the AODV Routing Protocol: Design, Limitations, and Extensibility

The AODV routing protocol is one of the most widely used reactive approaches to routing in MANETs and is described in the *Request for Comments* (Perkins et al., 2003). Unlike proactive protocols such as OLSR, which maintain continuous routing tables, AODV only establishes routes when data transmission is required. While this on-demand behavior reduces control overhead in dynamic topologies, it introduces route discovery latency (Deepika et al., 2021). When a source node needs to communicate with a destination for which there is no valid route, it sends out a broadcast called a *Route Request* (RREQ), illustrated in Table 4. Each node that receives an RREQ stores information about the originator in its routing table, including the reverse route needed for potential future communication. Additionally, the node records the sequence number contained in the message to ensure that only the most recent route information propagates throughout the network. After updating its routing state, the node rebroadcasts the RREQ to its neighbors, thereby expanding the search region in a controlled, hop-by-hop manner. This process continues until the message either reaches the destination node or an intermediate node that already maintains a recent and valid route to the intended target.

Table 4: Description of message type RREQ. This comprises a total of 23 bytes.

| Field | Size in bits | Size in bytes | Description |
|---|---|---|---|
| Type | 8 | 1 | Identifies the message as a Route Request |

| | | | |
|---|---|---|---|
| Flags | 8 | 1 | J, R, G, D, U and reserved bits |
| Hop Count | 8 | 1 | Number of hops so far |
| RREQ ID | 32 | 4 | Unique request identifier |
| Destination IP address | 32 | 4 | Address of the destination node |
| Destination sequence number | 32 | 4 | Latest known sequence number for the destination |
| Source IP address | 32 | 4 | Address of the source node |
| Source sequence number | 32 | 4 | Latest sequence number for the source node |
| **Total** | **184** | **23** | |

Once the destination or an intermediate node with a valid route is reached, a *Route Reply* (RREP) is generated and sent back as a unicast message along the reverse path. The structure of the RREP is shown in Table 5.

Table 5: Description of message type RREP. This comprises a total of 20 bytes.

| Field | Size in bits | Size in bytes | Description |
|---|---|---|---|
| Type | 8 | 1 | Identifies the message as a Route Reply |
| Flags | 8 | 1 | R, A and reserved bits |
| Prefix Size | 8 | 1 | Length of the destination network prefix in bits (0 = host route). |
| Hop Count | 8 | 1 | Number of hops back to the source |
| Destination IP address | 32 | 4 | Address of the destination node |
| Destination sequence number | 32 | 4 | Latest known sequence number for the destination |
| Originator IP address | 32 | 4 | Address of the node that originated the RREQ |
| Lifetime | 32 | 4 | Duration that the route is considered valid |
| **Total** | **160** | **20** | |

Once the RREP reaches the source, the routing table of each node along the path is updated. Each entry contains the destination address, the next Hop, the Hop count, the latest known sequence number and a validity flag. This ensures that subsequent data packets can be forwarded directly, without the need to rediscover the route. Routes remain active for as long as they are used. Otherwise, they expire after a predefined period of time. These two message types form the core of the AODV route discovery mechanism, while *Route Error* (RERR) messages are used to notify link breaks and trigger route maintenance (Tanenbaum & Wetherall, 2011). To prevent duplicate processing and routing loops, AODV uses a number of internal mechanisms. Each node maintains a message cache that stores recently seen RREQ identifiers and originator addresses. Duplicate requests are discarded. To prevent redundant processing and routing loops, each node stores a cache of recently received RREQ identifiers and originator addresses. This ensures that duplicate requests are discarded. A Time-to-Live counter limits the maximum Hop count of each routing message, ensuring outdated or looping packets are removed from the network. In dynamic environments, the TTL mechanism also helps to limit the area affected by flooding during route discovery (Bindra et al., 2013). Some AODV control messages include operational flags that specify the purpose of the message, such as initiating route discovery, acknowledging replies or broadcasting route errors. These mechanisms give the protocol a certain degree of flexibility, enabling it to adapt to different operational conditions.

In a hybrid maritime scenario, as conceptually illustrated in Figure 8, direct implementation of the classical AODV protocol in a hybrid version could enable route discovery between aerial, surface and underwater nodes. In this configuration, gateways located at the interface between communication domains would forward routing messages across radio and acoustic channels. However, this approach would only work if all gateway nodes could continuously translate between both media types and if propagation conditions remained stable. In practice, these assumptions rarely hold true. Differences in latency, bandwidth and link availability between domains lead to significant desynchronization and message loss, rendering the direct adoption of classical AODV inefficient for Cross-Domain communication.

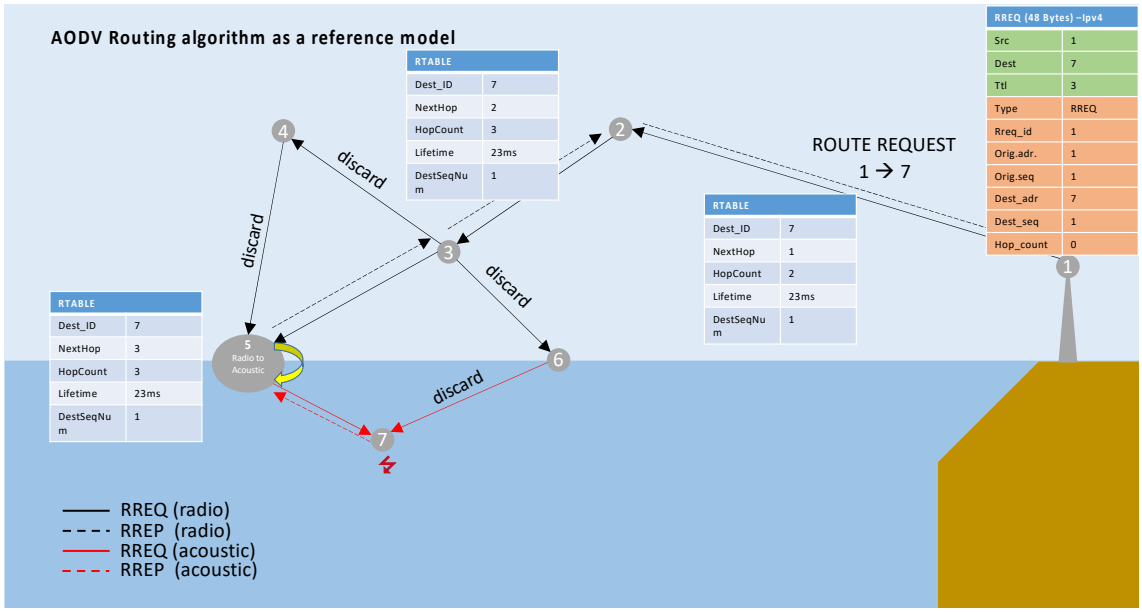


Figure 8: AODV reference protocol sequence. A sender node 1 attempts to discover a route to the destination node 7 via RREQ. Routing tables are stored decentrally on the nodes.

Figure 8 illustrates the behavior of the classical AODV protocol in a hybrid maritime scenario. In this example, Node 1 aims to reach Node 7, which is situated in the underwater domain. In accordance with the AODV specification, Node 1 broadcasts a RREQ message across the network. Black arrows represent RREQ messages transmitted via the radio channel, while red arrows indicate RREQs forwarded acoustically. Each intermediate node that receives an RREQ stores the address of the previous Hop from which the message was received in its routing table as a reverse route entry. Before forwarding the message, the node checks its cache to verify whether the same RREQ has already been processed (identified by its unique RREQ ID and originator address). Duplicate requests are discarded to prevent routing loops and redundant broadcasts. Crucially, only the first RREQ with a given sequence number and originator ID is entered into the routing table. Any subsequent RREQs with the same identifiers arriving later are ignored, even if they correspond to shorter or more reliable paths. While this mechanism reduces control overhead, it can result in suboptimal route selection in environments with asymmetric propagation delays, particularly where acoustic and radio transmissions coexist. When a RREQ reaches a gateway node, the gateway forwards the request into the underwater domain using the same broadcast principle. However, whereas propagation delays in the radio channel are typically in the range of microseconds, hydroacoustic transmission introduces delays of several hundred milliseconds. Consequently, RREQ messages from multiple gateways may overlap in time, resulting in interference and signal collisions within the

underwater medium, described in section 2.1. The extent of this interference depends strongly on the signal-to-noise ratio (SNR), the modulation scheme employed (e.g. FSK, PSK or OFDM) and the Bit Error Rate (BER) resulting from multipath propagation, scattering and ambient noise. In scenarios with a low SNR or with overlapping transmission bursts, the probability of decoding errors increases significantly, which can lead to corrupted RREQ packets and incomplete route discovery (B. Li et al., 2019). Once Node 7 receives a valid RREQ, it generates a RREP message and sends it back to the source along the reverse path recorded in the routing tables. This message traverses both acoustic and radio links again, resulting in a total round-trip delay that is primarily determined by the underwater segment. Once the route has been established, data packets can be transmitted along it until the route expires or a link failure occurs.

This simplified scenario illustrates the behavior of the classical AODV mechanism in a hybrid environment. Although the protocol facilitates basic route discovery, it does not consider differences in propagation delay, bandwidth or link reliability between communication domains. Notably, AODV does not implement selective gateway management, meaning every gateway node that receives a Route Request forwards it to the next domain simultaneously. Consequently, all gateways broadcast redundant RREQs into the underwater channel, even when only one would suffice to reach the destination. This uncontrolled flooding leads to unnecessary transmissions in the hydroacoustic medium, which has limited bandwidth, long propagation delays and high energy consumption. Consequently, the standard AODV protocol exhibits significant inefficiencies and reliability issues when applied to Cross-Domain maritime networks. To ensure scalable operation, the participation of gateways must be limited to a small, carefully selected group of nodes in order to reduce redundant control traffic and preserve valuable underwater communication resources. In particular, establishing a route within the underwater domain introduces considerable overhead and delay. Each RREQ and RREP transmitted acoustically must traverse a slow medium which is highly susceptible to transmission errors due to multipath propagation, noise and signal attenuation. The large number of packets required for route discovery increases the probability of collisions and bit errors further. Therefore, it can be concluded that underwater route establishment should be minimized wherever possible. This implies keeping the number of packets transmitted and received acoustically as low as possible, favoring routing strategies that limit control traffic within the hydroacoustic segment while maintaining connectivity across domains. The following

core challenges can be identified when applying AODV to a Multi-Domain maritime scenario, based on these observations:

1. Lack of gateway awareness: All available gateways forward RREQs simultaneously, causing redundant transmissions.
2. Inefficient flooding: The broadcast-based discovery mechanism leads to excessive control traffic and channel congestion.
3. Propagation delay mismatch: Large timing differences between radio and acoustic links cause outdated route information.
4. Absence of context-aware routing mechanisms: The classical AODV protocol lacks any integration of geo-based or signal-quality-based decision criteria, which are essential for reliable routing within the underwater domain.

## 2.4 Simulation Tools for Multi-Domain Ad-Hoc Networks

The development and evaluation of Ad-Hoc communication protocols usually requires the use of realistic simulation environments. Numerous simulation frameworks have been developed for modeling terrestrial networks or underwater hydroacoustic communication. However, simulating Multi-Domain Ad-Hoc Networks that combine aerial, surface and underwater nodes poses specific challenges due to the heterogeneous nature of the communication media involved. The following overview in Table 6 summarizes the capabilities and limitations of the simulation tools commonly used for Ad-Hoc and underwater Networks.

Table 6: Overview of current simulation frameworks. The aim is to determine whether an existing simulation framework can be used to validate the MDO routing scenario.

| **Simulator & Domain** | **Underwater Support** | **Cross-Domain** | **Flexibility** |
|---|---|---|---|
| **NS-3 /NS-2:** Native terrestrial; can be extended to underwater via external modules (Kurniawan et al., 2020), (Ching et al., 2021), (H. Zhang & Guo, 2017). | Proprietary extensions: Optical underwater communication (UOWC) (Yan et al., 2021), AquaSim in NS-2 in (Hakim et al., 2018) and in NS-3 in (Martin et al., 2017). | Proprietary ARCNet implementation in (Jiang et al., 2023) (NS-3) and in (Ding et al., 2021). Both extensions not available as open source and | Widely used simulation tool, but the extensions for underwater are not reusable due to a lack of public access. Physical models are |

| | | | |
|---|---|---|---|
| | SUNSET extension designed for simulation and real-world deployment (Petrioli et al., 2015). | and not well described. | hardly or not at all described.<br><br>Flexibility: Medium. C++; modular architecture, but limited native support for underwater or hybrid domains. |
| **MATLAB:** Neither terrestrial nor underwater. The simulation environment is customized, especially for energy optimization (Jubair et al., 2019) and network performance (Kumar & Padmavathy, 2020) in terrestrial environments. | Proprietary development for analysis of SOAM protocol for underwater nodes, physical modeling approaches (Hyder et al., 2023), security and reliability approaches for AUV swarm communication (Zhu et al., 2023). | Previous approaches for Cross-Domain developed within the framework of this work (Brandl et al., 2024), (Brandl et al., 2025). | Although full flexibility enables the development of a customized simulation framework tailored to specific needs, OSI layers must be implemented manually and physical models require careful parameterization.<br><br>Flexibility High. Fully customizable, modular and adaptable to diverse routing and domain scenarios. Allows data export and integration with other tools for visualization or evaluation. |
| **OMNeT++:** Primarily designed for terrestrial networks, especially vehicular Ad-Hoc Networks (VANETs) (Elias et al., 2017) and for analyzing the network load in MANETs . Extensions for underwater communication are available**.** | Proprietary development of a physical model for a optimization of AODV through double flooding mechanism (Tong et al., 2016), and security analyses of underwater networks (Shi et al., 2024). | No known approach | Integrating an underwater domain is challenging because the physical-layer characteristics must align with the upper OSI layers. This alignment is not straightforward within the existing OMNeT++ framework.<br><br>Flexibility: Medium. There is a strong focus on VANETs. Some extensions for underwater communication exist, but none of them are openly available or maintained as open source. |

| | | | |
|---|---|---|---|
| **UnetStack:** Agent-based, configurable framework for software-defined underwater networks. It enables the rapid design, simulation, testing and deployment of these networks. | A detailed investigation of a suitable MAC protocol for underwater Ad-Hoc Networks (Ansa Shermin et al., 2024). In addition, the manufacturer of UnetStack offers a complete manual for the development of underwater networks (Chitre, 2020). | No known approach | Not suitable for our use case as it lacks support for above-water communication and Cross-Domain integration.<br><br>Flexibility: Excellent for detailed underwater network analysis and has open community support. However, it lacks Multi-Domain capabilities and is not easily extensible. |

A comparison of well-established simulation environments, such as NS-3, MATLAB, OMNeT++, and UnetStack, reveals that none fully meet the requirements for simulating realistic hybrid Ad-Hoc Networks across terrestrial and underwater domains. NS-3 and OMNeT++ have modular architectures, but the available extensions for underwater communication are either proprietary, inaccessible, or inadequately documented. MATLAB offers maximum flexibility but has mainly focused on terrestrial networks thus far. Existing underwater adaptations are limited in scope and often tailored to specific scenarios. As a modern framework for underwater communication, UnetStack provides comprehensive modeling capabilities in that domain but lacks support for above-water communication and Cross-Domain integration. Due to these limitations, it seems necessary to develop a custom simulation framework that incorporates physical models of underwater communication. However, these models require careful parameterization to accurately reflect environmental conditions.

The following key aspects were missing or inadequately supported in the reviewed frameworks:

1. Lack of openly accessible and reusable underwater communication extensions.
2. No integrated support for hybrid or Cross-Domain scenarios involving above- and underwater nodes.
3. Missing or insufficiently documented physical-layer models, particularly with regard to hydroacoustic propagation.

4. No built-in mechanisms for Cross-Domain queue management or domain-aware message forwarding.
5. Limited options for exporting structured data and integrating it with external analysis or visualization tools.

## 2.5 Technological and Scientific Gaps

The development of robust, interoperable communication frameworks that can operate across underwater, surface and aerial domains is still in its early stages. While a number of domain-specific routing protocols and simulation tools exist, most are either domain-specific or lack practical applicability in real-world, Multi-Domain scenarios. The following gaps have been identified as central challenges that this work aims to address:

(a) Lack of Integrated Routing Protocols

Although existing routing protocols such as AODV offer well-established mechanisms for terrestrial or surface-based networks, they are not consistently extended across domains. The adapted AODV, which has been optimized in (Tong et al., 2016) for the underwater domain, is of exemplary relevance in this context. However, it is noteworthy that this optimization is exclusively applicable within the confines of the underwater domain. In (Jiang et al., 2023), AODV is used to enable opportunistic underwater nodes to send their data to a specific gateway. However, this does not enable the discovery route to continue within the underwater domain, which conflicts with FR2 and FR3. The focus is clearly on reaching the other domain as quickly as possible. However, these individual use cases cannot be scaled up to accommodate others. For routing protocols intended for Cross-Domain use, the fundamental question is whether they adhere to an IP-based standard. In (Goetz & Nissen, 2012), the protocol is designed so that message addressing takes place within the data packet. Although this saves 20 bytes for IPv4 and 40 bytes for IPv6, interoperability is lost through the IP header, which contradicts FR5. When a routing protocol such as AODV is applied across the boundary between surface and underwater communication, environmental factors can significantly interfere with the expected routing behavior. AODV is designed for networks where propagation delays are minimal and transmission times are short. In contrast, underwater hydroacoustic communication introduces substantial delays due to the slow speed of sound in water. These delays can reach several seconds over moderate distances and are further affected by variable channel conditions. Consequently, key control messages such as Route Replies frequently arrive later

than anticipated and are disregarded by the protocol, despite the fact that the corresponding route would still be valid. This leads to repeated route discovery attempts, increased energy consumption and inefficient route setup. Studies have shown that establishing a reliable route in an underwater environment requires consideration of both maximum propagation delay and packet transmission time. As standard AODV implementations do not account for these delay characteristics, they are unsuitable for Multi-Domain communication without further adaptation, which conflicts with NFR2 and NFR3 (X. Wang et al., 2023).

Another key limitation of traditional routing protocols in Multi-Domain environments is their lack of awareness of link quality, which is particularly important in underwater communication. Classical protocols such as AODV base their routing decisions primarily on Hop count, implicitly assuming that all links are equally reliable. However, this assumption is invalid in underwater domains, where hydroacoustic links are highly variable and susceptible to noise, distance-related attenuation and environmental disturbances (Petroccia et al., 2018). Recent research has demonstrated the importance of signal quality metrics, such as SNR, in informing routing decisions in underwater hydroacoustic networks, which relates to FR2 and FR3. Cross-layer strategies have been proposed in particular to map effective SNR values to link adaptation mechanisms, thereby improving packet delivery and energy efficiency under varying channel conditions (I. Ahmad & Chang, 2019).

While these approaches significantly improve the robustness of routing within the underwater domain, they are confined to Intra-Domain scenarios. In Cross-Domain networks, however, the decision to forward data from aerial or surface nodes into the underwater domain requires prior knowledge of underwater link quality. This means that SNR-aware metrics and environmental awareness must exist not only locally within the underwater nodes, but also be accessible or anticipated by the over-water nodes. Without this integration, routing decisions across domains are unable to recognize the true quality and viability of the underwater path, which often results in unstable routes or failed transmission. Similar to SNR-based strategies, geographic routing mechanisms have been proposed for both the underwater and over-water domains. These approaches exploit the positional awareness of nodes to optimize packet forwarding, often relying on local neighbor information and destination coordinates (Coutinho et al., 2016). However, such methods remain proprietary and are not integrated into widely adopted standard protocols,

such as AODV. Furthermore, they are usually confined to a single domain and do not consider Cross-Domain position information. Consequently, routing decisions across domain boundaries, such as selecting the most suitable gateway node for underwater-to-surface communication, are made without considering spatial topology or Inter-Domain node locations, which conflicts with FR2, FR3, and NFR3. This limits the efficiency and adaptability of the overall routing strategy.

(b) Lack of Gateway Awareness in Routing Protocols

Extensive research has already been conducted on the integration of Relay nodes at the interface between the above-water and underwater communication domains. These cross-medium Relay are categorized into two main types: Static Relay, such as fixed-position buoys, and mobile Relay, such as autonomous surface vehicles. They serve as intermediaries between physically incompatible transmission media, such as radio frequency, hydroacoustic or optical communication, and thus form the basis for exchanging data across domains. However, recent studies have highlighted several limitations of Relay-based systems. The main challenges include long communication times due to multiple medium transitions, the high energy consumption of Relay nodes, increased latency caused by additional transmission Hops and strong dependency on Relay nodes. The failure of a single node can lead to a complete breakdown of cross-medium communication. These limitations demonstrate that, although physical interface technologies exist, integrating them into adaptive, routing-capable networks remains a significant challenge. In particular, routing protocols lack the ability to actively detect, localize and incorporate such Relay nodes and their characteristics into the route discovery process, which conflicts with FR3 and NFR4 (Azad et al., 2014). Recent research approaches have explored the integration of space and marine robotics, highlighting the potential of Relay nodes equipped with self-organizing network capabilities. However, these studies do not address the integration of such nodes into existing routing protocols or propose mechanisms for their dynamic discovery or selection (Yu et al., 2024). The publication (Ding et al., 2021) introduces a gateway authority mechanism. This approach enables specific nodes, particularly USVs, to temporarily assume the role of a cluster head and assist in forwarding data across cluster boundaries. The mechanism is based on the reception of cluster head confirmation packets, which grant forwarding privileges and enable a node to belong to multiple clusters simultaneously. While this method offers some degree of cross-cluster coordination, it has several fundamental limitations. Firstly, gateway node selection is not

based on dynamic network parameters such as current position, signal quality or energy level. Secondly, there is no mechanism to evaluate or compare multiple Relay nodes to select the most appropriate one. Furthermore, the entire concept is closely tied to a hierarchical clustering structure and is not embedded within the routing protocol layer itself. Consequently, the system lacks true gateway awareness at the routing level and is therefore not well suited to adaptive, scalable communication in dynamic environments, which contradicts FR3 and NFR3.

(c) Lack of a Suitable Simulation Framework for Multi-Domain Communication

A systematic review of existing simulation environments reveals a clear scientific gap. Although there are numerous tools for terrestrial and underwater communication, there is no open, modular framework that supports integrated, Multi-Domain Ad-Hoc Networking across both. Extensions for underwater communication are often proprietary, use-case specific, or lack sufficient documentation and validation. Furthermore, no available framework natively supports mechanisms for transitioning between domains, such as queuing strategies or time-synchronized handovers between radio-based and hydroacoustic communication layers. This lack of Cross-Domain interoperability and flexibility significantly limits the ability to simulate heterogeneous networks with interacting UAVs, USVs, and UUVs. Consequently, it remains challenging to investigate and evaluate new routing strategies or architectures under realistic operational and physical constraints within a unified simulation environment. A dedicated simulation environment designed specifically for this work is required to comprehensively develop and validate the requirements defined in FR1–FR5 and NFR1–NFR4.

The following Table 7 summarizes key missing capabilities in current approaches based on the identified gaps regarding the lack of integrated routing protocols and the absence of gateway awareness. This work aims to address these limitations by developing a solution that enables adaptive, Cross-Domain communication through enhanced routing mechanisms.

Table 7: Summary of current gaps in terms of available routing protocols and simulation frameworks.

| Category | Identified Gap | Implication | Related Requirement |
|---|---|---|---|
| **Integrated Routing Protocols** | There is no standard protocol for Cross-Domain communication. No protocol is adapted for seamless Cross-Domain operation. Neither from above to underwater nor from underwater to above. The reason is the physical characteristics of both domains. | Inefficient or failed route discovery across media boundaries. Alternatively, there are opportunistic protocols that only work when a scenario is static and it is possible to reach a node in the other domain. | FR2, FR3, NFR 3 |
| **Propagation Delay Handling** | Existing protocols, such as AODV, do not account for high and variable delays in underwater hydroacoustic. This is because AODV-RFC (Perkins et al., 2003) does not include manual timers for longer propagation times. | Route Replies may be discarded prematurely. Approaches with constant confirmation messages, such as *hello* messages, only work in theory because signal overlaps and Doppler effects are ignored. | FR2, NFR2, NFR4 |
| **Link Quality Awareness / Geographical Awareness** | Routing decisions that transition from over-water to underwater do not consider underwater-specific factors, such as the signal-to-noise ratio. | This causes unstable routes and poor performance in underwater communication. The signal quality can only be predicted based on the geographical location of a destination node. However, no suitable approach is found for the geographical anticipation of target nodes. | FR2, NFR3 |
| **Relay Node Integration** | Relay nodes are not integrated into routing logic, but treated as fixed infrastructure. In existing approaches, these are assumed to be existing nodes. In reality, however, | Relay failures disrupt communication due to a lack of redundancy or adaptability. The underwater canal is unnecessarily filled with messages that | FR3, NFR1, NFR3, NFR4 |

| | | | |
|---|---|---|---|
| | they can also be a Relay on the water surface at one time and submerged at another (like the AUV). In addition, not every USV has a gateway function at all times. | will probably never reach the destination node. | |
| **Simulation Framework** | No Open-Source, modular simulation environment that supports realistic Cross-Domain communication between terrestrial and underwater networks. | Existing tools cannot be used to systematically analyze, compare, or validate hybrid network architectures. | FR1-5, NFR1-4 |

# 3 Design and Routing Logic of the extended Ad-Hoc Protocol

This chapter aims to develop and analyse routing models that enable efficient communication across diverse maritime domains. The focus is on optimizing route discovery and forwarding behavior in networks where radio and acoustic communication coexist. The hybrid AODV model is used as a reference implementation for benchmarking purposes, while the extended AODV variants are designed to enhance routing performance by incorporating domain awareness and selective gateway utilization. These extensions aim to establish an adaptive routing mechanism that can make context-aware decisions based on factors such as geographic location, link quality and communication domain. The extended protocol prioritizes efficient gateway selection and minimises unnecessary transmissions in the underwater channel to reduce channel load, propagation delay and packet loss while maintaining reliable connectivity across domains. Building upon the classical AODV structure, several routing models were developed to progressively incorporate these features. The hybrid AODV serves as an adapted baseline implementation that integrates separate queues for radio and acoustic transmission. This enables it to operate across multiple domains without requiring domain-specific optimization. Subsequent extended AODV variants build on this by introducing internal and external logic for domain-aware route selection, gateway control and adaptive forwarding. Additionally, a dedicated error management module improves route recovery and stability in varying propagation conditions. Table 8 provides an overview of all the routing models developed in this work, showing their routing direction and specific purpose within the overall framework.

Table 8: Description of the developed reference model of the hybrid AODV, as well as the developed extended AODVs and error management.

| **Routing Protocol** | **Routing Direction** | **Description** | **Purpose of this Work** |
|---|---|---|---|
| **Hybrid AODV (Reference Model)** | UAV to UAV, UAV to USV and **UAV to UUV (simulated)** | Adapted version of the classical AODV protocol with separate queues for radio and acoustic communication. No domain | Serves as the baseline reference model for performance benchmarking across domains. |

| | | | |
|---|---|---|---|
| | | awareness or gateway selection logic is applied. | |
| **Extended AODV (Internal Logic)** | UAV to UAV, UAV to USV and **UAV to UUV (simulated)** | Uses a geo-based routing approach. USVs act as potential gateways and decide locally whether to forward a request into the underwater domain based on the geographic proximity to the destination. | Enables selective gateway utilization and reduces unnecessary RREQ propagation in Cross-Domain routing. |
| **Extended AODV (External Logic)** | UAV to UAV, UAV to USV and **UAV to UUV (simulated)** | Applies a signal-quality-based routing approach. Among available USVs, the one providing the strongest link quality is selected as the active gateway for underwater transmission. | Introduces domain-aware route establishment based on real-time link quality and enhances underwater transmission efficiency. |
| **Extended AODV (Subsea Routing)** | **UUV to UUV (simulated)**, UUV to USV and **UUV to UAV (simulated)** | Allows both acoustic and radio transmission for upward or horizontal routing. Based on the internal logic, but optimized for underwater-originated communication. | Supports efficient upward data transmission and ensures robust connectivity from underwater nodes to underwater, surface or aerial assets. |
| **Extended AODV (Error Management)** | **All Directions are simulated** | Restores routes when a unicast message is lost due to link failure or node outage. Detects missing acknowledgments and triggers localized route recovery. | Increases fault tolerance and maintains network stability under dynamic maritime conditions. |

In all versions of the extended AODV protocol, the radio and acoustic domains are treated as strictly separate communication environments. Routing and data messages, such as RREQ, RREP and DATA, remain within their respective domain until a domain transition flag is explicitly set. This flag is managed at node level and updated when a message is received by a designated gateway. For example, when a USV acting as a gateway receives a DATA packet addressed to it as the target gateway, the flag changes from 0 to 1, prompting the packet to be transmitted into the hydroacoustic domain, illustrated in Figure 9. In

contrast, the hybrid AODV reference model does not enforce this type of separation. While it keeps separate queues for radio and acoustic communications, messages can pass between domains at any time. This behavior reflects the original AODV design, which does not differentiate between physical media. Therefore, the hybrid AODV reference model serves as the basis for evaluating the domain separation and selective forwarding mechanisms of the extended AODV versions.

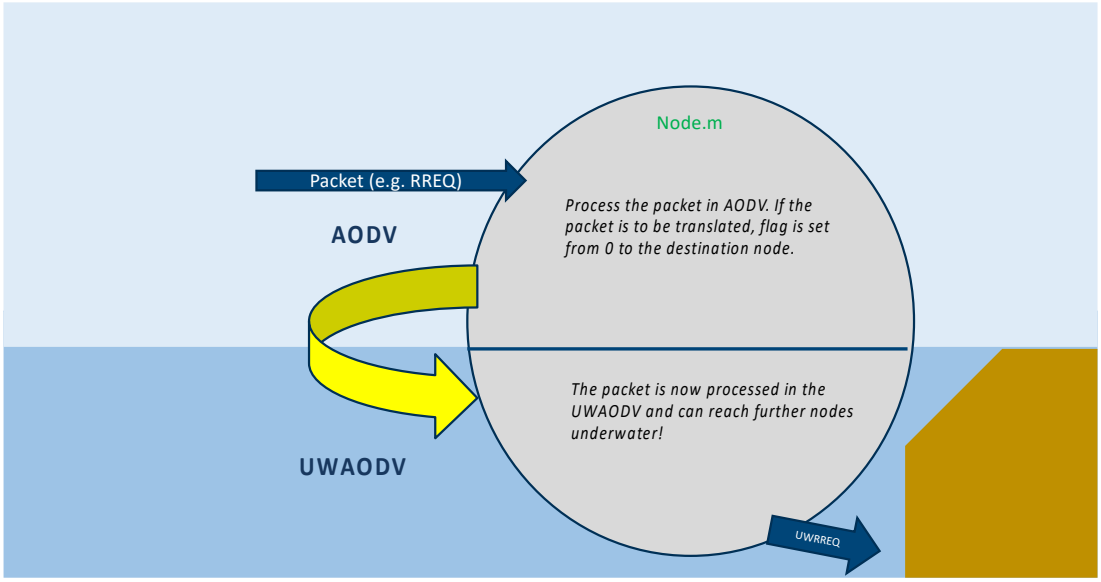


Figure 9: Illustration of the domain transition from radio to hydroacoustic communication on a USV depicted as a circular node. The change is made by setting the flag in a targeted manner.

To ensure transparent, structured representation of routing behavior, the routing logic is visualized with flowcharts. The diagrams adhere to widely used flowchart conventions to ensure readability and clarity. Rounded rectangles mark the start or end of a process, and standard rectangles indicate individual processing steps, such as computations, data handling or message forwarding. Decision points, where the process flow can branch based on conditions or logical checks, are represented by triangles. Due to the complexity of the routing mechanisms, certain parts are abstracted into green modular boxes, such as "*check_m*". These modules are defined once and reused across different flowcharts to reduce redundancy. Each step is assigned a unique identifier ranging from A1 to E19. This enables consistent references throughout the chapter and supports traceability between the description and visualization. Section 3.1 begins with the hybrid AODV

protocol as a reference protocol. Subsequent sections introduce proposed extensions that support communication between the UxVs.

## 3.1 Setting up a Hybrid AODV Routing Algorithm as a Reference Model

Figure 10 illustrates a flowchart that provides a visual representation of the hybrid AODV route discovery and data forwarding process. It illustrates the interaction between four example nodes. The first node is a sender representing a UAV. The first Hop denotes a USV without gateway functionality. The second Hop is represented by a second USV with gateway functionality, and the destination node represents a UUV. The diagram shows the sequence of control messages and routing steps required to establish a route from the UAV to the UUV. In this application, the UAV as the sending node is within communication range of Hop 1, which represents a USV. Hop 1 is within communication range of Hop 2, which also represents a USV, and the UUV is connected to Hop 2 via a hydroacoustic link, with gateway functionality. The flowchart serves as the basis for explaining the protocol logic in detail in the following sections. The process begins with the UAV. The sender initiates the route discovery procedure, shown in step A1. In step A2, the UAV generates and broadcasts a RREQ message. The RREQ contains several fields, including the destination node's address, a unique request identifier, the source node's address, and a Hop count initially set to zero. The message structures adhere to the definitions outlined in RFC 3561 (Perkins et al., 2003) for IPv4-based AODV. The RREQ message consists of 23 bytes, while the RREP message contains 20 bytes.

The RREQ is broadcast by the UAV and received by the Hop 1 (USV) which lacks gateway functionality. Step A9 involves checking whether the RREQ is received correctly. The message cache is examined in step A10 to see if the RREQ has already been processed. Next the routing tables are checked (A11) to verify if an entry for the source node already exists or needs updating. If the message is valid and not a duplicate, the process continues. Otherwise, the RREQ is discarded. In step A12, the node checks if it is the final destination. In the event that this is the case, the node responds with a RREP message in Step A13. This message, referred to as a RREP, confirms the existence of a valid route and is sent back along the reverse path to the originator. The RREP message structure follows the definitions in RFC 3561 (Perkins et al., 2003).

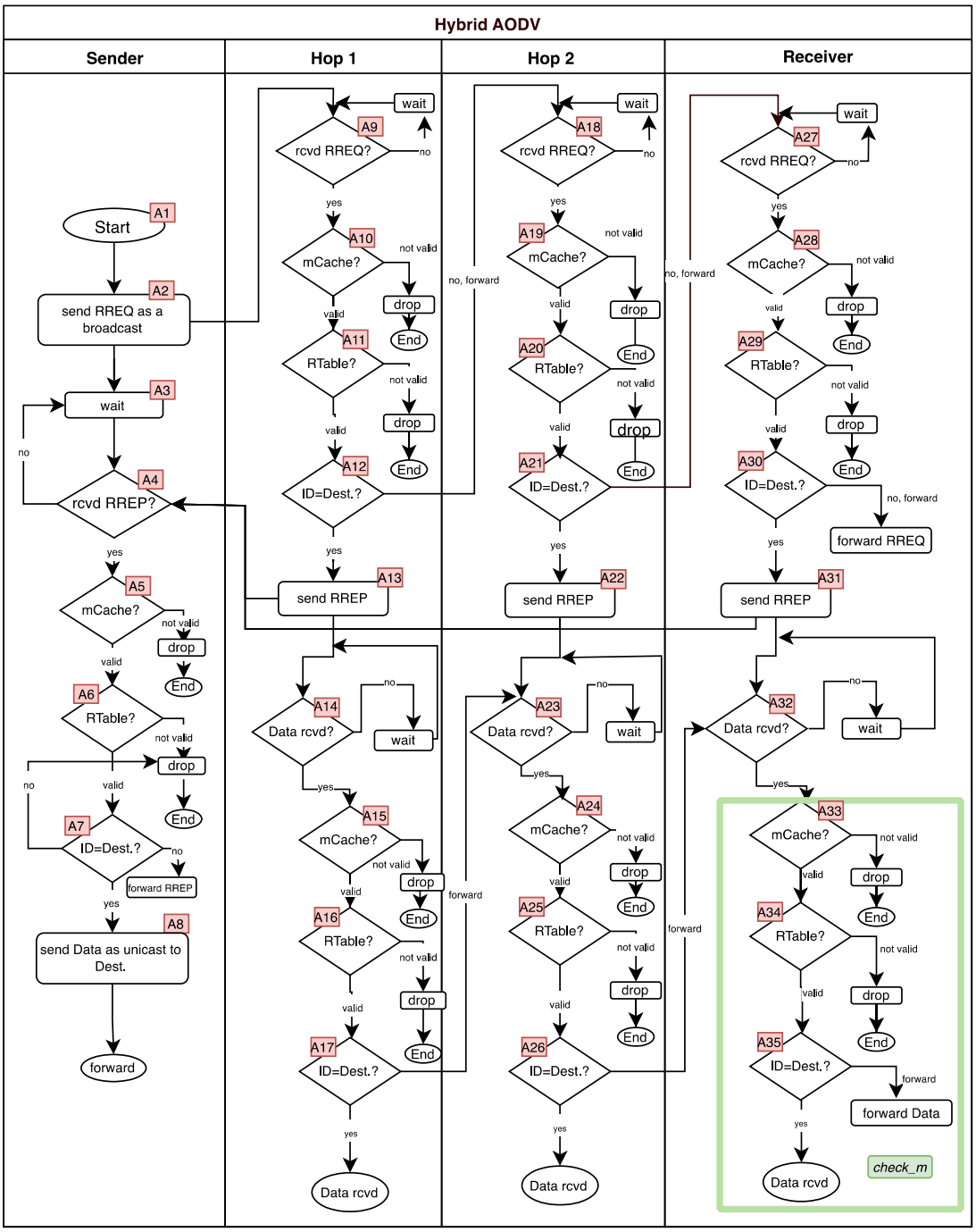


Figure 10: Flowchart of the AODV reference protocol with the assets of a transmitter (UAV), two Hops (USVs), and a UUV as receiver.

If the message has not yet reached its final destination, it is broadcast further and the TTL is reduced by 1 with each Hop. The second USV, referred to as Hop 2, receives the RREQ and performs the same validation steps. This includes checking the message cache and the routing table (steps A18 to A22). If the destination has not been reached, the USV converts the message from radio to hydroacoustic transmission and forwards it to the

UUV underwater. Upon arrival at the UUV, the RREQ undergoes another round of validation, as shown in steps A27 to A30.

If the destination has been reached, the UUV generates a RREP message in step A31. The RREP is then transmitted as a unicast message back to the sender via the intermediate node's routing tables. During this period, the sender UAV remains in a waiting state, as defined in step A3. Once the RREP reaches the sender, its validity is checked in steps A4 to A6. If the reply is valid, the sender transmits a data packet via unicast in step A8. The packet is then forwarded through the network based on the entries in the routing tables until it reaches the UUV. Upon arrival at the destination, the UUV initially verifies whether the data packet is received successfully (A32), then validates its contents in steps A33 to A35. If these checks are passed, the process is complete and the data transmission is considered successful. Steps A33 to A35 are summarized here in the green box as *check_m*. This is because the data packet is frequently checked in further illustrations and can then be abbreviated to *check_m*.

## 3.2 Data Routing from UAV to UUV (Internal Routing Logic)

The internal routing logic is based on the hybrid AODV protocol, but it introduces additional message types to support Cross-Domain communication. The overarching goal of the internal logic is to anticipate the most suitable gateway for the conversion of radio to hydroacoustic signals using a geographically based routing, thereby reducing the time required for route establishment underwater in accordance with NFR3. This routing approach assumes that the sender node is aware of the UUV's mission plan. It is distributed to the relevant platforms prior to the mission. Therefore, this extended routing logic eliminates the need for route discovery in the underwater domain. As shown in Figure 11, with a standard AODV setup, both the RREQ and the RREP must be transmitted hydroacoustically if the destination is a submerged node. This process can introduce delays exceeding more than one second. In contrast, the proposed logic ensures that only the final data packet containing the actual payload is sent underwater.

This optimization is critical because it addresses one of the core technical challenges, the significant latency of hydroacoustic communication. Since hydroacoustic signals propagate at only 1405–1560 m/s (Goetz & Nissen, 2012), whereas radio waves travel nearly

at the speed of light (~$3 \cdot 10^8$ m/s), underwater route discovery would be prohibitively slow and must therefore be performed above the surface whenever possible.

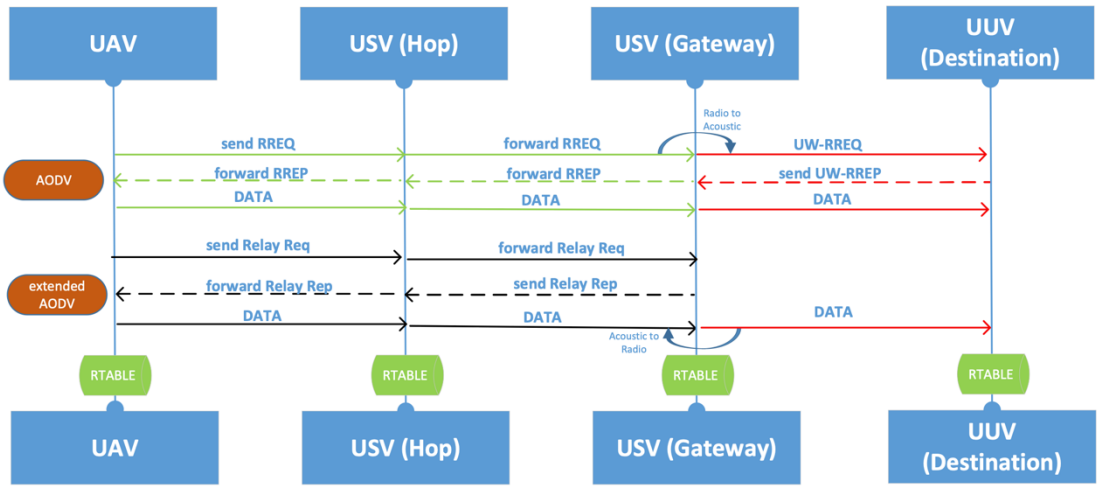


Figure 11: Sequential representation of the extended AODV of the internal routing logic. It shows that route establishment is completely omitted in the underwater domain and that the data packet is sent.

The process begins at node B1 when the sender initiates a transmission by broadcasting a RREQ. After broadcasting, the sender enters a 30 ms waiting interval. This short delay is intentionally configured to allow for the possibility of receiving a standard RREP from an over-water destination node, such as a USV or surfaced UUV. If such a node is within direct radio range and reachable via the hybrid AODV mechanism, the routing process proceeds according to standard AODV steps (see states A5 to A8 in Figure 10). If no RREP is received within the 30 ms waiting period, the timer is cleared and a new message type, the Relay Request (RelReq), is introduced (see state B8). This message closely resembles a standard RREQ but includes an additional flag indicating that it should be answered only by USV nodes. The RelReq is then forwarded to the first Relay node (Hop 1), where it undergoes a validity check (states B16 to B18). Unlike in the hybrid AODV, state B18 no longer verifies whether the node is the final destination. Instead, it verifies whether the node is a USV (B19). If the receiving node is a USV, it responds with a Relay Reply (RelRep) message. This message is structurally similar to a standard RREP, but it includes the USV's current GPS position (Figure 13). The RelReq is answered and forwarded by the USV, allowing other, more distant USVs to receive it.

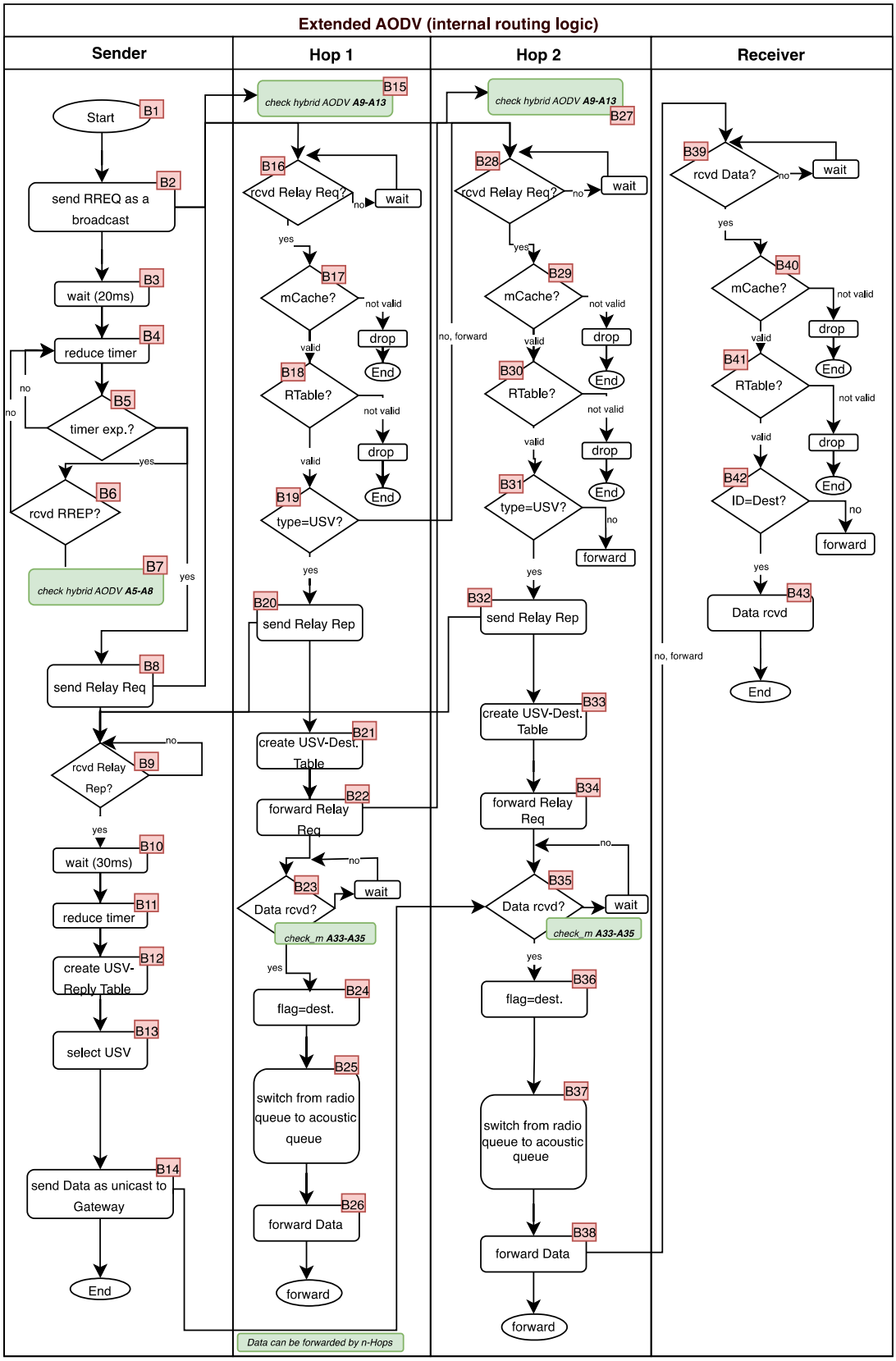


Figure 12: Flowchart of the extended AODV of the internal routing logic. This is based on the hybrid AODV, but has been conceptually expanded by the RelReq and RelRep messages. The aim is to reduce the hydroacoustic channel.

When a forwarded RelReq reaches the next USV (Hop 2), it undergoes validation again. This includes checking the message cache, looking up the routing table, and verifying that the node type is a USV (see states B26 to B31).

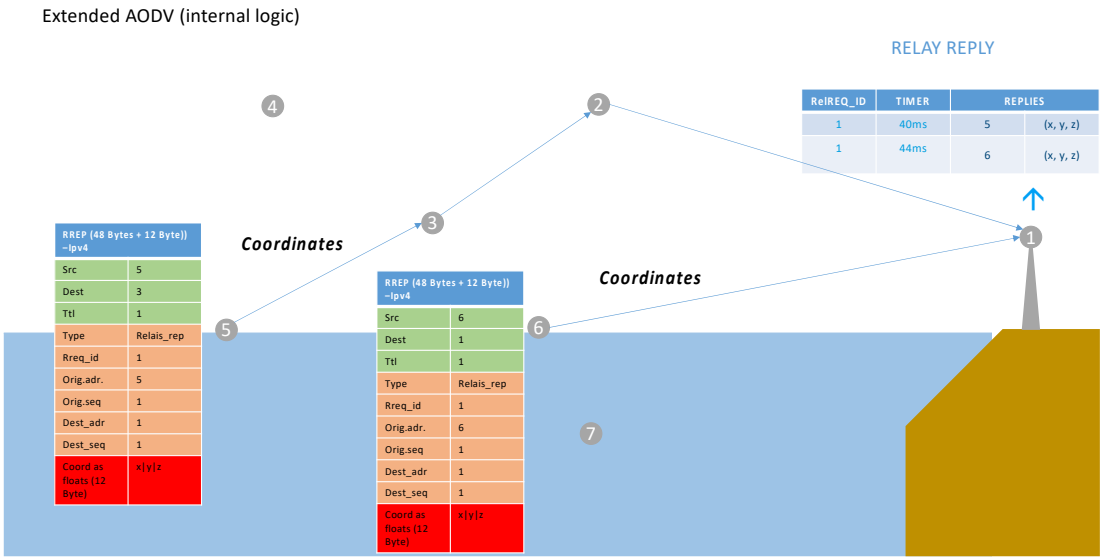


Figure 13: Illustration of the extended AODV internal routing logic. The coordinates of the USVs are sent back to the sender using the RelRep message type. The sender makes a routine decision based on the mission plan and the positions of the USVs.

If the USV type is valid, it also responds with a RelRep. Upon receiving a RelReq, each USV creates a local data structure called the USV Destination Table (Table 9), regardless of whether it is the first or second Hop.

Table 9: Illustration of the USV Dest Table. This can consist of just one line and has the task of storing the destination at the gateway (USV) where the data packet is to be sent. The data packet is sent specifically to the selected USV and must then receive information about where it is to be sent when switching from radio to hydroacoustic domain.

| Relay Request ID | Destination | Source ID |
|---|---|---|
| 1 | 7 | 1 |

This table temporarily stores the final destination address, which describes the address of the target UUV. This is necessary because the subsequent data packet is addressed only to the selected USV, which acts as a gateway. Without this information, the USV would not know where to forward the data once it is received. Therefore, the UUV destination address is buffered at the USV for later forwarding. The RelRep is sent either directly

back to the original sender, if within communication range, or forwarded over multiple Hops using the decentralized routing tables established in previous steps. Meanwhile, the sender waits for another 30 ms (see state B10). During this time, the sender collects all incoming RelReps and stores their contents in a local USV Reply Table (see state B12). This table is used to select the most suitable USV to act as a gateway for the final data transmission. The sender node keeps track of the estimated coordinates of all target UUVs at specific times. After receiving and storing multiple RelReps in the USV Reply Table, the sender evaluates each USV based on its geographic proximity to the final UUV destination.

For each Relay Reply, the sender calculates the Euclidean distance between the USV position $(x_{\mathrm{USV}}, y_{\mathrm{USV}}, z_{\mathrm{USV}})$ and the UUV position $(x_{\mathrm{UUV}}, y_{\mathrm{UUV}}, z_{\mathrm{UUV}})$ is computed using the following Formula 3.1:

$$d = \sqrt{(x_{\mathrm{USV}} - x_{\mathrm{UUV}})^2 + (y_{\mathrm{USV}} - y_{\mathrm{UUV}})^2 + (z_{\mathrm{USV}} - z_{\mathrm{UUV}})^2}. \tag{3.1}$$

The USV with the shortest calculated distance is selected as the optimal Relay node. This node will serve as the destination node for the next data transmission. After selecting the optimal USV based on the minimum Euclidean distance, the sender transmits the data packet with the routing flag set to 0. This packet is not addressed to the final UUV destination but rather to the selected USV (see state B14). The packet is then forwarded through the network using standard AODV mechanisms. At each Hop, the message undergoes validity checks corresponding to the standard procedure (check_m, states A33 to A35, Figure 10). If the current node is not the selected USV, the message is discarded and forwarded until the TTL value expires. When the designated USV receives the data packet (see State B35), the routing process shifts from radio-based to hydroacoustic communication. The packet is inserted into the hydroacoustic transmission queue and the routing flag is updated. Specifically, the destination address stored in the USV Destination Table at state B33 earlier is retrieved, and the packet is re-addressed to the final UUV. Subsequently, the message is transmitted via the underwater hydroacoustic link. Upon reception (state B39), the UUV performs standard validity checks (states B40 to B42). If the packet passes all checks, it is accepted successfully (state B43), completing the Cross-Domain transmission process.

## 3.3 Data Routing from UAV to UUV (External Routing Logic)

To address the limitations of conventional routing approaches in dynamic, infrastructure-less underwater environments, external routing logic has been introduced. An external logic is optional for the internal logic, which is based on the approach that the UUV's destination data are known to the sender node. This can be assumed when using proprietary platforms. However, when operating in conjunction with different partners, an extension, based on the premise that the UUVs regularly communicate their location, is needed. This is achieved in this approach through signal quality-based routing. The routing mechanism extends the existing protocol architecture by enabling decentralized gateway selection based on periodically transmitted Relay Echo messages. Before explaining the flowchart in Figure 12 in more detail, it is first necessary to explain the Relay Echo message in detail, as this is a central component of this routing logic. With the help of the Relay Echo message, it is possible to use the extended AODV even if the target coordinates of the UUV are not always known. Although generating data traffic underwater is necessary for this approach, a message timer based on the Poisson equation to minimize it, is proposed here.

### 3.3.1 Concept of the Relay Echo Messages

The present routing logic utilizes a modified version of the extended AODV protocol for internal routing operations. However, this system is predicated on the assumption of prior knowledge of the mission plan of the target vehicle in the underwater domain. In dynamic and heterogeneous missions, such an assumption is often unrealistic, thereby limiting the flexibility and efficiency of communication processes. Consequently, the ability to effectively protect a wide range of critical infrastructures may be compromised. The external routing logic has been developed to address this gap by introducing a routing approach that enhances adaptability in heterogeneous mission scenarios. The integration of the extended routing mechanisms is expected to enhance the system's overall network responsiveness and robustness in Multi-Domain environments. However, it is also to be expected that underwater data traffic cannot be avoided. The objective of the external routing logic is to develop a node selection concept based on the *Hello* messages defined in the AODV protocol (Perkins et al., 2003) and to integrate it into the existing MATLAB simulation framework. This extension, termed Relay Echo, is designed to facilitate the

utilization of essential status information to inform routing decisions. The implementation of such a more efficient routing mechanism has the potential to significantly improve communication between network participants, thereby enhancing the overall system performance under realistic mission conditions. In the AODV RFC, *Hello* messages are intended to maintain active links in the local network topology by periodically refreshing routing table entries (Perkins et al., 2003). In the context of the Multi-Domain extended AODV, the functionality of *Hello* messages is adapted to support node selection and status reporting across heterogeneous domains as part of the Relay Echo mechanism. The absence of predefined mission data has a significant impact on the availability of internal routing logic for underwater vehicles. Instead, Relay Echo messages are periodically transmitted by UUVs to proximate gateways within communication range, illustrated Figure 14. These messages are stored in a dedicated data structure at the gateway, referred to as the Relay Table. The entries in this table are temporary and contain relevant metadata used for routing decisions. Upon receipt of a RelReq, the gateway performs a search of the Relay Table for a matching destination entry. If such an entry is found, the gateway responds with a RelRep. The integration of the extended routing mechanisms is expected to enhance the system's overall network responsiveness and robustness in Multi-Domain environments. However, it is also to be expected that underwater data traffic cannot be avoided. Even more, it should be noted that the hydroacoustic channel becomes increasingly busy as the number of participating nodes increases due to the repetitive nature of Relay Echo transmissions.

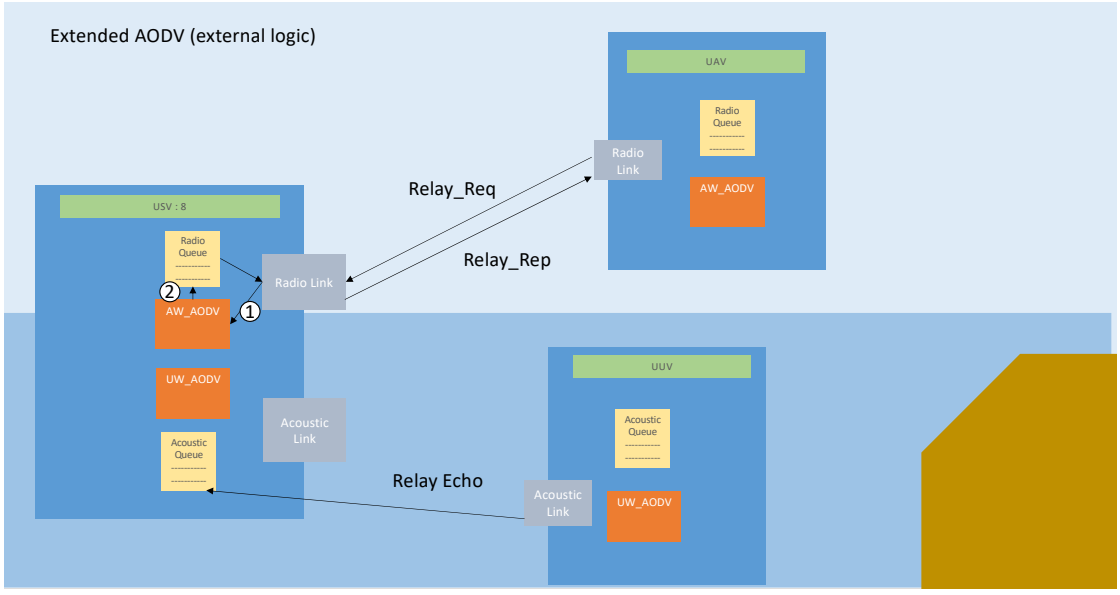


Figure 14: An example of the Relay Echo. A UUV sends a message to a USV, which stores the Relay Echo in a table.

### 3.3.2 Usage of the Relay Echo for Routing

When the Relay Echo function is activated, underwater vehicles periodically transmit messages at predefined intervals. The activation of Relay Echo can be triggered by events such as a UUV adhering to a fixed mission plan, but then drifting off course. This can occur due to strong lateral currents. As a result, the internal routing logic of the transmitter node would probably select the wrong gateway to send data. Nodes that function as Relay candidates store the Relay Echo messages they receive in dedicated data structures. This maintenance is done independently of any data transmission requests within the network. The initiation of routing is initiated when the Relay Table entries are accessed once a RelReq reaches the gateways via the above-water communication channel. The gateways then examine their tables for a matching entry corresponding to the requested destination node. In the event of such an entry existing, the gateway then computes a score, for example, based on the SNR of the most recently received Relay Echo. The gateway responds to the requesting node with a RelRep containing the score and its own address. In case of multiple RelReps being received, the node that made the request selects the gateway with the highest score. In the event of a tie, the selection is determined by the most recent timestamp among the highest scores. Once the gateway has been chosen, the route is established and the data transmission can begin. If there is no entry from SNR, the internal routing logic is used. In this particular instance, a SNR value of 20 dB has been designated as the threshold value, a decision that has been made following consultation with the relevant specialist department. In the event that the SNR falls below this threshold, the gateway is not considered for routing.

### 3.3.3 Structure of the Message Type Relay Echo

This section describes the structure of the information transmitted via Relay Echo messages and the data recorded when creating entries in the corresponding Relay table. It is important to note that not all of the listed elements are mandatory. Parameters deemed essential for node identification, such as the unique ID of the underwater node, are required, while others are considered optional and may enhance routing performance or support additional future functionality. The inclusion of additional data within Relay Echo messages results in an increase in packet size, which, in turn, raises the channel load and the likelihood of collisions, particularly in bandwidth-limited underwater environments.

In this work, several information elements are embedded with the intention of enabling future routing optimizations or alternative strategies. Furthermore, the collected metadata could be utilized by surface-level applications, as the data stored in the Relay Tables is only accessed in the above-water domain. This design reduces the latency for accessing up-to-date information and may support more responsive decision-making. As illustrated in Table 10, the internal Relay Echo table generated on a USV Gateway node is presented as an exemplary extract. The function of this table is to store metric values received from a UUV via a Relay Echo message. These values are used to support an extended routing logic that goes beyond the traditional AODV mechanism. In addition to SNR, which currently serves as the primary routing metric, further parameters such as the unique node identifier (Node ID), a sequence number, a timestamp, the node's speed, its horizontal orientation (heading), and battery status are considered in the overall design. The encoding of each parameter is accomplished through the utilization of a predetermined bit-length format, with the resultant data being integrated within the framework of an IPv4-based message structure for transmission.

Table 10: Example Relay Echo table of a USV. The most important entry for the internal routing logic is the SNR value. Further parameters can be used for optimized routing.

| **Field name** | **Bits** | **Example value** | **Unit** |
|---|---|---|---|
| Node ID | 32 | 192.168.1.14 | Ipv4 Address |
| Sequence number | 16 | 18 | - |
| Unix Timestamp | 64 | 32 | 1713427200 |
| SNR UUV | 8 | 23 | - |
| Speed | 8 | 8 | m/s |
| Battery | 8 | 32 | % |
| Heading (horizontal) | 16 | 42.4 | Degree |
| Heading (vertical) | 16 | -15.6 | Degree |

The Node ID is stored as a 32-bit field to allow the sending node to be identified unambiguously. The sequence number is encoded using 16 bits, representing a deliberate reduction from the 32-bit sequence number defined in the original AODV specification to reduce message size. The timestamp is implemented as a 64-bit Unix time value, providing high resolution and a long-term temporal reference. The SNR is transmitted as an 8-

bit integer in decibels. The node's speed is encoded using 8 bits, assuming a fixed-point representation with a resolution of 1 meter per second. The battery state is also represented using 8 bits. The horizontal and vertical headings are encoded as 16-bit fixed-point values with a precision of 0.1°. Based on these fields, the total message size is 168 bits (21 bytes) per entry, not including protocol headers or additional overhead. For demonstration purposes, the MATLAB implementation uses a reduced Relay-Echo format. Only a subset of these fields is transmitted, resulting in a significantly smaller payload of 8 bytes per message.

Calculation of the SNR metric

To evaluate the quality of hydroacoustic signals at the gateway nodes, the SNR is used, a key metric in underwater hydroacoustic communication for assessing reception quality. The hydroacoustic modems employed, such as the Evologics S2C series, can measure the received signal strength indicator (RSSI). RSSI is usually determined during reception and reported in the acknowledgment (ACK) packet. The RSSI measurement is based on detecting the maximum amplitude of the incoming signal via the hydrophone and is expressed in decibels relative to 1 µPa. In addition to RSSI, estimating SNR requires knowledge of the ambient noise level (NL), which the modem also captures using the hydrophone during idle listening periods. The SNR metric reflects the properties of the signal source, the influence of environmental noise, and transmission losses caused by spreading, absorption, and scattering. Thus, the SNR is used to quantify channel quality and support routing decisions at gateway nodes within the extended AODV protocol. The SNR is defined as follows (Ulrick, 2013):

$$\mathrm{SNR} = \mathrm{SL} - \mathrm{NL} - \mathrm{TL}, \tag{3.2}$$

where SL denotes the source level of the transmitted signal, NL denotes the ambient noise level, and TL denotes the transmission loss. All values are given in dB re 1 µPa. The implementation used in this work follows the formulation provided in the MathWorks documentation (The MathWorks, Inc., 2025). A higher SNR indicates a stronger signal relative to background noise, implying improved transmission quality. In addition to the standard evaluation based on the most recent SNR measurement, this work examined several alternative approaches. These include score-based methods, such as calculating the average SNR over time, incorporating stability measures like variance or standard deviation, and performing trend analysis by linearly extrapolating historical SNR values.

While these methods have theoretical advantages, such as smoothing short-term fluctuations or predicting future link quality, they present considerable challenges in the context of underwater hydroacoustic communication. First, these methods require the storage and processing of historical data, which increases computational load and memory usage. Second, the underwater channel is highly dynamic and nonlinear. This means that even minor environmental changes or obstacles can abruptly drop the SNR, making linear predictions unreliable. Furthermore, trend-based evaluations depend on sufficient past measurement density, which is often unavailable in sparse or energy-efficient networks. Lastly, predictions based on SNR trends may indicate improvements that occur too late to be relevant for real-time routing decisions. Due to these limitations, the more complex scoring and trend-based methods were not pursued further in this work. Gateway selection is therefore based solely on the most recent SNR measurement, providing a practical, resource-efficient solution that balances responsiveness with implementation simplicity.

### 3.3.4 Calculation of Possible Collision of the Relay Echo

The Relay Echo messages are transmitted via the common hydroacoustic channel, which means that there is a risk of collisions in the channel. A similar problem can be found in wireless networks of the data link layer, in particular with medium access control. In (Tanenbaum & Wetherall, 2011) a classical approach is described how in ALOHA and slotted ALOHA several network participants attempt to transmit simultaneously via a common transmission medium. Stochastic models such as the Poisson and exponential distribution are used to describe random transmission events and the time intervals between them in order to underestimate and mitigate this collision behavior (Haenggi, 2010). In this study, several UUVs want to access a common, non-coordinated medium. This makes it necessary to transfer established MAC principles and their mathematical modeling to hydroacoustic communication in underwater Ad-Hoc Networks in order to be able to probabilistically estimate and systematically avoid collisions. One suitable method for estimating the probability of packet collisions among multiple UUVs is to model the transmission of Relay-echo messages as a Poisson process. The probability of a collision depends on several parameters, including the number of active nodes, the length of the packet, and the physical transmission rate.

The following calculation is intended as an example for the transmission of 6 nodes at an example transmission interval of between:

$$T_{\text{Relay-Echo}} \in [10\,\text{s},\ 22\,\text{s}]. \tag{3.3}$$

This results in an average inter-arrival time of:

$$\overline{T_s} = \frac{T_{min}+T_{max}}{2} = \frac{10\,\text{s}+22\,\text{s}}{2} = 16\,\text{s}. \tag{3.4}$$

For our example of six UUVs, the total event rate of all participants is the product of individual events and the number of UUVs sending Relay Echoes. In the event that the SNR falls below a specific threshold, the gateway is not considered for routing:

$$\lambda_{\text{gesamt}} = 6 \cdot \frac{1}{\overline{T_s}} = \frac{6}{16\,\text{s}} = 0{,}375\,\text{s}^{-1}. \tag{3.5}$$

Furthermore, the critical time window $T_c$ in which a collision can occur is important for the analysis. This time window is made up of the transmission time $T_f$ of a data packet. This is made up of the packet length L and the transmission rate R:

$$T_f = \frac{L}{R}, \tag{3.6}$$

with an exemplary packet length of 248 bits and a transmission rate of 10 kbit/s, this results in a $T_f$ of 0.025 s. Added to this is the propagation delay ($T_{prop}$), which is dependent on the communication range $d_{max}$ and the propagation speed v:

$$T_{\text{prop}} = \frac{d}{v}. \tag{3.7}$$

In the example calculation, a range of 2000 m is assumed, and the speed of sound in water is set to 1500 m/s. This results in a $T_{prop}$ of 1.333 s. This results in the following critical time window in our case:

$$T_c = T_f + T_{\text{prop}} \approx 1.358\,\text{s}. \tag{3.8}$$

Within the time interval $T_c$, it is assumed that transmission events occur according to a Poisson process. The expected value of the events in $T_c$ corresponds to the product of the total event rate and the length of the time window. This can be described as follows:

$$\lambda = \lambda_{\text{gesamt}} \cdot T_c. \tag{3.9}$$

The Poisson distribution describes the probability that exactly k events occur in this interval:

$$P(k;\lambda) = \frac{(\lambda)^k \cdot e^{-\lambda}}{k!}. \tag{3.10}$$

A collision occurs if at least two transmission events occur in $T_c$. This calculates the collision probability $P_{collision}$ as:

$$P_{\text{collision}} = 1 - [P(0) + P(1)], \tag{3.11}$$

which means that, with six nodes and an average Relay Echo interval of 16 s, there is a collision probability of approximately 9.5% within the 1.36 s underwater critical time window. Figure 15 illustrates how the collision probability increases significantly with a higher number of nodes and smaller transmission intervals. Depending on the routing requirements, the criticality of the routing can be checked using these two parameters.

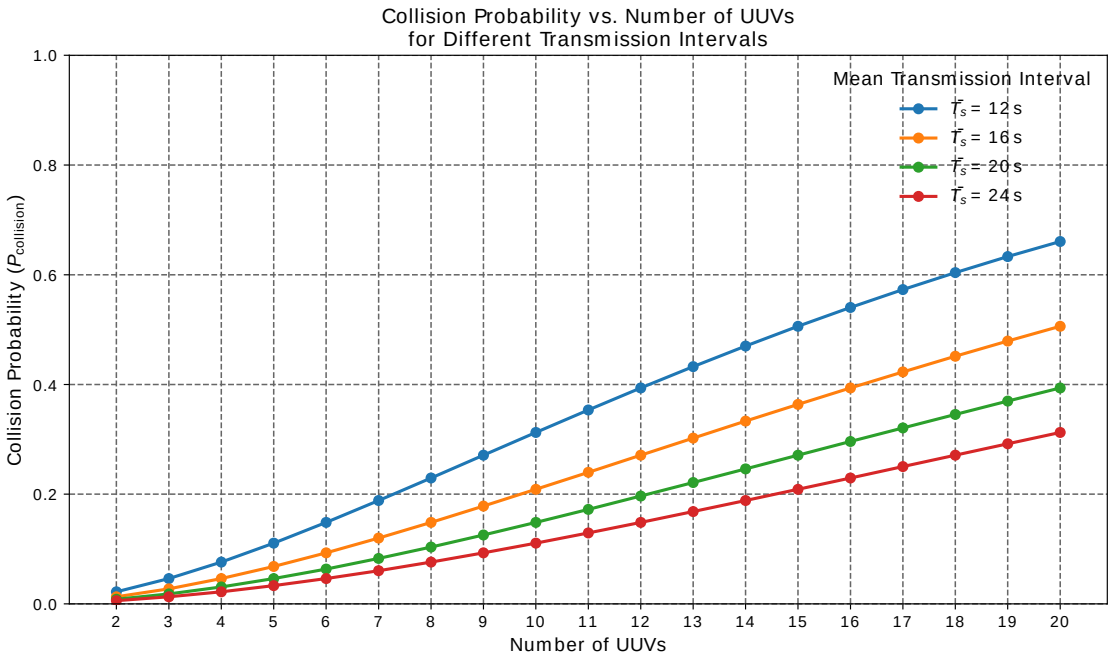


Figure 15: Collision probability as a function of the number of nodes (2–20) for different mean transmission intervals (12 s, 16 s, 20 s, and 24 s).

Figure 15 illustrates that a certain risk of collision always exists, even when the number of nodes is small and a large transmission interval is selected. The Poisson-based model provides a first-order approximation of this behavior under the assumption of statistically independent and memoryless transmissions. However, this assumption is a simplification of the actual system, as timing jitter, deterministic scheduling, and backoff mechanisms may influence the collision probability. More sophisticated models, such as slotted schemes, binomial distributions, and event-driven simulations, could refine this estimate

further. Nevertheless, the current model provides a practical basis for system-level design decisions. The respective handling of collisions within the routing protocol is described in the following section on error management.

### 3.3.5 Flow Chart of the Routing Logic with Relay Echoes

The Relay Echo timer procedures and SNR calculation procedures are essential components of the extended routing logic. These procedures are integrated into the following flowchart, illustrated in Figure 16. The process begins at step C1, which initiates the Rtelay-based decision logic. Unlike previous routing stages, this process can be initiated not only at the UAV level, but also at the UUV, as indicated by step C36. At step C37, the Relay Echo timer is initialized for a predefined transmission interval. The timer is then decremented at each time step. Once the timer reaches its expiration condition at C39, the Relay Echo message is transmitted at C40. The following Table 11 shows an example of a Relay echo message. Not all values need to be sent, as the decisive metric for SNR is the calculation of SNR from RSSI and noise. However, values such as battery status and coordinates can be used for further routing logic.

Table 11: Overview of the message structure Relay Echo. This is an example and can be adapted as required. It is not the content of the message that is important, but rather the SNR value determined at the USV from RSSI and noise.

| **Field** | **Size in bits** | **Size in bytes** | **Description** |
|---|---|---|---|
| Node ID | 8 | 1 | Unique node identifier (not a full IPv4 address, but a local ID from 1–255) |
| Sequence Number | 8 | 1 | Running number for identifying the message |
| Battery Level | 8 | 1 | Battery in % (0–100), scaled to fit in one byte |
| Heading horizontal | 8 | 1 | Direction (0–360°) |
| Heading vertical | 8 | 1 | Heading (−90° to +90°) |
| Position X | 8 | 1 | X coordinate, compressed to 1 byte |
| Position Y | 8 | 1 | Y coordinate, compressed to 1 byte |

| | | | |
|---|---|---|---|
| Position Z | 8 | 1 | Z coordinate (depth), compressed to 1 byte |
| **Total** | **64** | **8** | |

After the responding USV sends the Relay Echo message (C40), it is forwarded to the next USV in the network. At the second intermediate node, it is checked to see if a Relay Echo message (C30) has been received.

During this step, the message is validated using a verification mechanism (*check_m*), which ensures the uniqueness and integrity of the received data. Subsequently, a Relay Echo Table (C31) is created, where all relevant information from the received Relay Echo messages is stored. This table is used to select the most suitable Relay node in a later step (C12).

The structure and content of the Relay Echo Table are illustrated and explained in Table 11. Figure 17 shows what the USV Reply Table looks like in the simulation. The following step involves the USV at Hop 2 responding with a Relay Reply message (C32), which is initially triggered by a Relay Request message (C25). The Relay Reply message is routed back to the original sender via the nodes listed in the routing tables. Meanwhile, the sender continuously monitors for any Relay Replies that have arrived (C8). After the 30 ms timer expires, the sender creates the final USV Reply Table, which records all available USVs, their positions, and their associated SNR values. Based on this information, the sender selects the USV with the highest SNR value. If no SNR value is available, or if multiple USVs report the same SNR value, the system uses the internal routing logic and applies a geo-based routing process instead. The rest of the routing procedure follows the steps described in B14 and is referenced here as C13. Ultimately, the target UUV receives the data (C41), and the data packet is delivered successfully (C43). To ensure the relevance and timeliness of received Relay Echo information, a Relay lifetime value is assigned to each entry in the Relay Echo table. This parameter defines how long a Relay Echo is considered valid for the routing decision process. If a Relay Echo expires before the corresponding RelRep is processed, it is discarded and excluded from the USV selection. The Relay lifetime is implemented as a timestamp-based timeout mechanism. When a node receives a Relay Echo, it records the local (simulation) time. If no RelRep is generated or processed within the defined validity interval, the entry is marked as invalid.

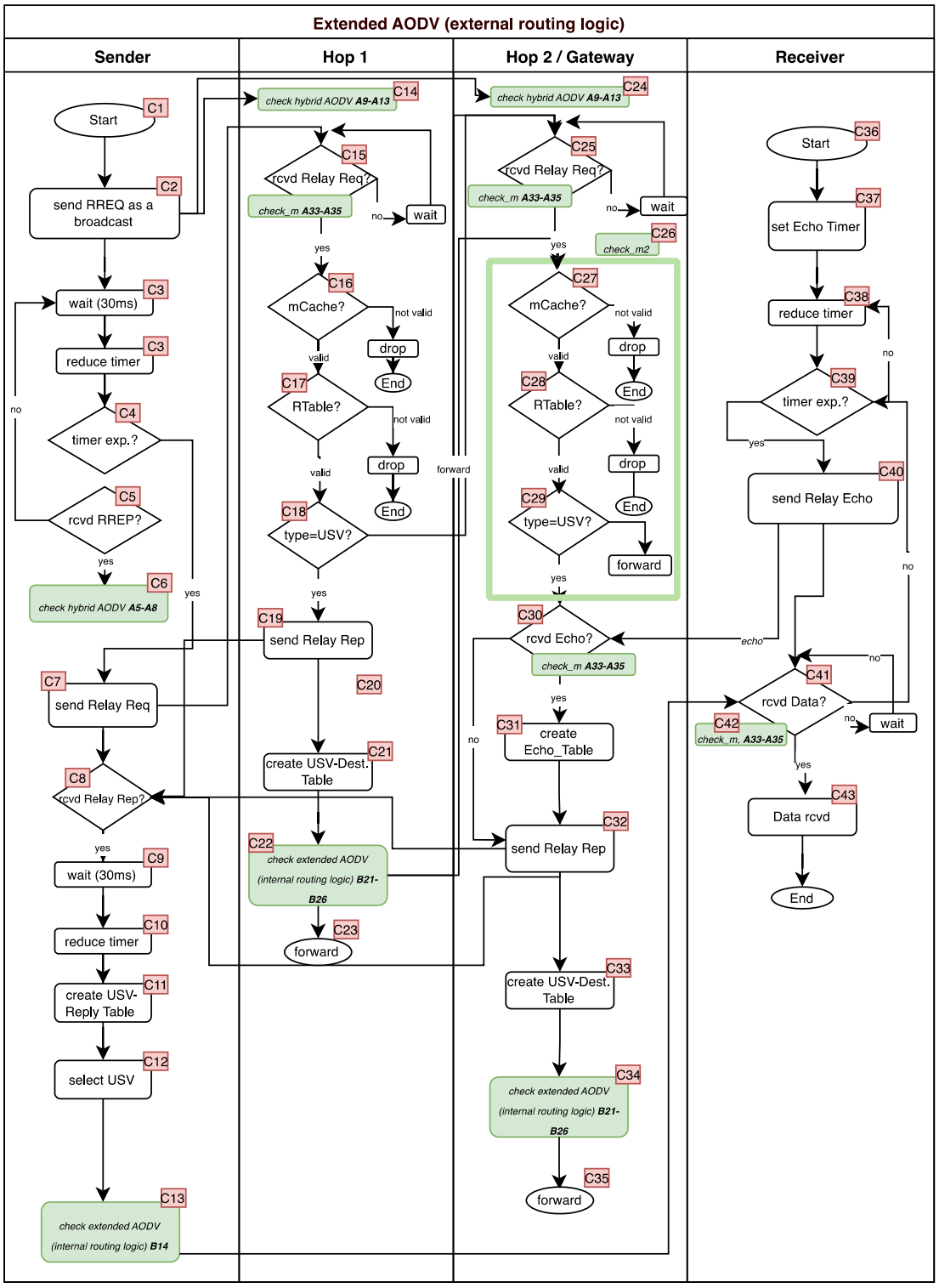


Figure 16: Flowchart of the extended AODV of the external routing logic. The focus here is on the Relay Echo message type, which is sent periodically from the UUV to the USV and serves as the basis for signal quality-based routing.

| RelReq_ID | Timestamp | USV_ID | Coordinates | Timer | Relay_Echo | SNR | Relay_Lifetime |
|---|---|---|---|---|---|---|---|
| 1 | 845 | {[ 7]} | {[ 1302 958 0]} | 2 | 0 | 0 | 0 |
| 1 | 845 | {[ 8]} | {[ 549 1305 0]} | 2 | 1 | 64.353 | 749 |
| 1 | 845 | {[12]} | {[1082 1675 0]} | 2 | 0 | 0 | 0 |
| 1 | 845 | {[17]} | {[ 1455 718 0]} | 2 | 0 | 0 | 0 |
| 1 | 845 | {[18]} | {[ 1454 279 0]} | 2 | 0 | 0 | 0 |
| 1 | 845 | {[19]} | {[ 627 839 0]} | 2 | 1 | 71.466 | 745 |
| 1 | 845 | {[ 1]} | {[ 384 1452 0]} | 2 | 1 | 65.855 | 751 |
| 1 | 859 | {[ 9]} | {[ 1913 871 0]} | 16 | 0 | 0 | 0 |
| 1 | 859 | {[11]} | {[ 165 1920 0]} | 16 | 0 | 0 | 0 |
| 1 | 859 | {[14]} | {[ 681 129 0]} | 16 | 1 | 71.82 | 745 |
| 1 | 865 | {[16]} | {[1390 1683 0]} | 22 | 0 | 0 | 0 |
| 1 | 871 | {[20]} | {[ 1755 307 0]} | 28 | 0 | 0 | 0 |
| 1 | 873 | {[10]} | {[ 140 115 0]} | 30 | 1 | 75.184 | 747 |

Figure 17: Excerpt from the command window of the MATLAB simulation. The USV Reply Table at Sender is shown. This is created when a RelReq is sent and filled with entries from the USVS at RelRep.

## 3.4 Data Routing from UUV to UUV/UAV

Since the same technical challenges of underwater communication arise when transmitting data from a UUV to another UUV or to a UAV (Figure 18), the routing mechanism has been extended to support the transition from hydroacoustic to radio-based communication. The previously introduced extended AODV, which integrates internal and external routing logic, was originally designed for establishing routes from a UAV to a UUV via USVs acting as gateways. This mechanism is now adapted to enable routing in the reverse direction, thereby ensuring consistent Cross-Domain communication between all platform types.

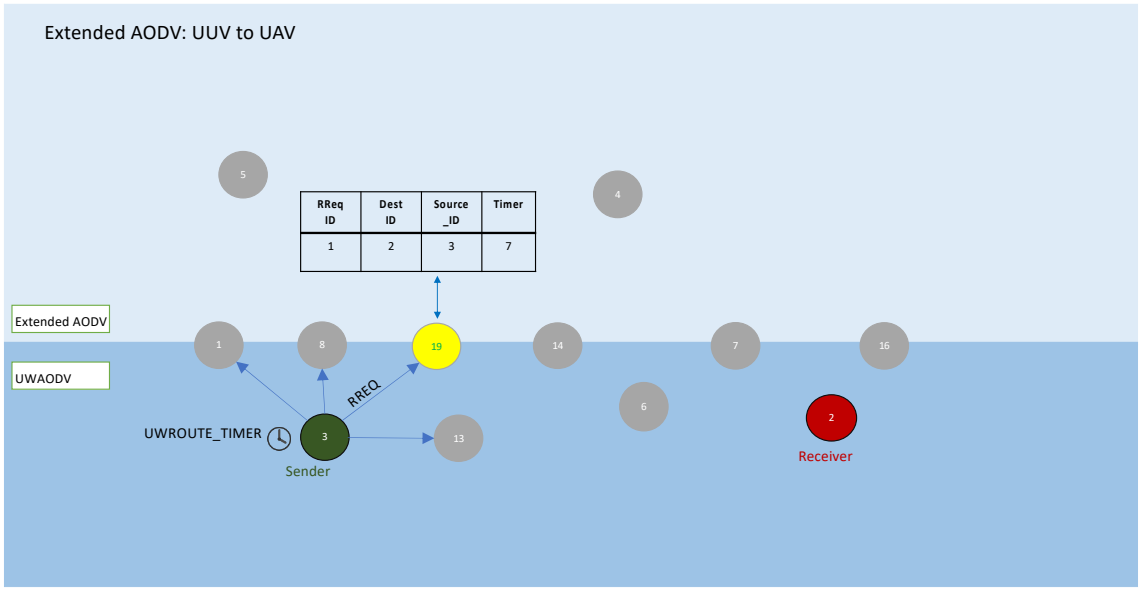


Figure 18: Illustration of the route setup from one UUV to another located several Hops away. The underwater route setup should be reduced to one Hop. In this case, the route is set up via the significantly faster and less interference-prone radio channel.

The protocol aims to take advantage of the significantly higher bandwidth and transmission speed of the radio channel by identifying suitable gateways on the water's surface (USVs).

Through a combination of geolocation- and signal quality-based routing, the extended protocol enables Cross-Domain communication between UUVs and UUVs or UAVs with minimal interference and transmission delay. The routing logic is described in detail in Figure 19. The process begins at step D1, where a RREQ is broadcasted. The RREQ is configured with TTL value of 1. This value serves a specific purpose. When the destination is a UUV or USV located only one Hop away, the protocol prevents unnecessary radio channel forwarding. After the RREQ broadcast, a timer is started. In this case, the waiting time is defined as 1500 ms. During this time, the node waits to see if it receives a RREP directly from the target UUV.

This ensures that direct hydroacoustic replies are prioritized whenever possible, as they offer lower latency in such close-range scenarios. Meanwhile, since the RREQ is broadcast, it is received not only by the intended destination, but also by all neighboring nodes within transmission range, including the USVs (D15). Each USV begins to act upon receiving the broadcast RREQ by creating a local USV Destination Table in D16, which temporarily stores the relevant destination information. To prevent multiple USVs from responding simultaneously, each USV randomly selects a backoff delay between 1 and 20 ms (D17) before initiating a response. After this delay, in D20 the USV sends a Ping message to the originator of the RREQ to indicate its availability as a potential Relay or gateway. At the sender node, the process continues with Step D9, in which it waits for incoming Ping messages from USVs.

However, if a RREP is received within the timer interval defined in Step D5, the protocol reverts to the established routing logic described in Steps A5 to A8, since a direct route to the destination has been discovered. If no RREP is received within the timeout period, the sender creates a Ping Reply Table to collect incoming Ping messages from the surrounding USVs. This table is used to select a suitable Relay node. Various selection strategies could be applied at this point. For example, it is possible to choose the USV with the best signal quality or the shortest distance to the target. In this case, however, the first-come, first-served approach is chosen.

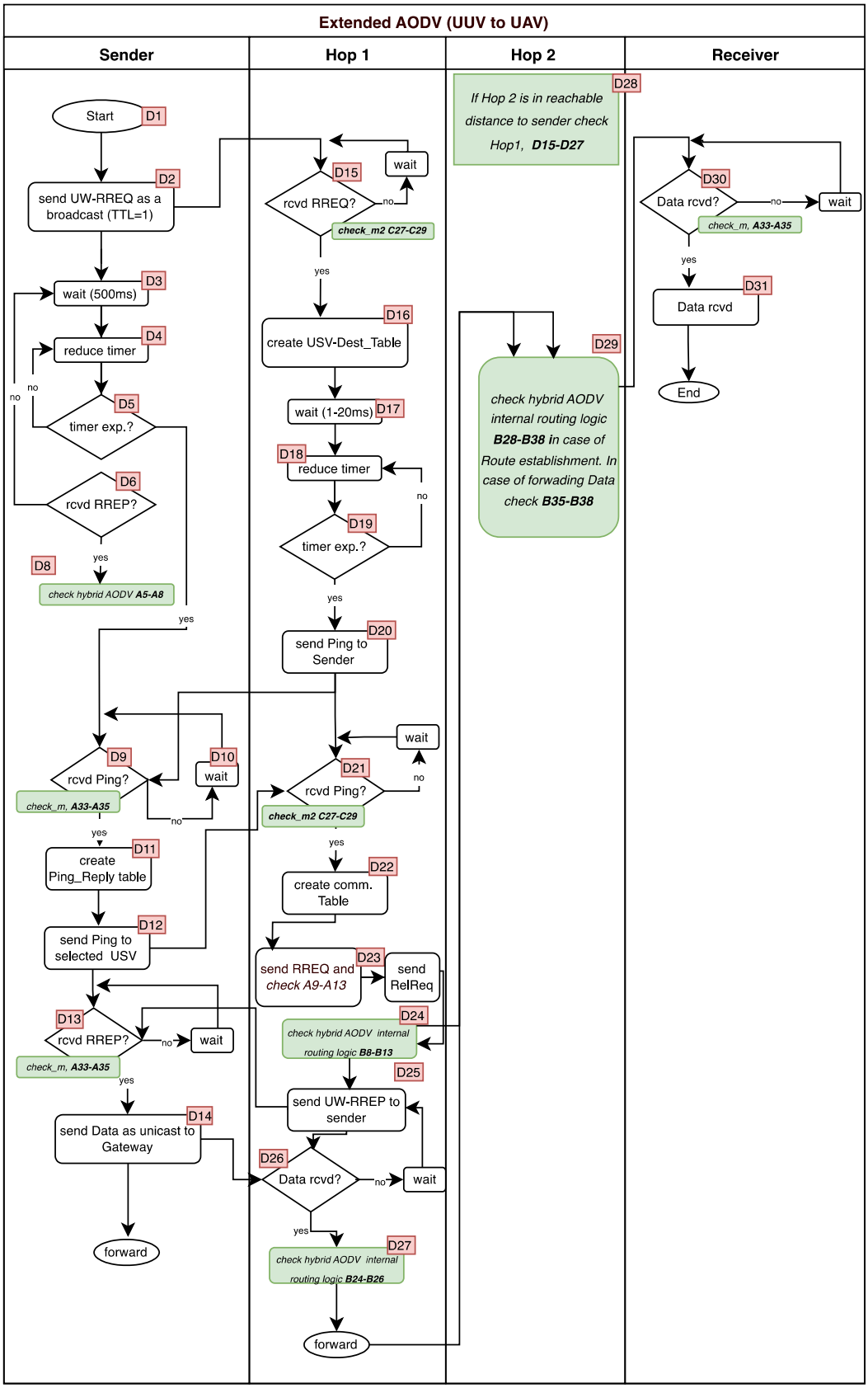


Figure 19: Flowchart of the routing logic of the UUV to the UUV or a UAV. The sending UUV can reach a target UUV with one Hop. If the target is further than one Hop away, the internal routing logic is used in the radio channel.

Specifically, the first USV to respond with a Ping message is selected as the forwarding node. Table 12 provides an example structure of the Ping Reply Table.

Table 12: Overview of the Ping Reply Table. The USVs receive the Ping, set a random timer, and respond with a Ping specifically directed at the sending UUV.

| **Req_ID** | **Timestamp (ms)** | **USV_ID** |
|---|---|---|
| 1 | 341 | 8 |
| 1 | 360 | 19 |
| 1 | 400 | 1 |

If a specific USV has been selected as the next Relay, a targeted unicast Ping is sent to it. This message initiates the routing process in the above-water domain. When the Ping message arrives at the USV (step D21), it is validated using the *check_m2* function, similar to the integrity checks described in steps C27 to C29. This ensures that only valid, timely Ping messages are processed further. In the next step, the USV creates a communication table that plays a critical role in the bidirectional routing process. This table stores the address and identity of the originating UUV. The table enables the USV to correctly forward the RREP to the UUV once a complete route has been established in the above-water (radio) domain.

Table 13: Illustration of the communication table. This is created to store the ID of the initial sender in the USV as the Source_ID.

| **Req_ID** | **Node_ID** | **Source_ID** | **Dest_ID** |
|---|---|---|---|
| 1 | 8 | 3 | 2 |

The communication table is shown as an example in Table 13. It consists of a RREQ ID for identification purposes, a node ID (the unique ID), a source ID (which indicates where the message originated), and a destination ID (which indicates the destination). Once the communication table has been created at the USV, the routing process in the above-water domain begins. This follows the same internal logic previously described. In step D23, the USV broadcasts a RREQ. If the destination is a UAV or another USV, an RREP is returned, as outlined in steps A9–A13, and the route is established. Otherwise, the USV

sends a RelReq, triggering the routing sequence outlined in steps B8–B13. If a second node (Hop 2) receives the request, it acts as a standard Relay node and executes the routing logic established from B26 to B38. If Hop 2 can be reached directly by the original sender (the initiating UUV), the routing logic defined for Hop 1 applies. This logic is illustrated in the flowchart under steps D15 to D27. At this point, Hop 1 queries all available USVs with gateway functionality and requests their coordinates via RelRep messages. Similar to the external routing logic, an optional, signal-quality-based routing mechanism can be integrated at this stage by evaluating the content of incoming Relay Echo messages. The USV that created the communication table initially evaluates the collected responses to determine the most suitable USV to act as a gateway. Based on this selection, the USV generates a RREP in step D25, which is sent back to the original sender. The destination address embedded in the RREP corresponds to the final USV selected as the gateway. At the selected USV, a USV Destination Table is created, analogous to step B27, containing the target UUV address. To cross from the radio domain back into the hydroacoustic domain, the RREP is marked with a flag derived from the source ID stored in the communication table. The destination UUV receives the data packet in step D30. As in earlier stages, the message undergoes a validation check using the *check_m* function to ensure the integrity and accuracy of the received data.

## 3.5 Error Management

In the original AODV protocol, as specified in RFC 3561 (Perkins et al., 2003), *Hello* messages are used to establish communication between nodes. These short messages inform all receiving nodes that the sending node is reachable. Each node periodically broadcasts a *Hello* message. Upon reception, these messages are used to refresh the validity of routes that include the sending node. Each routing table entry maintains a timestamp known as the TTL, which is continuously decremented for this purpose. When a *Hello* message is received, the corresponding TTL resets. If the TTL reaches zero, indicating that the last *Hello* message is not received, the route is considered invalid. An error is detected and the protocol transitions to error handling, which has limitations. Firstly, network overhead increases because all nodes must periodically transmit *Hello* messages, even when no data is being sent by users.

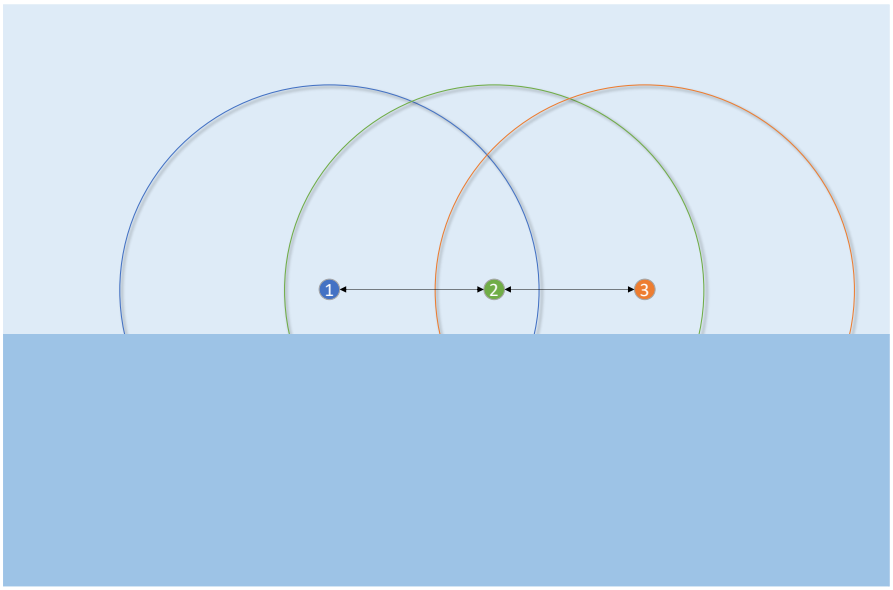


Figure 20: Isotropic representation of radio communication, which enables passive acknowledgment of error management.

Secondly, data packets transmitted between a link failure and the expiration of the corresponding TTL are not identified as lost. Such losses can be mitigated by reducing the interval between *Hello* messages. However, this will increase network traffic, which is undesirable, especially under constraint NFR2. To overcome these limitations, a new error detection approach based on passive acknowledgments, as described in (Y.-H. Wang & Chao, 2006) has been developed.

This method assumes that all nodes emit isotropic signals, based on the physical properties of antenna technology (Kark, 2006) and illustrated in Figure 20. Furthermore, all nodes transmit at the same power level, resulting in equal transmission ranges. Consequently, a forwarded message is received by not only the next Hop in the route, but also the previous Hop.

In the original AODV protocol, retransmitted messages are usually discarded because their contents have already been processed. However, when passive acknowledgments are used, these messages are reevaluated to ensure they are forwarded correctly. When data is transmitted along a route, each node stores the packet before forwarding it. When a node receives a subsequent packet, it checks if the packet matches the stored information and if the sender corresponds to the next Hop.

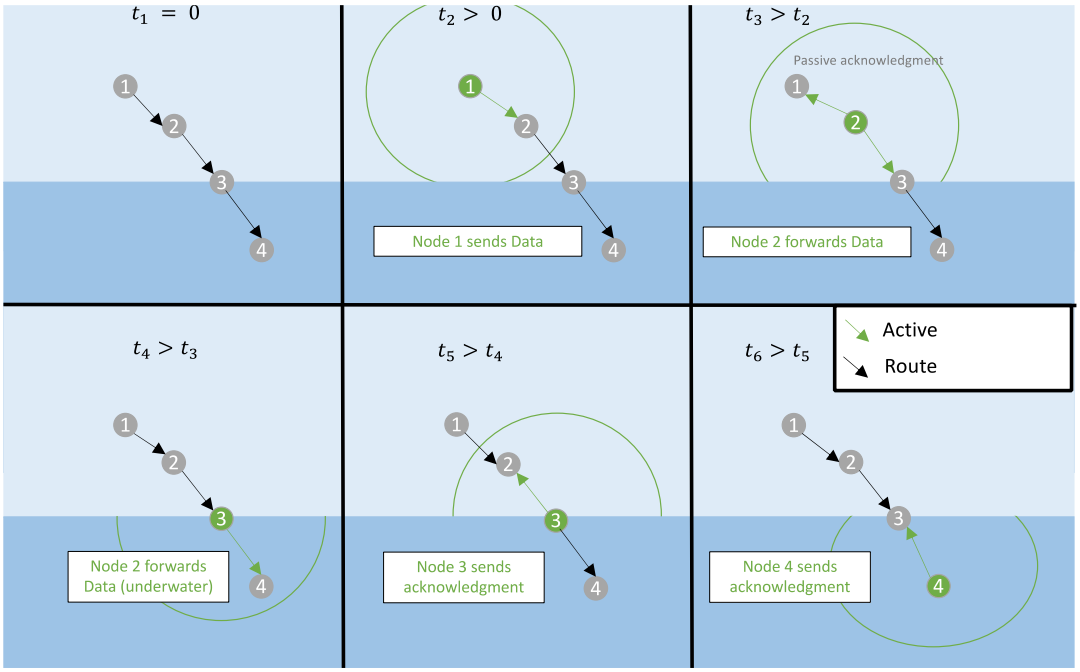


Figure 21: Sequential flow of passive acknowledgment of error management. The green arrow indicates active data transmission, while the black arrow indicates the route.

If both conditions are met, the packet is considered forwarded successfully and is removed from memory. Thus, transmission to the next Hop is implicitly acknowledged. If the next Hop is the final destination, the data packet is not forwarded. In this case, to ensure successful delivery, a new message type labeled "Acknowledgement" (ACK) is introduced. The destination node sends this message to the previous Hop to confirm receipt of a data packet, as illustrated in Figure 21. Similar to the reception of a *Hello* message, the ACK is bound to a specific time interval. However, this interval refers to an individual packet rather than the availability of the entire node. Consequently, the absence of an acknowledgment directly signifies the loss of a particular packet rather than being associated with a time window during which multiple packets could be transmitted. This error-detection approach has a structural limitation in that failures are detected only when the route is actively carrying data, since acknowledgements are event-driven. In the Error Management approach developed in this work, over-water links use passive acknowledgements, meaning that forwarding is inferred rather than explicitly confirmed. If forwarding is not detected within the timeout period, the upstream node issues a route error; however, this message may be lost if another node fails on the reverse path to the source. As an extension, outside the scope of this work, per-unicast reliability can be added, including for the error message itself. In contrast, periodic, *Hello*-based detection can identify broken links without the need for active data traffic, albeit at the cost of additional

overhead. However, due to the hybrid nature of the communication network, routes that span different domains cannot be verified using classical passive acknowledgment. When a packet reaches a gateway (USV) and is forwarded across domains, this forwarding step cannot be observed from within the original domain. This is due to the high attenuation of radio waves in water and hydroacoustic signals in air. For this reason, a different approach to acknowledgment is required at domain interfaces. When the gateway forwards a packet to another domain, it must issue an acknowledgment within the source domain. To enable this, the ACK message, which is previously only associated with the final destination, is repurposed. This results in a hybrid passive acknowledgment mechanism that extends the classical approach by incorporating additional ACK messages at domain transitions.

The process begins in Figure 22 at step E1, where a route to the destination node is established. Depending on the scenario, this may involve the internal routing logic of hybrid AODV, the external decision-making logic of extended AODV or a hybrid routing path from a UUV to a UAV via the extended protocol. Once the route has been successfully established, the process moves to step E2, where transmission of the data towards the final destination begins. As soon as the sender has transmitted the data packet, an acknowledgment timer is initialized (E3). The packet is then forwarded to the next Hop. When it arrives at the receiving node, the process checks that the packet has been received successfully (E11). The standard validation steps are executed subsequently: The node verifies whether the message is already stored in the local cache (E12) and whether it matches an active entry in the routing table (E13). At the receiving node, the first step is to determine whether the packet belongs to the over-water (radio) or underwater (hydroacoustic) domain. If the packet belongs to the over-water domain, two scenarios are possible. First, the node checks whether the packet has reached its final destination (E17). If the packet has reached its final destination, a hybrid acknowledgement is sent to confirm successful delivery (E19). If the destination has not yet been reached, the packet is forwarded to the next Hop. Due to the isotropic nature of signal propagation in the radio domain, the previous node will also receive this forwarded packet. This reception serves as a passive acknowledgement that the message has been forwarded successfully. Upon arrival of the data packet in the underwater (hydroacoustic) domain, the receiving UUV sends a hybrid acknowledgement to confirm successful reception.

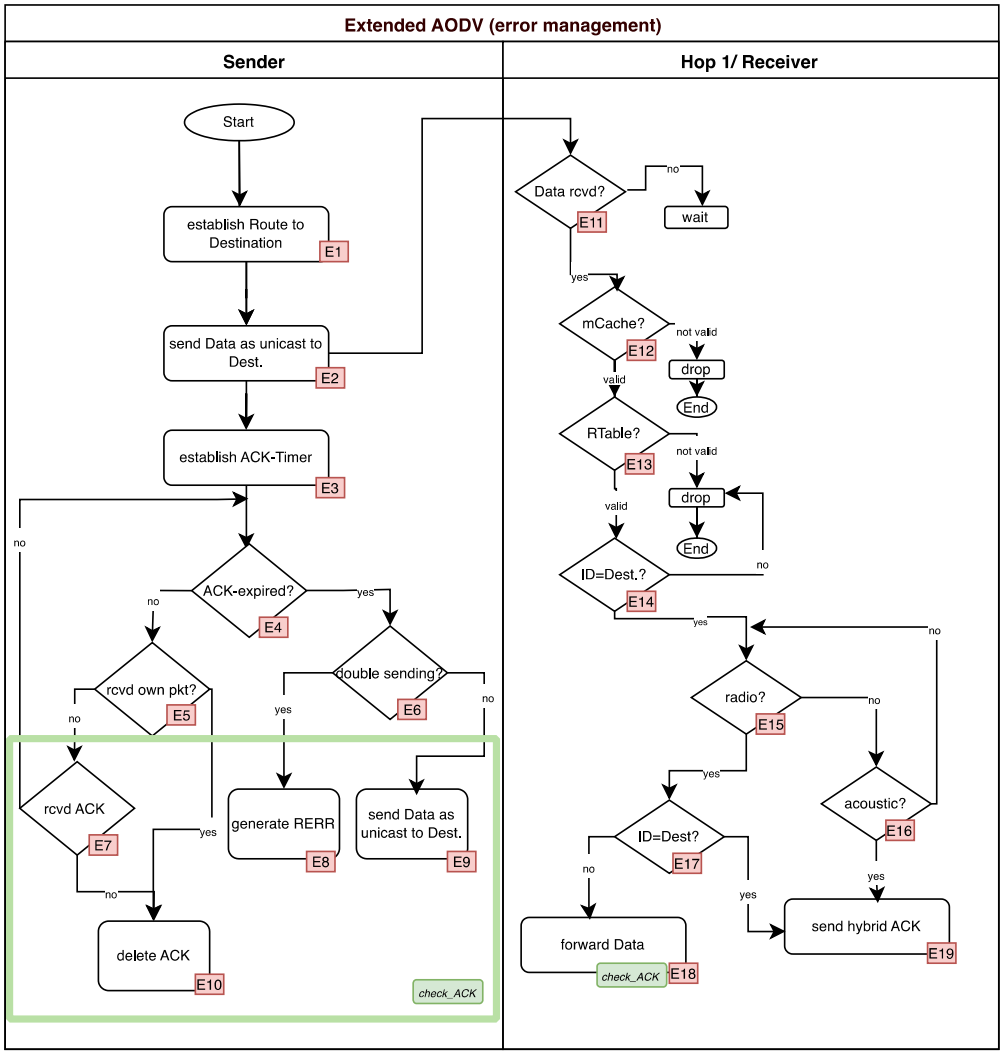


Figure 22: Flowchart of error management with passive hybrid acknowledgment.

The routing approach of this work also allows for multi-Hop underwater communication, based on the same principle used in the over-water domain. However, the question of whether isotropic signal propagation behaves similarly in the hydroacoustic channel remains unanswered and is beyond the scope of this work. In step E4, the original sender simultaneously starts a defined acknowledgement timer set to 30 ms and waits to see if its own data packet is received within this time window. Assuming it is not the final destination, the next Hop (Hop1) forwards the packet (E18) based on its isotropic radiation pattern. The original sender is therefore expected to receive a copy of its own data packet. Upon reception, the sender considers the transmission to be successful and removes the corresponding ACK timer entry from its internal table (E10). However, if the ACK timer expires (E4) before the sender has received its own packet, it retransmits the data packet once. This retransmission is included because random packet losses may occur on the

channel, particularly underwater. If the timer expires again without successful reception, a RERR message is generated. This RERR message is sent back to the sender via unicast (step E8), prompting them to restart the route discovery process according to the chosen extended AODV logic (internal or external). The RERR message includes information on the destination to which the route has failed.

The same error-management mechanisms that are used for unicast data messages are also applied to unicast control messages, such as RERR, RREP and RelRep, when these messages are lost. This ensures consistent handling of packet loss across the data and control planes.

## 3.6 Comparative Overview of Developed Routing Concepts

The following section provides a summary of all the developed routing concepts and a comparison of their potential advantages and disadvantages in relation to the reference protocol hybrid AODV. This comparison is presented in Table 14.

Table 14: A summary of the developed routing concepts.

| **Routing Concept** | **Description** | **Advantages vs. hybrid AODV** | **Possible Drawbacks** |
|---|---|---|---|
| **Hybrid AODV (Reference Model)** | The hybrid AODV protocol has been implemented with the particularity that messages propagate in both the radio and hydroacoustic domains, resulting in significant propagation delays. | Serves as a reference baseline; enables the use of hybrid AODV in a Multi-Domain setting, thereby providing fundamental communication between UAVs and UUVs. | There is high control overhead in the hydroacoustic domain and it is not optimized for domain-specific conditions. |
| **Extended AODV (Internal Logic)** | Extension of hybrid AODV with the message types RelReq and RelRep, enabling the selection of the gateway with the shortest geographical distance to the UUV. | No need to transmit route establishment messages hydroacoustically; significantly reduces signaling load in the underwater domain. | The mission plan or UUV position must be kept in line with the predefined coordinates to avoid errors; this requires accurate |

| | | | |
|---|---|---|---|
| | | | knowledge of node positions. |
| **Extended AODV (External logic)** | The internal logic is extended by the Relay Echo message, which UUVs periodically transmit to enable signal-quality-based routing. | Provides precise information about the availability of UUVs and, more importantly, about the current link quality. | Generates additional traffic in the already interference-prone hydroacoustic domain, potentially reducing efficiency under high noise conditions. |
| **Extended AODV (Subsea Routing)** | Subsea routing describes routing in the hydroacoustic domain over multiple Hops or to the surface domain. | Hop limitation underwater, reduction of redundant messages, use of faster radio channel. | The duration of route establishment depends on several timers, which lead to similar propagation times as hybrid AODV even when data traffic is reduced. |
| **Extended AODV (Error Management)** | The network is maintained through passive acknowledgement mechanisms whenever errors occur during the transmission of unicast messages, such as Data, Error, RREP or RelRep, for example due to a node failure along the route. | It increases the reliability of unicast message delivery and provides unified error handling across data and control messages. It also improves robustness in dynamic networks without generating additional data traffic through ACKs. | Additional processing overhead; passive acknowledgments may not be sufficient in high-loss environments, potentially leading to delayed route recovery. |

# 4 Simulation and Analysis

This chapter introduces the MATLAB-based simulation environment that is used to evaluate and compare the extended and hybrid AODV. First, the simulation's setup and configuration, including its modular structure, key components, and supporting configuration files, is described. The underlying assumptions are subsequently outlined. The agent-based role assignment used in the simulation is also introduced. In the simulation, each node represents a specific platform type. Examples of these platform types include UAV, USV, or UUV. The following section explains the integration of routing protocols into the modular simulation framework, as well as their functional interaction with communication layers. This ensures that the routing behavior can be evaluated reproducibly and comparably within the simulation environment. Afterwards, the simulation is parameterized based on a practice-oriented demonstrator that represents a scenario for Cross-Domain communication. The demonstrator serves as a reference model and basis for adjusting simulation parameters in a realistic operational context. After outlining the foundations of the simulation, the final section of this chapter presents the results. These results align with the Case study described in Section 1.2. A selected set of nodes is used to test and analyze all the routing protocols developed in Chapter 3 under comparable conditions with respect to key performance metrics.

## 4.1 Setup and Configuration of a Simulation Framework

As illustrated in Figure 23, the Multi-Domain Ad-Hoc simulation framework comprises various interrelated classes and functions. The entire simulation framework is implemented in MATLAB R2021a and is based on a foundational structure initially available on GitHub (Voitenko, 2017). This framework provides a conceptual basis for simulating Ad-Hoc Networks and centers on the OLSR routing protocol. However, it is not a complete or State of the Art simulation environment for this purpose and is therefore not included in the related work section. To meet the requirements of this study, the original framework is expanded from a two-dimensional setup to a fully three-dimensional simulation environment. According to the defined use case, physical models were added to support the above-water and underwater domains. These models reflect the unique physical properties of each communication medium. Since this work primarily focuses on the

network layer, the simulation emphasizes this level in particular. In this context, several routing strategies, including standard AODV and an extended version of AODV tailored for domain transitions (UAV–UUV and UUV–UUV/UAV) are developed and tested, as well as dedicated error management mechanisms. The simulation architecture is structured around three core classes, each of which is responsible for specific functionalities and internal logic.

In addition to these classes, a centralized configuration file (*config.ini*) defines all relevant simulation parameters. This file enables the flexible definition of key simulation properties, such as the size of the simulation area, the number of participating nodes, the total simulation time, and the physical characteristics of the wireless and hydroacoustic transmission systems. Users can adapt the simulation to a variety of operational scenarios by modifying the *config.ini* file. For example, it is possible to enter the specific physical properties of antennas for aerial communication or hydroacoustic modems for underwater links. This modular design provides high flexibility and ensures that realistic and custom hardware profiles can be seamlessly integrated into the simulation. The *manet.m* file serves as the central entry point for executing the simulation. It initializes all core components, including network nodes, physical models, routing protocols, and agent roles. Using the parameters defined in the external configuration file, the script sets up the simulation environment and assigns UAV, USV, and UUV roles to the nodes. It also generates their initial positions in three dimensions. Throughout the simulation loop, the script performs several tasks: It manages discrete time steps, updates the network topology, processes communication events, and visualizes transmission activities.

The *manet.m* is structured as a discrete-time loop that iterates through all nodes at each simulation step. For each time step, the loop sequentially processes every node and executes the associated class methods and functions. These operations include packet generation, routing logic, and link evaluation. This discrete-time approach enables time-synchronized updates of all network components and ensures consistent interaction across domains with different propagation characteristics. Depending on the node type and underlying communication medium, radio or hydroacoustic channels are selected. Additionally, data generation, packet forwarding, and visualization are handled dynamically during runtime. The modular structure allows for the flexible adaptation of routing logic and supports protocol-specific behaviors, such as message dissemination, Relay selection, and error handling.

To support reproducibility and facilitate further analysis, the simulation can be configured to record the positions of the nodes and output performance statistics at the end of each run. The *missionplan* file contains the UUVs' movement schedule. This plan reflects the predefined paths known to the UUVs according to their routing logic. During the simulation, the positions of the UUVs are logged throughout the entire runtime, which enables the evaluation and visualization of node trajectories. While the mission plan is typically generated automatically based on the simulation setup, it can also be defined manually to simulate specific operational scenarios or mission profiles. The simulation framework is organized into several core classes, which are located in the classes directory. Each class is responsible for a distinct functional aspect of the simulation. For example, the class *Waypoint* governs node mobility and enables either deterministic or randomized movement patterns depending on the scenario configuration.

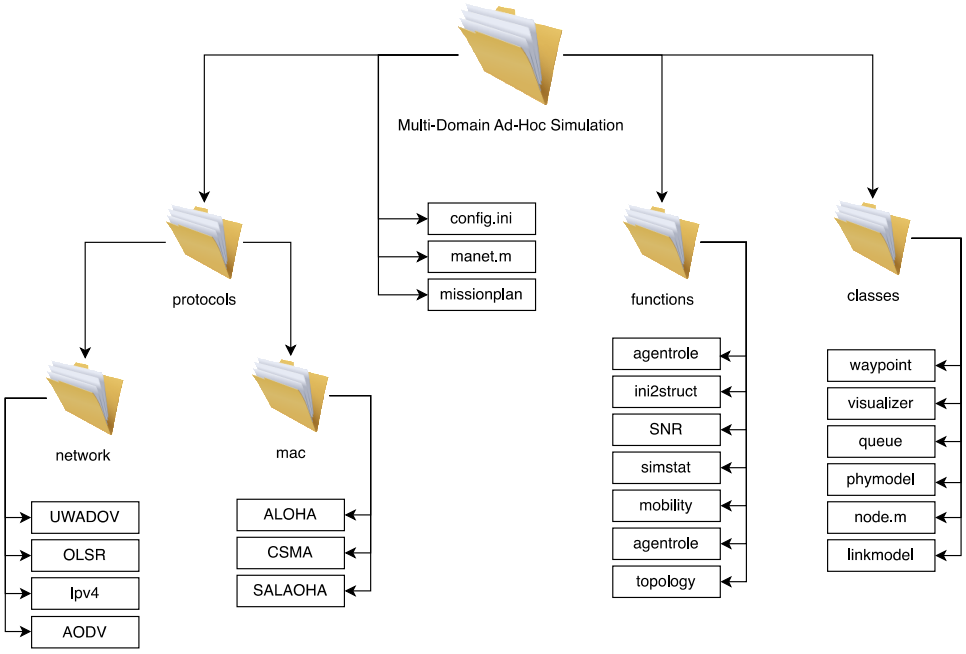


Figure 23: Folder structure of the MATLAB-based simulation framework. The fundamental structure was initially adopted from an open-source implementation available on GitHub (Voitenko, 2017), while the network layer was extensively redesigned in this work to support two communication domains and dynamic domain transitions.

The class *Visualizer* provides the simulation's graphical interface. It visualizes node positions and communication events in real time, providing a three-dimensional representation of network behavior. Messages transmitted between nodes via radio or hydroacoustic

links are dynamically displayed to support an intuitive understanding of domain-specific routing processes. The simulation's core element is its queuing system. Each node maintains separate queues for radio and hydroacoustic communication. UAVs only have a radio queue, UUVs only have a hydroacoustic queue, and USVs have both. This domain-specific queue architecture ensures strict separation between aerial and underwater communication, preventing unintended Cross-Domain message propagation. The class *PhyModel* is essential for modeling the physical layer. It calculates transmission time and propagation delay based on the physical characteristics of the communication medium. These characteristics include signal attenuation, range, and environmental factors, such as water depth and temperature, in hydroacoustic channels. The current physical layer model does not include Doppler effects or multipath propagation, as these would require complex, site-specific acoustic modeling and significant computational resources to be represented realistically.

As the main aim of this study is to evaluate routing performance in a Multi-Domain Ad-Hoc Network rather than to carry out detailed channel modeling, a simplified transmission model is employed. The class *Node* defines the behavior and internal state of each node. It handles data generation, packet processing, queuing, and establishing communication links via *connectListener* functions. Each node can react dynamically to network events, initiate or forward messages, and update its routing status accordingly.

Lastly, the class *LinkModel* ensures proper channel behavior. It accomplishes this by marking links as busy during ongoing transmissions and releasing them afterward. This prevents channel conflicts and accurately models the temporal dynamics of wireless and hydroacoustic transmissions. Several central functions support the execution and evaluation of the simulation. The *agent role* function assigns each node a specific role (UAV, USV, or UUV) based on the configuration. This classification determines the node's mobility model, communication capabilities, and participation in routing processes. The SNR function calculates the *signal-to-noise ratio*, which is important for the external routing logic. The function considers signal power, environmental noise, and transmission loss in order to evaluate the quality of the link between nodes. The *simstat* function generates key performance statistics at the end of each simulation run. These include metrics such as total data traffic, the number of dropped packets, and the number of successful transmissions. This statistical output is essential for analyzing the effectiveness and reliability of the implemented routing protocols. The *topology* function determines the

network topology at each discrete time step. It returns an adjacency matrix indicating the current connectivity between nodes. A value of 1 represents a radio connection and a value of 2 represents an hydroacoustic link. This dynamic topology matrix manages neighbor relationships and channel access during the simulation. The MAC directory contains simplified versions of ALOHA-based MAC protocols. These protocols are integrated into the simulation to enable basic channel access mechanisms and model medium contention in a realistic yet computationally efficient manner. The MAC logic primarily detects and resolves signal collisions to ensure that simultaneous transmissions do not result in data loss or interference. Although the implementation is abstract, it provides enough functionality to evaluate routing behavior under realistic network load and access conditions. The Network directory contains the core components of the network layer within the simulation framework. The *IPv4* class forms its foundation and handles basic IP-related functions, such as message formatting, addressing, and validation. Routing protocols are implemented on top of this layer in a modular and domain-specific manner. There are currently two integrated routing protocols in use. One is a customized version of AODV. This is used for above-water communication. The other is an extended UW-AODV, which is used for underwater communication. These implementations are strictly separated to reflect the physical and functional differences between the two domains. The terms "AODV" and "UW-AODV" do not represent the standard form of the original AODV protocol. Rather, both protocols have been modified and adapted to the specific routing logic developed in this work (hybrid AODV, extended AODV with the internal and external logic, as well as the error management). The modular architecture allows for the integration of additional underwater routing protocols, such as OLSR. These protocols remain in their original, unmodified form, however, and are not part of the evaluation presented in this thesis.

## 4.2 Assumptions for the Simulation - Simulation with Agents

As shown in Figure 24, a specified number of nodes are distributed across a defined simulation area in the three-dimensional Cartesian simulation environment. Each node is assigned an agent role using the *agentrole* function. The nodes are then initialized accordingly in the main simulation script and receive the appropriate communication protocols and transmission queues based on this assignment. The configuration of node roles is

based on a typical setup for Multi-Domain Ad-Hoc Networks, which usually consists of aerial, surface and underwater agents. A common configuration includes:

- UAVs for aerial reconnaissance.
- UUVs to generate a subsurface situational awareness picture.
- USVs operating on the water's surface act as gateways.

USVs serve multiple purposes. They provide a visible physical presence and can be used as sensor platforms equipped with radar, AIS and sonar. They can also function as landing and charging stations for UAVs. Each agent is assigned a role and an initial coordinate within a domain-specific altitude or depth range. UAVs typically operate at altitudes between 0 and 100 meters. USVs are located at the water's surface (z = 0) and UUVs are positioned at depths between 0 and –30 meters.

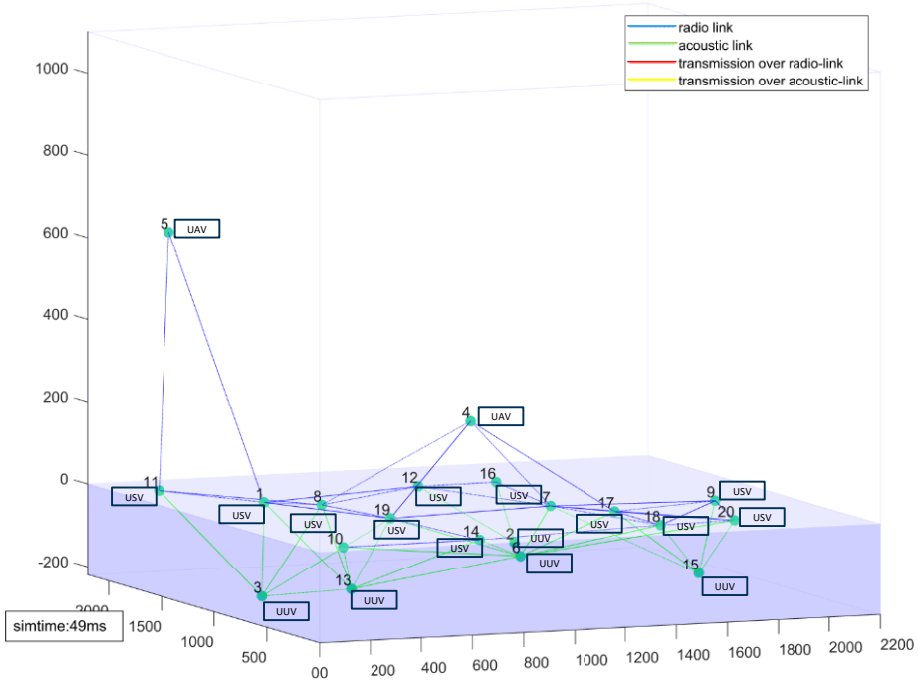


Figure 24: Excerpt from the visualization of the MATLAB simulation. It illustrates which agent roles are assigned to the nodes. These are UAVs, USVs as gateways, and UUVs.

This configuration realistically demonstrates the separation of domains and the dynamic conditions that affect routing processes. Although node mobility can be configured variably, it plays a minor role in route establishment as routes usually form within one to two seconds, during which time node positions do not change significantly. Mobility becomes more relevant during error management evaluations, particularly when link disruptions are introduced intentionally. In addition, the physical models used in the simulation are subject to several assumptions. These models are parameterized for specific environmental conditions as defined in this work, and cannot be generalized. Variations in temperature, modem frequency, wind or wave conditions may significantly alter the noise level and consequently the channel characteristics.

## 4.3 Necessary Integration Steps to Implement the Routing Protocol into the Framework

To enable evaluation of different routing protocols within the MATLAB simulation framework, specific integration steps were undertaken. Although a thorough explanation of all implementation details is beyond the scope of this work, an overview of the fundamental interaction behavior between two nodes is provided to establish a foundation for the subsequent chapters on parameterization and simulation results. This interaction represents the minimal functional structure required for route establishment and packet forwarding, regardless of the routing logic employed.

Figure 25 depicts the underlying flow of communication between two nodes and illustrates the essential sequence of operations and message exchanges that form the basis of all protocol variants implemented in this simulation. Depending on the specific simulation scenario and execution step, the nodes may be connected via a radio link (above-water) or an hydroacoustic link (underwater). The topology function dynamically updates the communication link assignment between several nodes in each simulation step by recalculating the current connectivity based on node positions and environmental conditions. This process is implemented using an *adjacency matrix* that reflects the availability and type of links for each node pair. Before establishing an external communication interface, each node registers internal listeners using the *add Listener* method. This step establishes connections between specific functions within the same node object, enabling internal events, such as packet generation, reception, or status updates, to trigger corresponding responses. These internal bindings enable modular, event-driven behavior within each

node during the simulation process. The process begins in step F1, where the *connect Listeners* are registered in the main simulation file, *manet.m*. The same registration is performed for the second node in step F18. These listeners establish a communication interface between the sender and receiver nodes within the simulation environment. Functionally, the *listener* mechanism allows the nodes to detect and respond to transmissions from other nodes. It determines whether a node is within communication range and can receive a transmitted signal. This behavior is governed by the current connectivity matrix and the physical communication model. When a node sends a packet, only nodes with an active link in the matrix can receive and process it. This setup provides the necessary abstraction to accurately model packet propagation and reception and ensures that the routing protocol logic operates under realistic link constraints. The next step, labeled as F3, involves invoking the *generate_pkt* function on the sender side. This function belongs to the *node.m* class and generates a new packet. It is called during each iteration cycle of the simulation whenever a specific node is processed. This ensures that each node can generate and transmit packets according to its predefined behavior or role in the simulation. After generating the packet in *generate_pkt*, the simulation proceeds by interacting with the *LinkModel* to determine if the node is engaged in an active transmission. This is achieved by invoking the *link.timeout* method (step F4), which checks if the channel associated with the transmitting node is occupied. If an ongoing transmission has a remaining time greater than 0 ms (step F5), the corresponding internal timer is reduced until the transmission is complete (step F7). Only the antenna transmission time, consisting of the packet size and bit rate, is selected as the transmission duration. Radio transmission occurs at the speed of sound, so this delay is irrelevant for the simulation.

The propagation delay underwater is calculated on the transmitter side (*rcv_pkt).* This mechanism ensures that concurrent transmissions are managed consistently and that collisions or resource conflicts are avoided. Once the channel becomes available, the *sent_packet* function is executed (step F8). This central function coordinates the actual packet transmission by signaling that the link channel has been cleared for the current sender. The function acts as a synchronization point between the physical channel state and the node's logical packet-handling logic. This mechanism ensures that packets are only transmitted when the channel is idle, reflecting realistic medium access behavior within the simulation environment. If no transmission is active on the node during Step F5, the simulation invokes the *aodv.timeout* method within the *node.m* class.

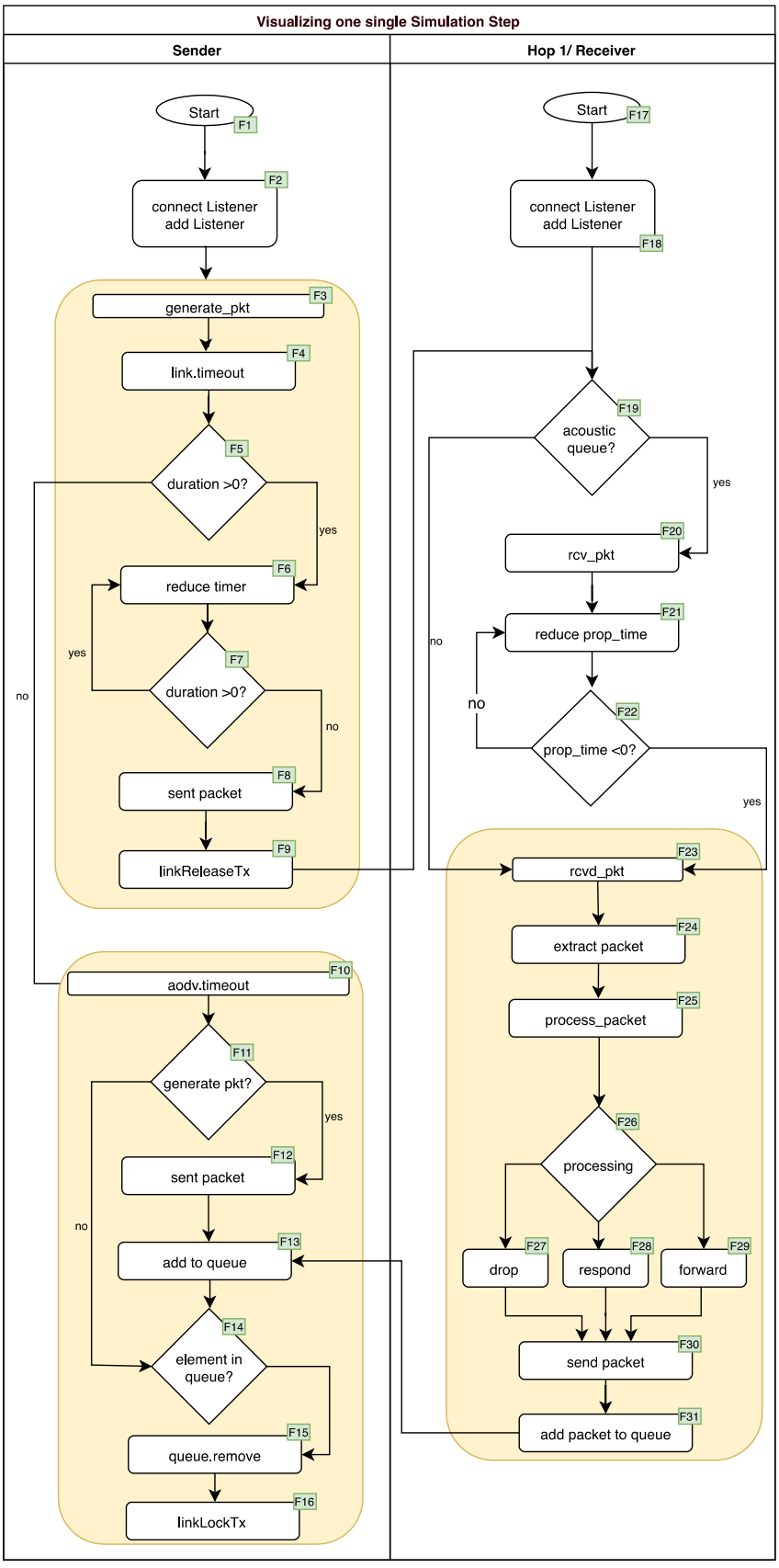


Figure 25: Flowchart of the MATLAB simulation consisting of 2 nodes. The aim is to show how communication between the nodes takes place. The listener function is used to activate events between both nodes.

This triggers the routing protocol layer (AODV or UW-AODV, depending on the node's domain) to determine if a new packet should be generated. If applicable, the *generate_pkt* method is called to create a protocol-specific packet, such as *RREQ, RREP, RelReq, RelRep, RERR, or DATA* (step F11). In the current simulation setup, the sending of messages is explicitly defined by the user. However, the framework also allows for stochastic message generation if required for specific use cases.

After generating a packet, the node triggers the *LinkLockTx* function (step F16) to indicate that it is currently busy transmitting. This step prevents concurrent transmissions from being initiated on the same node and ensures that channel usage reflects realistic physical-layer behavior. Once the node is locked for transmission, the packet is sent through the previously selected communication channel. This busy state affects subsequent iterations. In the next simulation step, when the same node reaches the *generate_pkt* check again (Step F3), it verifies that the remaining transmission time is greater than zero. In this case, no new transmission is initiated and the internal timer continues to count down until the current transmission is complete. This creates a closed-loop process in which each node is evaluated in every simulation step, ensuring consistent temporal coordination and enforcing the exclusivity of transmission events. Once the transmission time has elapsed, the *linkReleaseTx* function is called on the sender side (step F9). This indicates that the physical antenna transmission is complete and the packet has been handed over to the receiver.

A registered listener triggers the actual transfer, simulating the arrival of the packet at the receiving node. The receiver then determines whether the packet came from the radio or hydroacoustic queue (step F19). For a hydroacoustic transmission, the *rcv_pkt* function is invoked (step F20). At this point, the packet is not delivered immediately to the upper layers. Instead, the underwater propagation delay is calculated based on distance and environmental conditions. The corresponding propagation time is then added to the packet. Once the propagation time has expired (step F22), the *rcvd_pkt* function within the node class is called. This finalizes the reception process and delivers the packet to the appropriate processing routines within the node. Separating transmission duration and propagation delay is critical for accurately modeling underwater communication because transmission and propagation times differ significantly and must be handled independently to reflect physical reality. After the propagation delay expires and the packet reaches the receiving node, it is passed to the network layer and specifically to the AODV or UW-

AODV protocol module. There, the packet is first extracted (step F24) and then processed (step F25). During processing, the protocol determines if the packet should be dropped due to redundancy or if it should trigger a response, such as a RREP or RelRep. It also determines if the packet needs to be forwarded to the next Hop. If forwarding is required, the packet is returned to the node level via the *send_packet* function and inserted into the appropriate transmission queue (either radio or hydroacoustic) depending on the next Hop's domain.

Change of Domains: From Radio to Hydroacoustic and from Hydroacoustic to Radio Domain:

In the event of forwarding being necessary, the packet is sent back to the node level using the *send_packet* function and placed into the relevant transmission queue (radio or hydroacoustic), depending on the next Hop's domain. In some cases, though, the domain changes between two Hops, such as from radio to hydroacoustic communication or from hydroacoustic to radio. In Figure 26, the process begins when the packet arrives at the node and is received via the *rcvd_pkt* function (step 2). Afterwards, it is passed to the AODV protocol layer, where it is extracted and processed (steps 3 and 4). During processing, the protocol determines whether the packet is valid and intended for a destination in another domain, such as a USV. In such cases, the destination ID is assigned to a specific internal flag (*flag = dest),* indicating that the packet must be handed off to another domain for further routing. The destination ID is critical because it must be carried across domain boundaries to ensure correct addressing. After this decision is made, the packet returns to the node level (step 5). Typically, it would be inserted into the appropriate transmission queue based on the current domain. However, if the domain-switch flag is set, the packet is not queued immediately. Instead, it is redirected to the routing module responsible for the target domain. In the case of above-to-underwater communication, it refers to UW-AODV (Step 6). In the UW-AODV, the packet is processed again (step 7) and returned to the node level via the *rcvd_pkt* function. Finally, the packet is inserted into the correct transmission queue (step 9) and associated with the appropriate communication medium. Unlike its initial path, the packet is scheduled to be transmitted via the hydroacoustic channel instead of the radio channel. The domain-switching mechanism is bidirectional, supporting both radio-to-hydroacoustic and hydroacoustic-to-radio transitions. This ensures seamless communication across hybrid, Multi-Domain Network topologies.

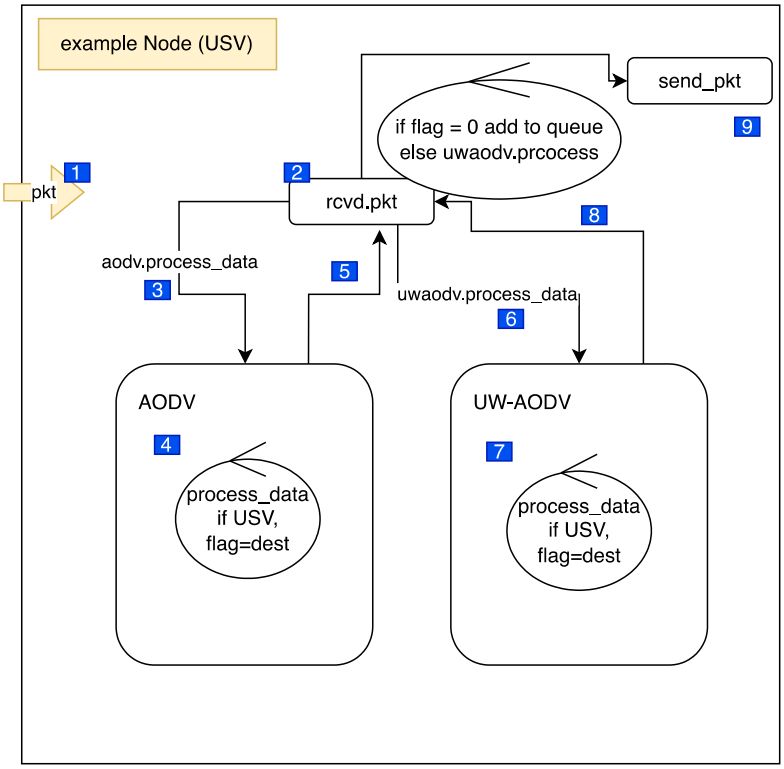


Figure 26: Transition from AODV above-water to UWAODV underwater and from UW-AODV to AODV. The decisive factor is that the flag is set in the respective protocol so that an assignment is made at the node level.

## 4.4 Parameterization and Physical Models of the Simulation

In order for the simulation outcomes to accurately reflect real-world behavior, the MATLAB simulation framework must be parameterized with configurations that represent realistic environmental and system conditions. Given that the scenario involves both above-water and underwater communication segments, the physical parameters of sound velocity, attenuation, range, and bit rate must be defined separately for each domain. The physical communication properties are parameterized in the simulation framework via two central sections of the configuration file: *Phymodel* for radio communication and *Phymodel* for hydroacoustic underwater communication (Figure 23). Both Models make it possible to map domain-specific transmission characteristics realistically and to respond flexibly to different scenarios by specifically setting different parameters. The parameters are derived from the hardware datasheets. For the hydroacoustic subsystem in particular, they were refined in close collaboration with EvoLogics' central development team. The Raspberry Pi version 5 is used for wireless communication (Raspberry Pi Ltd,

2025), while the Evologic S2CR 18/34 hydroacoustic modem is used for hydroacoustic communication (EvoLogics, 2014). Before introducing the parameter values in detail, the hardware components used for simulation parameterization are presented. This includes a description of the individual elements derived from practical field tests and manufacturer data sheets. Where applicable, it also includes the assumptions or simplifications made. Since the physical test setup consisted of a reduced system with only four nodes, the simulation is adapted to replicate this minimal configuration. Therefore, the network topology in the simulation reflects the same node setup used in the practical evaluation. Subsequently, the simulation's routing logic is compared to the routing behavior observed during the physical tests. This comparison validates the extent to which the simulation reflects real-world behavior, demonstrating how effectively the implemented models align with the actual system's constraints and dynamics. The following Figure 27 illustrates the MATLAB simulation. It shows the four nodes that are also replicated in the real-world demonstrator.

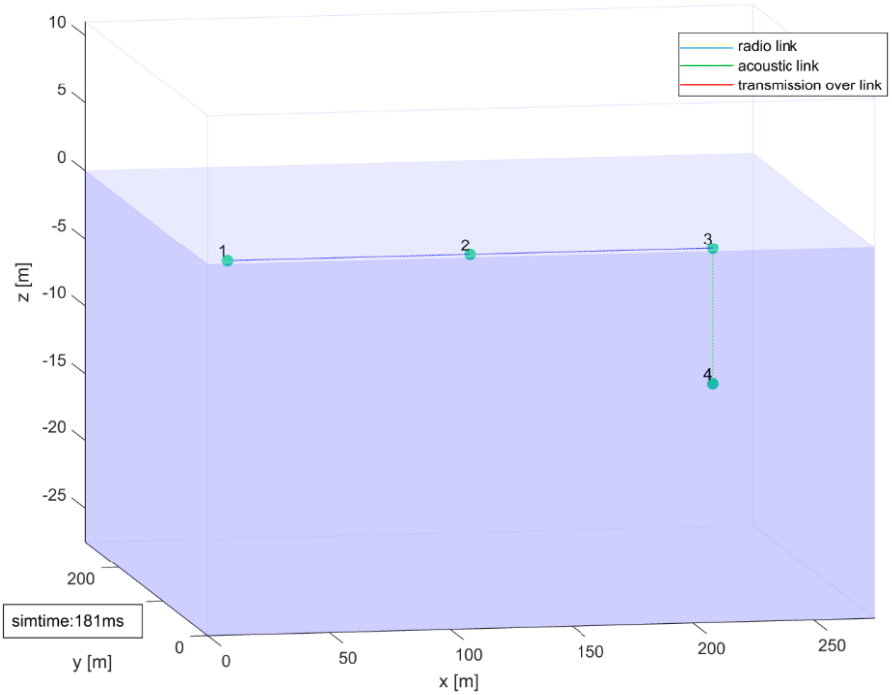


Figure 27: Customized MATLAB simulation for parameterization with the practical demonstrator. In accordance with practical deployments, the setup consists of four nodes: a transmitter, a Hop, a USV with gateway functionality, and a UUV connected to the USV via a hydroacoustic link.

Figure 28 and Figure 29 are presenting the setup of the practical demonstrator. It consists of five Raspberry Pis, which will hereafter be referred to as Pis: One transmitter, one Hop,

one receiver, and one listener Pi. The receiver and listener are each connected to the hydroacoustic modem. The drone in the middle serves to lift the Hop Pi into the air as a Relay to increase range.

Figure 28: Overview of the drone used with a Raspberry Pi as a Relay Node (photograph by the author).

In the practical demonstrator, the receiver Pi refers to the Pi corresponding to the hydroacoustic modem 1 in Figure 30, while the listener Pi corresponds to the hydroacoustic modem 2. The Receiver Pi is responsible for receiving data packets from the radio domain and forwarding them to the hydroacoustic domain. The listener Pi then verifies that the data has been received correctly and measures the time taken for transmission.


Figure 29: Overview of hardware components: The drone in the middle, five Raspberry Pis and two hydroacoustic modems (photograph by the author).

Figure 30 illustrates the test setup in detail in the sequence diagram. The three right-hand Pis are the Pis, which set up routing over-water within the WLAN radio range. For the transition from radio to acoustic, the Raspberry Pi is connected to the acoustic modem via an Ethernet interface. According to the internal routing logic, they identify the Pi/Gateway connected to the modem via the messages RelReq and RelRep. The Pi, connected to the hydroacoustic modem 1, receives the RelReq that arrives via a Hop and responds with a RelRep that includes its coordinates and the SNR value. In this setup, the UUV refers to the underwater node represented by the Raspberry Pi that is connected to the hydroacoustic modem 2 (Figure 30). According to the external routing logic, the Relay echo message is sent via hydroacoustic modem 2 to the USV (receiving Pi). The receiving Raspberry Pi is connected directly to hydroacoustic modem 1 (Figure 30). The SNR of this Raspberry Pi is calculated using the received RSSI and the noise level supplied by the hydroacoustic modem. This information is transmitted to the transmitter with the RelRep. The transmitter can then make a routing decision based on the geographical coordinates and signal quality. As shown in Figure 30, there are two parameterized parts. First is the above-water communication part. This consists of the processing time of the Pis, the clock time offset, and the comparison of the route establishment time with the simulation time. The second part is the underwater parameterization. This is achieved by comparing the runtimes of the Data and Relay Echo message types. Afterwards, the SNR value from this comparison is calculated. Finally, all values are compared with the MATLAB simulation.

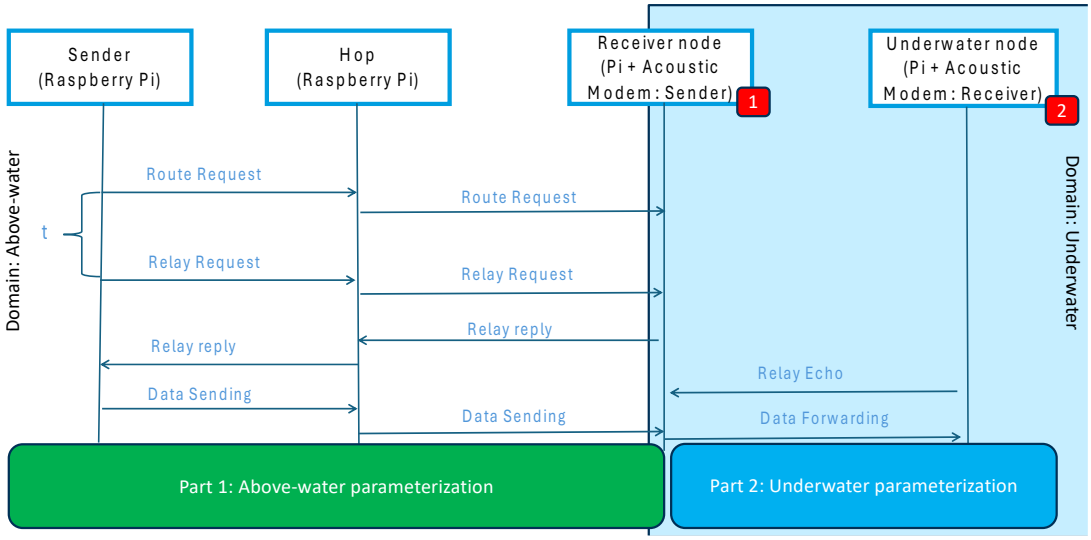


Figure 30: Sequence diagram illustrating the hardware configuration of four Raspberry Pis. These are designed for the surface and underwater domains. The receiver Pi is connected to modem 1 and the hydroacoustic listener is connected to hydroacoustic modem 2.

The deployed Raspberry Pis handle multiple processes simultaneously. These include the classic AODV message types, such as RREQ and RREP, as well as the extended message types, RelReq and RelRep. They are also responsible for data transmission. To enable parallel handling of the different message types, the implementation is executed entirely in user space using Python. Multiple receiving and sending routines run concurrently via Python threads, while inter-node communication is realized through User Datagram Protocol (UDP) sockets. This allows efficient message exchange within the Ad-Hoc Network without requiring kernel-level modifications.

Development and testing were performed in Visual Studio Code, and the Raspberry Pis operate in Ad-Hoc mode during all experiments. The following Pseudocode in Figure 31 describes the handling of the extended message types RelReq and RelRep. On the sender's side, a RelReq is transmitted once a predefined timer has expired. Before sending the message, the Hop counter is incremented by one to indicate that the message has traversed an additional node. The message is then serialized, meaning that the internal data structure of the RelReq object (such as source ID, destination ID, Hop count, and payload) is converted into a byte stream that can be transmitted over the network.

**Algorithm 1** Relay Request / Reply Handling

```
1:  procedure SendRelayRequest(relq)
2:      relq.hops ← relq.hops + 1
3:      message ← Serialize(relq)
4:      Broadcast(message, port=60003)
5:  end procedure
6:  procedure ReceiveRelayRequest(relq)
7:      message ← Receive(port=60003)
8:      if self.NodeType = Gateway then
9:          reply ← CreateRelayReply(message)
10:         Send(reply, nextHop)
11:         ForwardRelayRequest(reply)
12:     else
13:         ForwardRelayRequest(message)
14:     end if
15: end procedure
```

Figure 31: Pseudocode for Relay Request handling. Relay replies are only answered and forwarded by USVs.

This serialized message is then broadcast on port 60003, which is reserved for Relay-related communication within the Ad-Hoc Network. This ensures that all neighboring

nodes listening on this port can receive the message simultaneously. In parallel, a dedicated thread on each node remains active, waiting to receive incoming RelReq messages on port 60003.

Upon reception, the node first checks whether it is the type USV. The USV serves as the designated gateway node within the Multi-Domain scenario. In that case, the relevant RelRep is created and sent back to the node that requested it. Unless the node is a gateway, it does not terminate the process. Instead, it forwards the RelReq message to its neighbors, ensuring that it propagates across multiple Hops until it reaches a suitable gateway. Once the route has been successfully established, the system proceeds with the actual data transmission. The following Pseudocode in Figure 32 illustrates this process in more detail. To achieve this, a socket is bound to the Raspberry Pi's local IP address. In this context, binding a socket means that the program explicitly reserves a specific combination of IP address and port number for sending data.

**Algorithm 2** Data Reception and Acoustic Modem Forwarding

```
1:  procedure ReceiveData
2:      Bind UDP socket to (192.168.1.15, 60000)
3:      while True do
4:          data ← Receive UDP packet
5:          Log packet as recv
6:          Put data into FIFO queue
7:      end while
8:  end procedure
9:  procedure SendToAcousticModem
10:     Connect TCP socket to (172.18.1.110, 9200)
11:     while True do
12:         if FIFO queue not empty then
13:             data ← FIFO.dequeue()
14:             Build acoustic frame: AT*SENDIM + data
15:             Send frame to modem
16:         end if
17:     end while
18: end procedure
```

Figure 32: Pseudocode of the software sequence on the receiver Pi. This part is responsible for receiving the data packet after the route over-water has been successfully established. The data packet is placed in a FIFO memory and transferred to the hydroacoustic modem for transmission.

This prevents interference from other applications and enables the underlying operating system to direct incoming acknowledgements or responses to the correct process. Once the socket has been established, the outgoing data packet is placed in a first in, first out

(FIFO) buffer. This acts as a temporary storage area, guaranteeing that messages are forwarded in the same order in which they were generated. This ordering is essential for preserving the logical sequence of higher-level data flows and avoiding inconsistencies at the receiving end. The packet is then passed from the FIFO buffer to the hydroacoustic modem, illustrated in Figure 32. The modem interface requires a specific, proprietary command structure. In this case, the *AT*SENDIM* command is used to send an *instant message* via the hydroacoustic modem's firmware. Along with this command, the serialized data payload is sent to the modem. The FIFO buffer is then emptied and the transmission logic continues to handle subsequent packets according to the defined protocol sequence.

### 4.4.1 Parameterization for Radio Communication

The Raspberry Pi 5 is chosen for the practical field tests because of its cost-effectiveness and flexible network configuration, which is enabled by its Linux-based operating system. Each unit supported dual-band Wi-Fi (802.11 b/g/n) and Bluetooth 5.0, and ran on a lightweight, 64-bit Linux distribution called Raspberry Pi OS Lite (Raspberry Pi Ltd, 2025). This setup enables the flexible and direct configuration of network interfaces. Additionally, the compact form factor and the ability to be powered by mobile power banks make Raspberry Pis ideal for use on mobile platforms, such as drones (Figure 28). Raspberry Pi devices are configured by default to connect to existing wireless networks in managed mode.

For this project, however, a decentralized, infrastructure-independent communication network is required. Thus, all wireless interfaces were reconfigured to operate in Ad-Hoc mode. This allows the nodes to discover and communicate with each other directly, eliminating the need for an access point. To prevent configuration conflicts, the default Network Manager is disabled. Furthermore, the wlan0 interface is manually configured using the *ifupdown* manager. Each device is assigned a static IP address within the 192.168.1.0/24 subnet. Wireless mode is set to Ad-Hoc, and all devices shared the same SSID (MDO) and channel (channel 2 at 2.412 GHz).

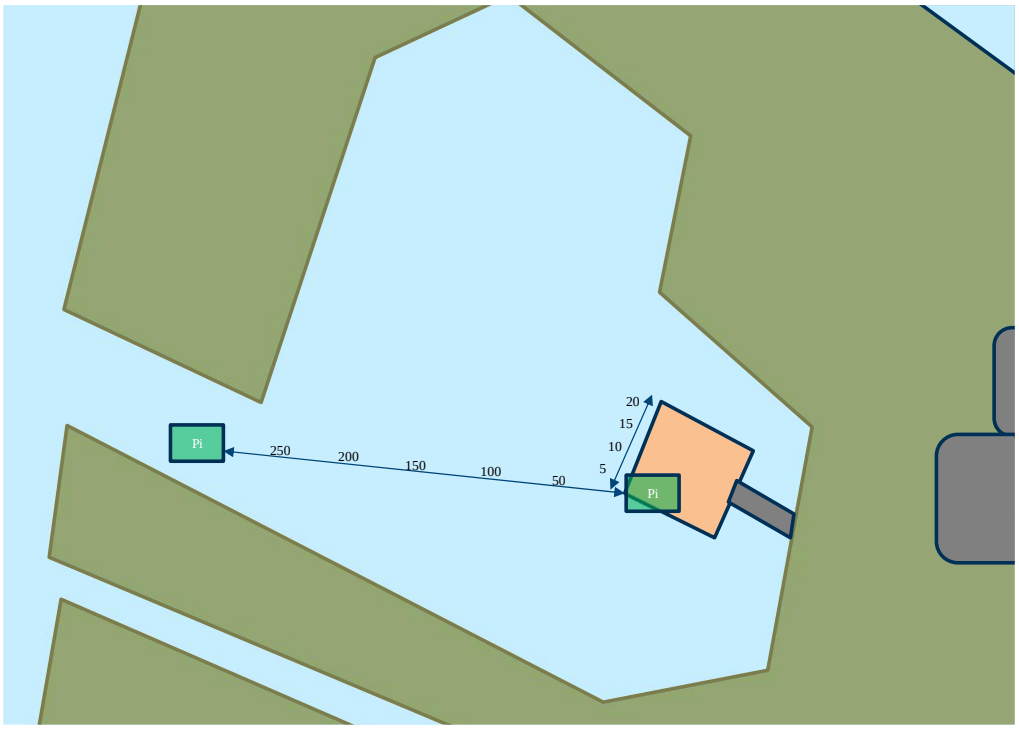


Figure 33: Overview of the test site, with experiments conducted on the south side of the floating pontoon. All measurements are in meters. The distances at the pontoon (5 m–20 m) are the distances for the underwater (acoustic) test; the distances to the estuary (50 m–250 m) are the distances for the above-water (radio) test.

This configuration ensures deterministic addressing, domain-wide visibility, and synchronization within the radio communication domain. To obtain an initial estimate of the radio range, the classical Friis equation under free-field conditions using typical WLAN parameters from commercial chip specifications is applied. The resulting theoretical maximum distance is approximately 5.6 kilometers, representing an idealized value under perfect conditions. Since this approach does not account for multipath propagation, interference, or obstacles, practical field tests were conducted to determine more realistic ranges. To determine the maximum effective distance between two Raspberry Pi nodes in a wireless Ad-Hoc setup, practical field tests were conducted at test facility of TKMS ATLAS ELEKTRONIK GmbH (Figure 33). The objective was to empirically determine the transmission range between two nodes under realistic outdoor conditions with minimal interference and a clear line of sight. Twenty measurements were recorded for each of the predefined distances: 50 m, 100 m, 150 m, 200 m, and 250 m. For each distance the packet delivery success rate was recorded. Based on these results, illustrated in Table 15, a stable and reliable connection could be maintained up to approximately 200 meters.

At 250 meters, however, packet loss increased significantly, resulting in an unreliable communication link.

Table 15: Range test of the Raspberry Pi Hop to Hop wireless communication. A range of 200 m is achieved with a 90% success rate. It should be noted that this is on the water surface where interference is minimal and there are no high waves on the lake.

| **Number of tests** | **Distance (m)** | **Successful transmissions** | **Success rate (%)** |
|---|---|---|---|
| 1 | 50 | 20/20 | 100 |
| 2 | 100 | 20/20 | 100 |
| 3 | 150 | 18/20 | 90 |
| 4 | 200 | 18/20 | 90 |
| 5 | 250 | 6/20 | 30 |

The following parameters were assumed for the practical demonstration of over-water communication: A maximum communication range of 200 meters, a carrier frequency of 2.4 GHz, a bit rate of 10 Mbit/s per second, and a coding rate of 1/2, as illustrated in Table 16. The coding rate is the ratio of useful data bits to the total number of transmitted bits, including error correction bits. A coding rate of 1/2 means that for every data bit, an additional parity bit is transmitted. While this reduces the effective data rate to 5 Mbit/s, it improves transmission robustness, especially in noisy or interfered environments. The simulation neglects propagation delay for radio communication because electromagnetic waves at 2.4 GHz propagate at nearly the speed of light ($3 \cdot 10^8$ m/s). Over a distance of 200 meters, the delay is less than one microsecond and is therefore negligible within the simulation's time resolution.

In summary, the parameterized values of radio communication are used for the physical models of the simulation, as shown in Table 16. Although the signal propagation delay of radio communication can be neglected, two types of delays occur when using Raspberry Pis that must be considered sources of measurement error and accounted for when evaluating results.

Table 16: Configuration of radio communication for parameterized simulation.

| Parameter | Value | Unit | Comment |
|---|---|---|---|
| Communication range (d) | 200 | meters | Based on field tests |
| Frequency | 2400 | MHz | (IEEE 802.11 b/g/n) |
| Bit rate | 10 | Mbit/s | Maximum under ideal conditions |
| Coding rate | 1/2 | - | Simple forward error correction |

Measurement of the Processing and Offset Time of the Raspberry Pis

First, there is a timing offset because the Pis must be manually synchronized in Ad-Hoc mode and do not share a common reference clock. Second, there is a processing delay resulting from the time each device requires to handle packet reception and transmission at the application and protocol levels. To quantify these effects, a controlled measurement setup is created.

Three Raspberry Pis were positioned in a triangular formation on a table, each separated by approximately five meters (see Figure 34). With this configuration, the timing offset is measured first, followed by the average processing delay. These measurements were then used as correction factors when interpreting the experimental results. Since Raspberry Pis are not connected to a central time server via Wi-Fi or the Internet, they lack a common reference point for clock synchronization. To address this issue, the sender node periodically broadcasts its local timestamp. This value does not represent the actual wall-clock time, but it serves as a common reference for the network.

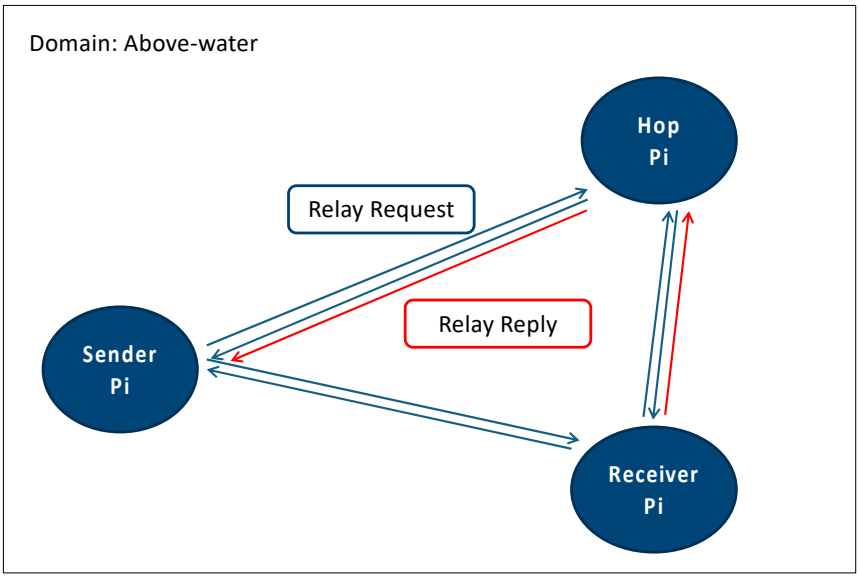


Figure 34: Practical experiment to determine the processing and offset time of the Raspberry Pis. The range between the nodes is negligible due to the speed of light. It is important that they are within communication range.

Upon receiving the broadcast, the recipient measures the message's arrival time and adjusts its internal clock accordingly. In the experimental setup shown in Figure 34, the sender transmits its timestamp to the intermediate Hop, which then synchronizes its local clock. Then, a short control message is sent and the departure and arrival times are recorded via the terminal. The propagation time is disregarded because it corresponds to the speed of light. The Raspberry Pi's processing time is disregarded since the transmitted message is not processed by the receiving Pi. Instead, only the arrival time is logged directly upon reception without additional processing steps. However, since the sender continues to operate for a few milliseconds after completing the transmission, the receiver's clock lags slightly behind. In effect, the receiver operates marginally in the past. The exact measurements are listed in the Appendix, Table 32. The mean value is calculated using the following Equation 4.1:

$$\overline{x_{\text{offset}}} = \frac{1}{40}\sum_{i=1}^{40} x_i \approx 7.73 \text{ ms}. \tag{4.1}$$

The corresponding standard deviation is given by the following Equation 4.2:

$$\sigma_{\text{offset}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x_{\text{off}}})^2} \approx 0.75 \text{ ms}. \tag{4.2}$$

These two values are essential for adjusting the measured underwater transmission times. The average delay is added to compensate for the timing offset induced by the hardware, and the standard deviation is considered when evaluating measurement uncertainty.

The next parameter to be determined is the processing delay, which is the time it takes for a message to travel from one node to the next. This delay is influenced by the internal routing logic implemented in Python, which handles communication between Kernel-Threads and processes via sockets. To quantify the processing delay per Hop, a practical test is conducted using two Raspberry Pis within direct communication range of each other. The sender Pi transmitters a RREQ message to the Hop Pi (Figure 34), which is configured to immediately send the received RREQ back to the sender without delay or processing logic changes. The arrival time of the returned RREQ is recorded on the sender side. Because the RREQ is generated by the sender, this setup enables a round-trip time (RTT) measurement without requiring an additional timestamp exchange. Since the propagation delay in over-water radio communication is negligible due to short distances and high transmission speeds, the processing delay of a single node is approximated as half the measured RTT. This delay includes the time required for the internal processing of the AODV implementation, which is programmed in Python with multithreading and socket-based communication. However, these mechanisms introduce small but measurable latencies due to queuing, thread scheduling, and user-space message handling. This test determines the average processing delay, which is then integrated into the simulation framework to account for the realistic delays introduced at each Hop in a practical deployment. The mean processing delay is calculated as follows:

$$\overline{x_{\text{proc}}} = \frac{1}{40}\sum_{i=1}^{40} x_i \approx 4.59 \text{ ms}. \tag{4.3}$$

The corresponding standard deviation is given by the following Formula 4.4:

$$\sigma_{\text{proc}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \overline{x_{\text{proc}}}\right)^2} \approx 0.42 \text{ ms}. \tag{4.4}$$

To accurately account for total measurement uncertainty, the two error components, clock offset and processing delay, are combined using Gaussian error propagation:

$$\sigma_{\text{total}} = \sqrt{(0.42)^2 + (0.75)^2}\ \text{ms} \approx 0.86 \text{ ms}. \tag{4.5}$$

The coefficient of variation (CV) is used to evaluate the variability in processing time and delay offset among Raspberry Pis. CV is defined as the ratio of the standard deviation to

the mean and provides a unitless measure of dispersion. Lower CV values indicate greater measurement stability. A widely used CV cutoff value is 15% (Shechtman, 2013).

The CV for measuring the processing time is as follows:

$$CV_{\text{proc}} = \frac{\sigma_{\text{proc}}}{\overline{x_{\text{proc}}}} \cdot 100\% = \frac{0.42}{4.59} \cdot 100\% \approx 9.15\%. \quad (4.6)$$

And for the measurement of the offset time the CV is given by the Equation 4.7:

$$CV_{\text{offset}} = \frac{\sigma_{\text{offset}}}{\overline{x_{\text{offset}}}} \cdot 100\% = \frac{0.75}{7.73} \cdot 100\% \approx 9.70\%. \quad (4.7)$$

A CV below 10% means the standard deviation is less than one-tenth of the mean. This reflects low relative dispersion, suggesting the measurements are precise and highly reproducible. The CV of nearly 10% can be attributed to the behavior of the Raspberry Pi system, which is influenced by internal processing, Python threading, socket communication, and associated overhead.

Measurement of the Route Establishment Time in Radio Communication

To quantify the real-world performance of the implemented AODV routing protocol, two time-related parameters were measured in an over-water communication scenario with Raspberry Pis. The distance between the Pis can be disregarded here, as the speed of light does not measurably affect the transmission time. The exact measurement results can be found in Table 33 in the Appendix.

1. The time required to establish a route is measured from the time a RREQ is sent until the corresponding RelRep is received. This time is also known as the Route Establishment Time.

2. The time interval between receiving the RREQ and transmitting the first data packet after establishing the route, named here as Data Establishment Time.

The average time for the Route Establishment Time is calculated as follows:

$$\overline{x_{\text{est}}} = \frac{1}{40}\sum_{i=1}^{40} x_i = \frac{2273}{40} \approx 56.83 \text{ ms}. \quad (4.8)$$

The standard deviation amounts to:

$$\sigma_{\text{route}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x_{\text{route}}})^2} \approx 0.90 \text{ ms}. \quad (4.9)$$

The coefficient of variation is calculated as:

$$CV_{\text{est}} = \frac{0{,}90}{56.83} \cdot 100\% \approx 1.58\%. \tag{4.10}$$

The average time for the Data Establishment Time has a value of:

$$\overline{x_{\text{data}}} = \frac{1}{40}\sum_{i=1}^{40} x_i = \frac{2518}{40} = 62.95 \text{ ms}. \tag{4.11}$$

The corresponding standard deviation is given by:

$$\sigma_{\text{data}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x_{\text{data}}})^2} \approx 1.13 \text{ ms}. \tag{4.12}$$

The coefficient of variation takes the value:

$$CV_{\text{data}} = \frac{1.13}{62.95} \cdot 100\% \approx 1.80\%. \tag{4.13}$$

The time difference between completing the Route Establishment Time and the Data Establishment Time is minimal. This is due to the Raspberry Pi's processing delay, which usually takes about 5–6 ms to prepare and send data after receiving the RREP message. The measured values confirm that this delay is consistent across all trials. The CV for both parameters are relatively low: 1.58% for route establishment and 1.80% for data transmission. A low CV suggests that the values are closely concentrated around the mean, indicating minimal variability in the measurements compared to the mean. In practical terms, this means that the system's behavior is predictable and reproducible, with only minor fluctuations between individual runs.

Table 17: Comparison between simulation and practical measurement of route and Data Establishment time.

| **Parameter** | **Simulation (ms)** | **Real-World Measurement (ms)** |
|---|---|---|
| Route Establishment Time | 48 | $56.83 \pm 0.90$ |
| Data Establishment Time | 64 | $62.95 \pm 1.13$ |

The values obtained from practical measurements and simulations generally agree closely. In the simulation, the Route is fully established after 48 ms. However, data transmission does not occur until 64 ms, illustrated in Table 17. This delay is caused by the

RelRep mechanism that is implemented. Once the first RelRep is received, a 30 ms timer is started to allow for the reception of additional RelReps. Because the timer is only initiated upon the arrival of the first RelReps, the start of data transmission is postponed by the waiting period of the timer. A waiting period of 30 ms is also applied before generating the data packet on the Raspberry Pi implementation. The most noticeable difference is in the Route Establishment Time, where the simulation yields a shorter duration than real-world measurements. This deviation is due to the fact that Raspberry Pi devices execute routing logic in a multithreaded environment. Unlike in the simulation, where operations are performed deterministically and without operating system overhead, real-world execution involves thread scheduling, context switching, and internal data handling, such as message queuing and socket communication. These system-level factors introduce delays of around 2–3 ms, which are very small but measurable. Such delays cannot be fully captured in the simulation model because processing times vary across hardware platforms.

### 4.4.2 Parameterization for Hydroacoustic Communication

The EvoLogics S2CR 18/34 hydroacoustic modem is used as a reference for modeling underwater communication in the simulation (EvoLogics, 2014). This modem operates in the frequency range from 18 kHz to 34 kHz and is designed for medium to long distances in shallow or reverberant waters. A maximum communication range of 1000 meters is used in the simulation. According to the manufacturer, this value corresponds to a typical operating distance with an electrical transmission power of around 2.8 W (EvoLogics, 2020). The data rate is assumed to be approximately 13.9 kbit/s as the gross rate, which corresponds to the maximum transmission value of the modem under ideal conditions. The modems operate in *instant messaging* mode. This operating mode is optimized for quickly and efficiently exchanging small data packets, which significantly reduces protocol overhead. This mode is ideal for scenarios involving sporadic transmissions, such as the exchange of routing or control information in underwater networks. According to *EvoLogics*, operating the modem in *instant messaging* mode requires accounting for a coding factor of 1.3, which reflects the effective data rate. The gross bit rate is 980 bps. The operating frequency remains between 18 kHz and 34 kHz, as specified for the S2CR 18/34 modem. (EvoLogics, 2020)

Additionally, the receiver sensitivity is approximately between -190 dB and -195 re 1 V/ µPa. The source level is 184 dB re 1 µPa at 1 m, which aligns with typical hydroacoustic modem specifications in this frequency range. However, it is important to note that the *EvoLogic* modem can be operated in three levels depending on the range to be achieved. In this work, the modem operates at source level 3, which is used for short distances. The source level must be reduced by 20 dB, resulting in a value of only 164 dB re 1 µPa at 1 m. In addition, a signal gain can be set on the modem. If this is set to low, the modem is reduced by another 20 dB at the source level so that the source level is only 144 dB re 1 µPa at 1 m (EvoLogics, 2020). From this, the transmission value used in the MATLAB simulation as 144 dB re 1 µPa at 1 m can be derived. Likewise, the ambient noise level can be retrieved directly from the modem. Reading the RSSI value at the receiving modem allows one to infer the attenuation loss during transmission. However, the noise level estimation and attenuation loss calculation must both be implemented within MATLAB to enable proper simulation behavior. Additionally, the underwater propagation delay must be realistically represented. This delay is calculated as follows:

Calculation of the Propagation Delay for the MATLAB Simulation:

For the scope of the MATLAB simulation, where depths remain moderate (less than 2 km) and salinity deviations from standard oceanic values are minimal, the full complexity of the model is not required. The detailed formula has already been described in Section 2.2. The simplified model retains the essential temperature, salinity and linear depth dependencies, allowing efficient computation without compromising the relevant physical accuracy. The reduced form of the sound of speed model, as applied in the MATLAB simulation, is expressed as:

$$\begin{aligned} c(T,S,D) \approx\ & 1449.05 + 45.7T - 5.21T^2 + 0.23T^3 \\ & + (1.333 - 0.126T + 0.009T^2)(S - 35) \\ & + D(16.23 + 0.253T). \end{aligned} \tag{4.14}$$

The quadratic depth dependence and the temperature-salinity-depth interaction term have been omitted from the full formulation in this shortened version. Numerical evaluation indicates that these two components contribute less than 0.1% to the total sound velocity under the environmental conditions considered in this study, namely for depths up to 2 km and salinities close to 35 PSU. Their exclusion is made because it allows more efficient simulation while maintaining the necessary physical accuracy for realistic modeling of underwater hydroacoustic propagation. For example, at a distance of 500 m, a water

temperature of 10 °C, a salinity of 35 PSU and an average depth of 300 m, the propagation delay calculated using the simplified sound velocity formula is approximately 0.309 seconds. In comparison, using the full original formula gives a delay of approximately 0.308 seconds, which is less than 0.32% deviation.

Calculation of Noise and Signal Attenuation for the MATLAB Simulation:

An underwater hydroacoustic propagation model is implemented to enable the external routing logic of the extended AODV protocol to determine hydroacoustic signal attenuation and the resulting SNR. The model considers frequency-dependent absorption in seawater and the primary sources of ambient noise. The calculation proceeds in three steps.

a) Absorption Coefficient

Frequency-dependent attenuation is calculated using Thorp's empirical model for absorption in seawater at frequencies in kHz:

$$\alpha_{\text{Thorp}} = 0.11 \cdot \frac{f^2}{1+f^2} + 44 \cdot \frac{f^2}{4100+f^2} + 2.75 \times 10^{-4} \cdot f^2 + 0.003. \quad (4.15)$$

$\alpha_{\text{Thorp}}$ is expressed in dB/km and f is the hydroacoustic frequency in kHz. In the simulation, the frequency of 26 kHz is used. This is the middle frequency range found in the data lattice. It ranges from 18 to 34 kHz.

b) Noise Model

Ambient noise is composed of four dominant components: turbulence noise ( $N_t$) , shiping Noise ($N_s$ ) wind noise ( $N_w$) and thermal noise ($N_{\text{th}}$ ). Each component is calculated in dB and then summed energetically. To account for the signal bandwidth $\Delta f$, an additional term $10\log_{10}(\Delta f)$ is added:

$$N_L = 10\log_{10}(10^{0.1N_t} + 10^{0.1N_s} + 10^{0.1N_w} + 10^{0.1N_{\text{th}}}) + 10\log_{10}(\Delta f). \quad (4.16)$$

c) SNR Calculation

The SNR is obtained from the difference between the source level (in dB re 1µPa at 1m) and the transmission losses (geometric spreading + frequency dependent absorption), minus the total noise level (Stojanovic, 2007):

$$\text{SNR} = P_t - \left(k \cdot 10\log_{10}(d) + \frac{d}{1000} \cdot \alpha\right) - N_L. \quad (4.17)$$

Here, *k* denotes the spreading factor (1 for cylindrical spreading, 2 for spherical spreading), and *d* is the transmission distance in meters. This MATLAB function called SNR (Figure 23) can be used to determine the exact SNR value calculated at the USV acting as a gateway. Users are free to insert their own input parameters. For this evaluation, a transmission frequency of 26 kHz is applied. The wind speed *w* is set to 8 m/s because that was the prevailing measured wind condition on the day of the field experiment. The spreading factor is set to 1.5, representing a value between geometric and spherical spreading. The transmission distance is set according to field measurements and ranged between 5 and 20 meters. It is evident that no model can perfectly replicate real environmental conditions. In the real-world demonstrator, additional factors such as ambient noise, signal reflections, platform motion, and other disturbances may occur. In summary, the following values result for the parameterization of underwater communication, illustrated in Table 18:

Table 18: Configuration properties of hydroacoustic communication for parameterized simulation.

| **Parameter** | **Value** | **Unit** | **Comment** |
|---|---|---|---|
| Communication range (d) | 1000 | meters | Based on 2.8 W transmit power (EvoLogics, 2020) |
| Frequency | 26 | kHz | In IM mode (EvoLogics, 2020) |
| Bit rate | 968 | Hz | In IM mode (EvoLogics, 2020) |
| Coding rate | 0.3 | - | Models simple forward error correction (FEC) (EvoLogics, 2020) |

Validation of the Internal and external Routing Logic (Relay Echo Message and Data Packet Underwater)

This experiment examines two aspects of underwater data packet transmission. First, the propagation time of the Relay Echo message sent from the UUV to the USV is measured, followed by the propagation time of the data packet. As illustrated in Figure 35 and recorded using a hydrophone, the underwater node (Raspberry Pi) connected to the hydroacoustic modem 2 (Figure 30) transmits the Relay Echo message twice. The receiver

Raspberry Pi, connected to the hydroacoustic modem 1 (Figure 30), receives this message. Upon receiving it, the receiver sends a Relay Reply to the sender (radio communication), who then transmits the data packet. The packet is then forwarded from the Raspberry Pi receiver underwater via the first hydroacoustic modem to the underwater node connected to the second hydroacoustic modem. As shown in Figure 35, the tests were carried out at distances of 5, 10, 15, and 20 meters.

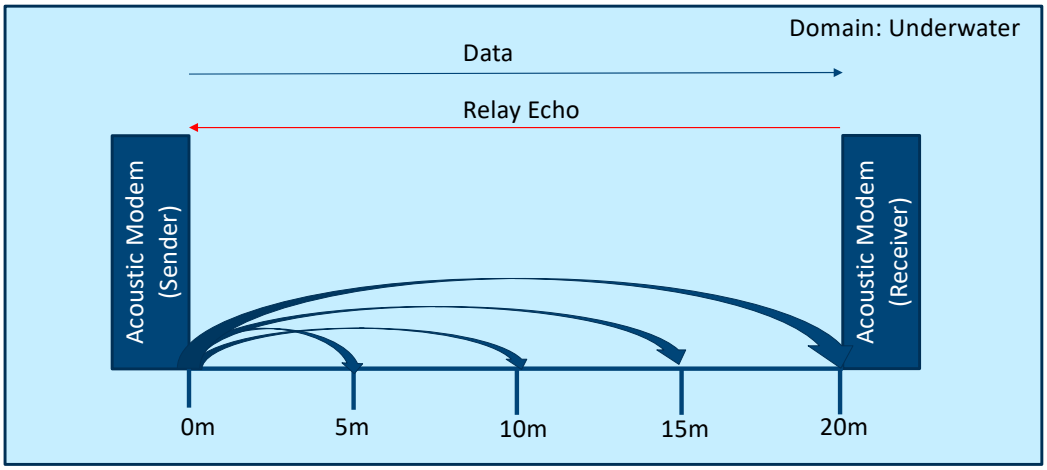


Figure 35: The range for data traffic and transmission of the Relay Echo in the hydroacoustic channel is established for distances from 5m to 20m.

For the underwater hydroacoustic measurements, a hydrophone is deployed to monitor the transmitted messages. Figure 36 shows two short Relay Echo messages followed shortly by a slightly larger Data message. The tests were conducted at the *Atlas lake.* The 8 m/s wind conditions observed during the measurements were incorporated into the simulation. The recorded parameters include the RSSI, ambient noise level, and SNR. The SNR can be directly calculated on the Raspberry Pi. The RSSI value is provided by the *EvoLogics* modem in the acknowledgment message. When the acknowledgement flag is enabled, upon successful delivery, the transmitting modem generates a confirmation containing the bearing and position of the target. In addition to transmitting data, the hydroacoustic modem can measure the ambient noise level when switched to noise mode. The corresponding commands for this mode can be obtained from EvoLogics (2020).

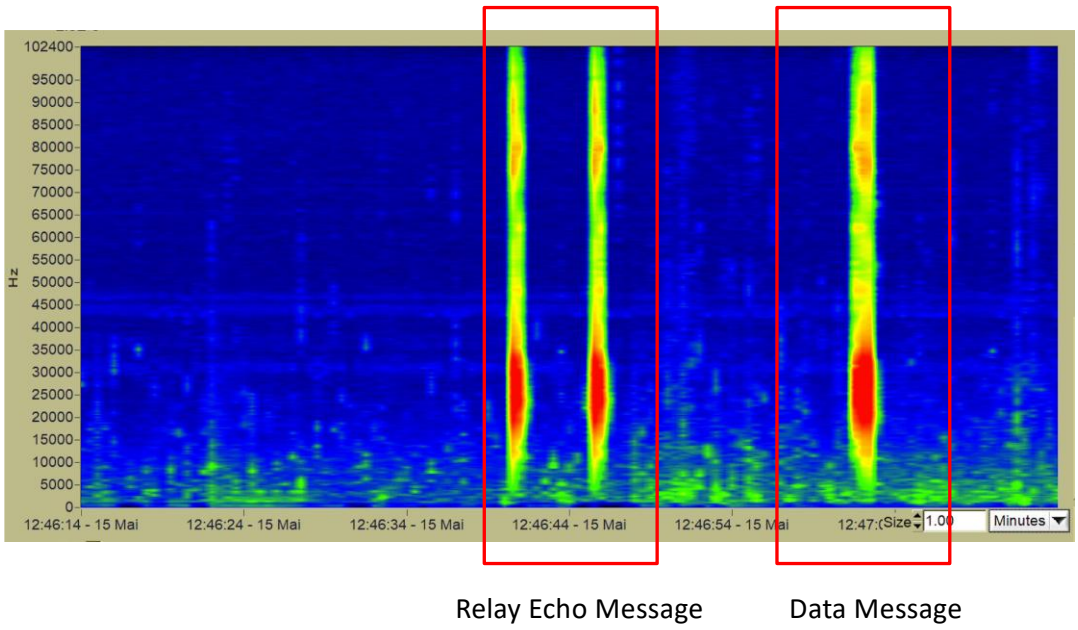


Figure 36: Recording of a hydrophone during Route establishment of the external logic of the extended AODV. The Relay Echo is sent twice for security reasons. The Route above-water is established and the data packet is transmitted (photograph by the author).

For the purposes of the experiment, the Relay Echo message is configured with an 8-byte payload size on the Raspberry Pis, which represents the minimum message size. For example, information on the current battery status is transmitted. According to *EvoLogic*, the protocol overhead amounts to 25 bytes and cannot be modified in *instant messaging* mode. The total message size for the data packet, including the payload, protocol overhead, and additional headers, is 45 bytes and for the Relay Echo message 33 bytes.

To measure the latency of the Relay Echo message, the mean and standard deviation were calculated from ten measurements at each distance, illustrated in Table 19. The standard deviation is affected by systematic errors resulting from the processing time of the Raspberry Pis and the clock offset of the devices. The exact measurement values are provided in the Appendix (Table 35). This table contains the latency results (mean value), the standard deviation, the standard deviation adjusted by Gaussian error propagation. The percentage deviation is calculated to compare the difference between practical measurements and simulation results. In this section, the simulation value is used as the reference since the parameterization is based on the simulated model. In the case of negative values, this indicates that the practical measurements are lower than the simulation results. The percentage deviation is computed using the formula:

$$\Delta_{\%} = \frac{X_{\text{Test}} - X_{\text{Ref}}}{X_{\text{Ref}}} \cdot 100. \quad (4.18)$$

Here, $X_{\text{Ref}}$ denotes the simulation value, while $X_{\text{Test}}$ represents the corresponding practical measurement. The sender Pi broadcasts its timestamp. Meanwhile, the receiver Pi connected to hydroacoustic modem 1 and the underwater node Pi connected to hydroacoustic modem 2 operate based on the same reference timestamp (Figure 30). Therefore, the measured delay of 7.73 milliseconds does not need to be added. However, the measurement uncertainty caused by timestamp deviation (Equation 4.2) must be included in Gaussian error propagation.

Table 19: Results of the runtime of the Relay Echo message at various distances.

| **Distance (m)** | **Mean Latency (ms)** | **Std. Dev. (ms)** | **CV in %** | **Std. Dev. (error propagation in ms)** | **Latency (Simulation in ms)** |
|---|---|---|---|---|---|
| 5 | 348.10 | 2.08 | 0.60 | 2.21 | 360 |
| 10 | 349.70 | 1.83 | 0.52 | l.98 | 364 |
| 15 | 352.00 | 2.54 | 0.72 | 2.65 | 368 |
| 20 | 356.60 | 1.65 | 0.46 | 1.81 | 370 |

The comparison in Figure 37 reveals that the practical measurements exhibit a trend similar to the simulation results, with deviations primarily within the measured standard deviation range. The maximum difference between the practical measurements and the simulation is 16.00 ms at 15 m, corresponding to a relative deviation of −4.35% with respect to the simulation value. This difference falls within an acceptable range for the current parameterization of the simulation in this scenario, suggesting that the model accurately reproduces the observed behavior.

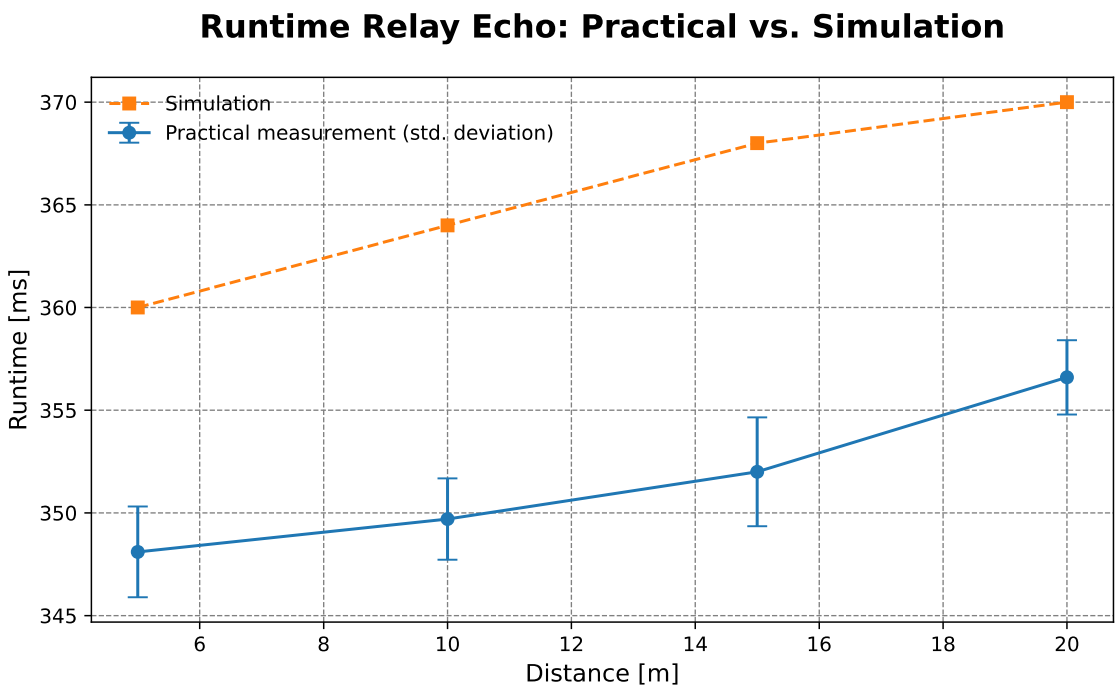


Figure 37: Runtime of the Relay Echo packet over different distances in direct comparison with the simulation.

According to the external routing logic (Figure 16), the Relay Echo message is the message sent by the UUVs to calculate the SNR at the USV gateway. This value is then returned to the sender via the RelayRep and is used for gateway selection.

Next, a data message is transmitted first to Acoustic Modem 1 with Receiver Pi (Figure 30), then forwarded to Acoustic Modem 2. However, the data message is larger, containing 20 bytes, while the Relay Echo has a size of 8 bytes (excluding protocol overhead). Consequently, the transmission duration and propagation time are both significantly higher. The measured parameters for this process are latency, standard deviation, deviation including Gaussian error propagation (total deviation of the offset and the runtime) , and coefficient of variation (CV), as presented in Table 20. The entire series of measurements can be found in Table 36 in the Appendix. The latency trend of the data packet between 5 and 20 meters is compared with the corresponding simulation results (Figure 38). The practical measurements closely align with the simulation results, with deviations ranging from 19.50 ms (−4.01% with respect to the simulated value) at 5 m to 5.50 ms (−1.11% with respect to the simulated value) at 20 m.

Table 20: Runtime of the data packets over different distances.

| Distance (m) | Mean Latency (ms) | Std.Dev. (ms) | CV in % | Std.Dev. (error propagation in ms) | Latency (Simulation in ms) |
|---|---|---|---|---|---|
| 5 | 466.50 | 5.30 | 1.13 | 5.35 | 486 |
| 10 | 472.10 | 7.88 | 1.67 | 7.92 | 490 |
| 15 | 478.20 | 8.40 | 1.76 | 8.43 | 494 |
| 20 | 490.50 | 7.22 | 1.47 | 7.26 | 496 |

The standard deviation is slightly higher in the practical results, likely due to the longer runtime leading to greater variability. Furthermore, at 20 meters, the practical measurements and simualtion values converge significantly.

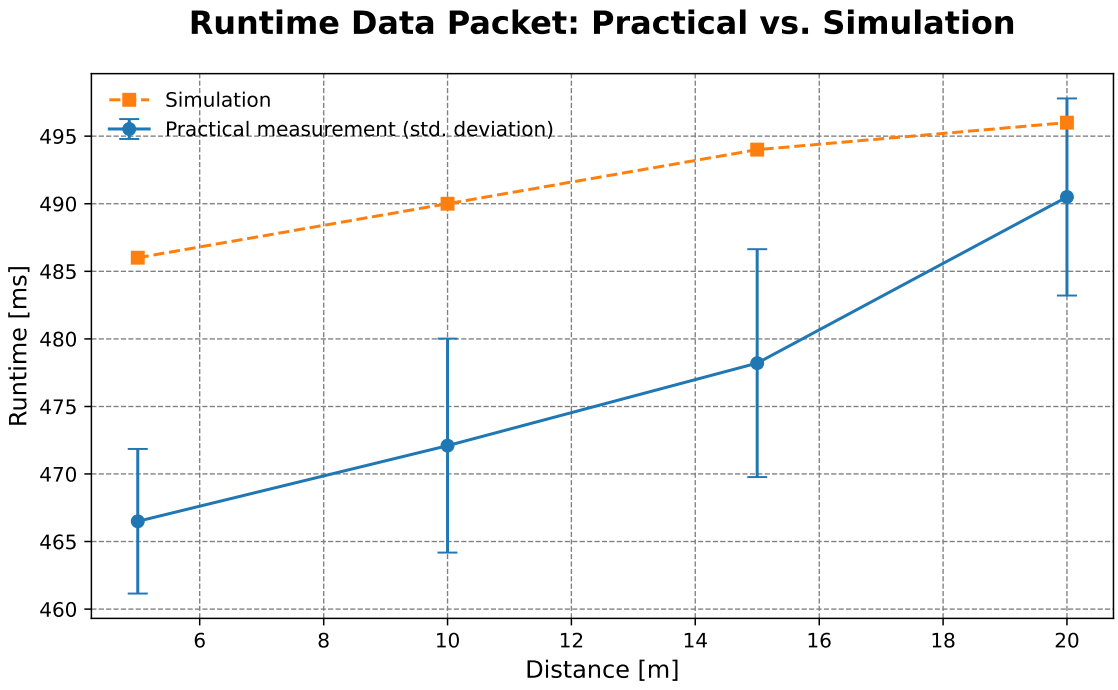


Figure 38: Comparison of the data packet's runtime over various distances with the simulation.

Validation of the SNR

In analyzing the RSSI from the acoustic modem, the *EvoLogics* modem used in this study reports the received voltage level as a logarithmic measurement in decibels referenced to one volt (dB re 1 V). This is a logarithmic representation of a linear voltage measurement, calculated according to:

$$L_V = 20 \log_{10} \left( \frac{V}{1\,\mathrm{V}} \right), \tag{4.19}$$

where V is the measured voltage. The hydrophone sensitivity S is specified in decibels relative to 1V/µPa (dB re 1V/µPa), defining the ratio between the output voltage and the incident sound pressure.

Since both quantities use compatible reference units, the received sound pressure level (SPL) can be determined directly in the decibel domain (Garcia Benadí, 2014), (Benadí et al., 2015):

$$\mathrm{SPL}_{(\mathrm{dB\,re\,1\,\mu Pa})} = V_{(\mathrm{dB\,re\,1\,V})} - S_{(\mathrm{dB\,re\,1\,V/\mu Pa})}. \tag{4.20}$$

In an example measurement, the distance between the two modems is 10 meters. The received signal is specified with a RSSI value of -65 dB re 1 V. The sensitivity of the hydrophone used is -197 dB re 1 V/µPa. The following formula can be used to calculate the sound pressure level (SPL) in dB re 1 µPa:

$$\mathrm{SPL} = -65 - (-197) = 132\ \mathrm{dB\ re\ 1\ \mu P}. \tag{4.21}$$

According to the *EvoLogic*, the modem's sensitivity specification should not be considered a calibrated measurement reference. The built-in hydrophones operate within a nominal sensitivity range of approximately −184 dB re 1 V/µPa to −202 dB re 1 V/µPa. For this practical experiment, it is assumed that the effective sensitivity must be verified through dedicated measurements to determine the value that most closely matches the simulation results. To evaluate the SNR, the measured value at Acoustic Modem 1 (Figure 30) on the Receiver Pi when the Relay Echo message arrives is considered. This value is stored temporarily and sent back to the sender via the Relay Reply. According to the routing logic, the RelRep is also sent when no SNR entry is available. In this case, routing proceeds based on internal logic and coordinates. The SNR is then evaluated for distances ranging from 5 m to 20 m.

Since the SNR measurement is independent of processing time and unaffected by potential timestamp deviations, these factors can be disregarded. In Section 3.3.3, the SNR is theoretically derived as the difference between the source level and the sum of the total transmission loss (geometric spreading and frequency-dependent absorption) and the total noise level. In practice, however, the RSSI directly represents the source level minus the total transmission loss. Consequently, the practical computation of SNR can be expressed in a simplified form as follows:

$$\mathrm{SNR} = \mathrm{RSSI} - N_L, \tag{4.22}$$

where $N_L$ is the measured Noise. The RSSI is measured by the receiver Pi at Acoustic Modem 1. The measured values are provided in Table 21. The entire series of measurements can be found in Table 34 in the Appendix. As expected, the results show a consistent decrease in SNR with increasing distance. SNR decreases from 48.40 dB at 5 m to 38.70 dB at 20 m, which is consistent with the increase in transmission loss from 10.40 dB to 19.90 dB.

Table 21: SNR parameter trend over distance, which is used for external routing logic as signal quality-based routing.

| **Distance (m)** | **Pt in dB re 1 µPa at 1 m** | **Noise in dB** | **TL in dB** | **SNR in dB** | **SNR-Std. Dev. (dB)** | **CV in %** | **SNR Simulation in dB** |
|---|---|---|---|---|---|---|---|
| 5 | 144 | 85.20 | 10.40 | 48.40 | 0.70 | 1.43 | 48.78 |
| 10 | 144 | 85.30 | 13.10 | 45.60 | 1.65 | 3.62 | 44.24 |
| 15 | 144 | 85.30 | 16.50 | 42.20 | 0.79 | 1.84 | 41.56 |
| 20 | 144 | 85.40 | 19.90 | 38.70 | 2.06 | 5.32 | 39.65 |

The differences between the measured and simulated SNR values remain small (less than 1.5 dB), which confirms the accuracy of the simulation model for short-range underwater hydroacoustic links. The low standard deviation values (0.70–2.06 dB) and coefficients of variation below 10 indicate stable channel conditions and high measurement repeatability.

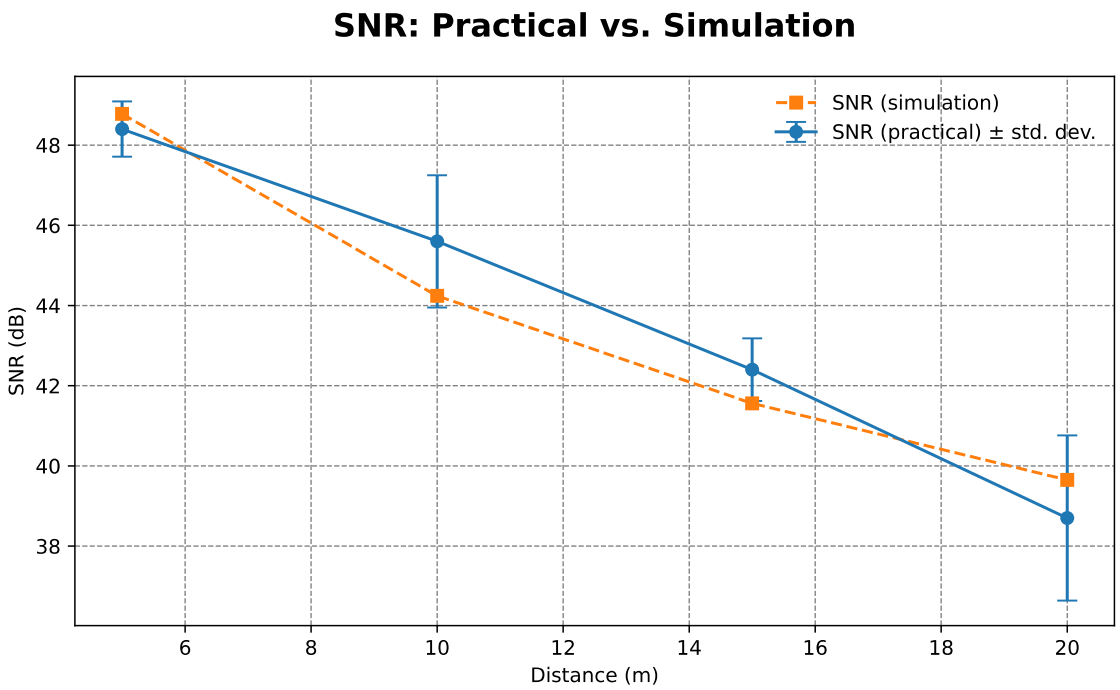


Figure 39: Graphical progression of the SNR value in direct comparison with the simulation values over distances from 5 to 20 m.

As shown in the Figure 39, the measured and simulated SNR values follow a similar trend across all tested distances. Both gradually decrease as the distance increases from 5 m to 20 m, and the deviation between the curves remains small. This indicates that the simulation model accurately reflects the practical measurements. Including the standard deviation as error bars highlights the variability in the measurements, which increases slightly with distance but remains within an acceptable range.

### 4.4.3 Deviations and Adjustments between Practice and Simulation

During the parameterization of the MATLAB simulation, the physical characteristics of the Raspberry Pi devices and acoustic modems were modeled using hardware specifications and measurement data. The deviations between the simulated and measured results are summarized below.

a) Deviation in the Processing Time of the Raspberry Pi

In the practical measurements, a mean processing time of 4.59 ms per message and a standard deviation of 0.42 ms were measured over 40 samples. This can partly be attributed to the multithreaded implementation, which has separate threads for RREQ,

RREP, RelReq, RelRep and Data. Additional delays are to be expected under higher system load, as the threads cannot execute fully in parallel due to the limited number of CPU cores and associated scheduling constraints. In the MATLAB simulation, a fixed delay of 5 ms is applied to the processing time between nodes.

b) Deviation in Timestamp Offset

In practice, the mean timestamp offset is 7.73 ms with a standard deviation of 0.75 ms. This is the result of the synchronization procedure, whereby one Raspberry Pi broadcasts its local time and the receiving Raspberry Pi accepts this as the reference and performs the processing required to update its clock. These stages introduce a small but measurable delay. The offset is defined in relation to the internal clock of the Raspberry Pi providing the time. Consequently, when calculating the one-way propagation delay between the Pi and a receiving node, the offset must be added to the measured value. However, when considering two Raspberry Pis not directly involved in the time-sharing process, no offset time needs to be accounted for. Nevertheless, this deviation cannot be integrated into the MATLAB simulation because it represents a hardware-specific local offset rather than a property of the communication channel.

c) Deviation in Over-water Route Establishment and Underwater Communication

Upon examining the Route Establishment Time, an approximate deviation of 8.83 ms is observed between the simulation and the practical demonstrator (Table 17). This difference is due to internal processes on the Raspberry Pis, such as operating system scheduling, network stack processing, and driver interactions, which cause slight delays in packet handling. By contrast, the Data Establishment Time (the period from sending the RREQ to the data being transmitted in Table 17) is highly accurate, with a deviation of only about 1.05 ms. In the simulation, a predefined timestamp is processed before data transmission begins. The Raspberry Pis have an identically configured timestamp, which explains the minimal deviation.

Considering the Relay Echo transmission time from Acoustic Modem 2 to Acoustic Modem 1 (see Figure 30), a maximum deviation of 16.00 ms is observed. On average, the simulation yields slightly longer runtimes than the practical measurements. However, this deviation remains within acceptable limits. This can be attributed to the fact that the operating frequency, bitrate, and modem sensitivity parameters were configured according to *EvoLogics'* specifications. Minor fluctuations are expected as the modems operate in

IM mode within a 18-34 kHz frequency band. Figure 38, shows similar results. The transmission time of the data packet from Acoustic Modem 1 to Acoustic Modem 2 indicates that the simulation yields slightly higher runtimes than the practical measurements at all tested distances. The observed deviation ranges from 5.50 to 19.50 ms and remains within an acceptable range. However, it should be noted that the uncertainty in the time base due to Gaussian error is up to 8.44 ms. While this deviation is acceptable, it indicates that the simulation parameters are slightly conservative. For this reason, these values are retained as they were provided by *EvoLogics*, and the resulting deviation remains within an acceptable margin.

Although the increase in Relay Echo and data transmission time does not appear to be perfectly linear, this can be explained by the discrete simulation time step of 2 ms, which introduces quantization effects into the measured runtime. Unlike the Relay Echo packets, the data packets show a greater increase in end-to-end latency with distance. This is mainly due to their larger payload size, the longer acoustic transmission time and occasional retransmissions caused by reduced signal quality over greater distances. Furthermore, the results of the runtime of the data packet illustrate that simulation and measurement converge at larger distances. This behavior is mainly caused by the low bit rate of 968 bps, which results in a transmission time of approximately 480 ms for the 45-byte data packet. In contrast, the distance-dependent propagation delay contributes only about 3 ms at 5 m and approximately 13 ms at 20 m, based on the underwater delay formulas introduced in Chapter 2.1.

As a result, the transmission time is far larger than the propagation delay, which makes the influence of distance relatively small over short ranges. The remaining difference between simulation and measurement appears to be a nearly constant offset caused by modem-internal processing effects that are not part of the physical delay model. As the propagation delay increases with distance, this offset becomes relatively smaller, which likely contributes to the observed convergence at a distance of 20 m.

Examining the SNR value measured at acoustic modem 1 (Figure 39) reveals that the results from the practical measurements and the simulation are closely aligned. The maximum deviation is 1.36 dB, considered very small for this type of measurement. This shows that the physical models in the simulation, which consider attenuation loss and noise, accurately represent real-world conditions. The simulation is configured with the

same wind speed conditions present during the practical measurements to ensure consistent representation of environmental influences.

# 5 Conducting the Simulation for the Use Case Borkum Riffgrund 2

This chapter presents a simulation of the use case for protecting the Borkum Riffgrund 2 offshore wind farm with UxVs. As shown in Figure 5, the scenario is designed so that a UAV establishes a route to a UUV, and a UUV establishes a route to a UAV within a Multi-Domain Ad-Hoc Network. The various routing concepts presented in Chapter 3, including error management, are compared with each other and with the proposed hybrid AODV approach.

## 5.1 Necessary Adjustments in the Simulation

To enable direct comparison with the developed extensions to the routing protocol, the hybrid AODV protocol is implemented as a reference protocol. First, the necessary modifications to the MATLAB simulation framework for the scenario are described. Afterwards, the simulation setup is presented, including spatial node placement and network initialization. Finally, the results are evaluated. The results cover the Route Establishment Time, the number of data packets successfully received, and the Reception Drop Rate.

### 5.1.1 Customization of the Configuration for the Use Case

To model the routing behavior regarding the protection of the Borkum Riffgrund 2 offshore wind farm using UxVs, the simulation domain is set to 2,000 m by 2,000 m. This size is large enough to capture the Multi-Domain characteristics of the scenario, such as multi-hop routing and gateway selection, while ensuring that the computational runtime remains low enough to allow for multiple simulation runs under different parameters. This balance ensures that the results are statistically meaningful across several iterations and are not based on a single configuration. The nodes are placed randomly within the simulation domain. The resulting topology remains constant throughout all experiments to ensure comparability. The scenario incorporates 13 USVs, 5 UUVs and 2 UAVs. The maximum depth is set to -30 m reflecting typical conditions in the Baltic Sea near Borkum.

In the hybrid AODV implementation, a single shared transmission queue is used. Consequently, all message types can be sent via USVs across both radio and hydroacoustic links. However, this coupling allows control and data traffic to propagate into the underwater hydroacoustic domain, which is undesirable if the aim is to minimize utilization of the underwater channel (NFR3). In the extended AODV, the transmission buffers are explicitly separated into radio and hydroacoustic queues. The scheduler applies a simple, deterministic queueing policy (Figure 40): A packet is transmitted over the medium only if (i) the corresponding queue is non-empty and (ii) the respective link is currently free. This decoupling ensures that routine control and data exchanges remain on the radio interface, while the hydroacoustic interface is only used when explicitly required by the routing logic. This minimizes unnecessary underwater transmissions and reduces overload on the hydroacoustic channel. As shown in Figure 41, forwarding using the AODV/UWAODV protocol is preserved, with medium-aware decisions being applied at the queuing stage. Specifically, it prevents the inadvertent Cross-Domain forwarding of RREQ/RREP//RelReq/RelRep/DATA (radio to hydroacoustic or hydroacoustic to radio) unless this is required by the protocol state. This aligns the transmission behavior with the design goal of limiting underwater traffic.

**Algorithm 3** Separation of Radio and Acoustic Queues

```
1: if radioQueue not empty and RadioLink free then
2:     send packet from radioQueue
3:     update counters/bytes
4: end if
5: if acousticQueue not empty and AcousticLink free then
6:     send packet from acousticQueue
7:     update counters/bytes
8: end if
```

Figure 40: Pseudocode for separating the queues. This clearly separates data traffic between the radio and hydroacoustic domains.

In the simulation, each node first classifies an incoming message and assigns it to either the radio or hydroacoustic transmission queue. Once scheduled, the packet is transmitted from the sending node to the receiving node, where it is processed by the network layer within the AODV routing logic. At this stage, a protocol-specific message (e.g. RREQ, RREP, RelReq, RelRep, DATA or RERR) is generated. Furthermore, the routing logic determines whether the packet is explicitly addressed to a USV acting as a domain

gateway. Packets explicitly addressed to a USV gateway have their forwarding flag set to a non-zero value, marking them for Cross-Domain transmission. In practice, this mainly applies when the destination is located in the underwater domain. Upon arrival at the node level, the packet is evaluated by the *received function*. If the flag is equal to 0, the packet remains within the current domain and is forwarded to the corresponding queue (radio or hydroacoustic). Otherwise, the packet is re-processed by the relevant protocol instance (AODV or UWAODV) and forwarded across the other medium. This mechanism ensures that both control and data packets are forwarded in a medium-aware fashion while supporting gateway-based domain transitions as required.

**Algorithm 4** Routing decision via flag (AODV vs. UWAODV)

```
1: if linkType = radio then
2:     pkt ← radioLink.tx_pkt
3:     if pkt.next = AODV then
4:         (pkt_fwd, pkt_rep, flag) ← AODV.process(pkt)
5:         if flag = 0 then
6:             send pkt_fwd, pkt_rep over radio
7:         else
8:             pkt' ← UWAODV.process(pkt); send pkt' over acoustic
9:         end if
10:    end if
11: else if linkType = acoustic then
12:    pkt ← acousticLink.nxt_pkt
13:    if pkt.next = UWAODV then
14:        (pkt_fwd, pkt_rep, flag) ← UWAODV.process(pkt)
15:        if flag = 0 then
16:            send pkt_fwd, pkt_rep over acoustic
17:        else
18:            pkt' ← AODV.process(pkt_fwd); send pkt' over radio
19:        end if
20:    end if
21: end if
```

Figure 41: Pseudocode for implementation at the node level between AODV and UWAODV. If the flag is not equal to zero, the domain is changed at the USV.

Another significant adjustment to the simulation involves the allocation of physical models. As outlined in Section 2.2, the representation of transmission behavior differs between radio and hydroacoustic links. In the over-water (radio) domain, the model only accounts for transmission duration and Raspberry Pi processing time. In contrast, underwater (hydroacoustic) communication includes both the transmission duration and the propagation delay. In the simulation framework, this behavior is realized in the LinkModel through two distinct timers. The first timer, named *until*, is maintained at the

sending node and models the transmission duration of a packet over the respective medium, illustrated in Figure 42. The second timer, named *rx_pkts*, is maintained at the receiving node and models the propagation delay associated with hydroacoustic transmission. Consequently, packet transfer is completed in two stages: (i) the sender's *until* timer expires to represent the end of transmission, and (ii) the receiver's *rx_pkts* timer expires to represent completion of propagation. Both timers are computed within the *PhyModel* class, which encapsulates the physical properties of radio and hydroacoustic links.

**Algorithm 5** Transmission and propagation timers

```
1: if rx_pkts not empty then                                  ▷ acoustic reception only
2:     for all i in rx_pkts do
3:         rx_pkts.PropagationDelay[i] ← rx_pkts.PropagationDelay[i] −t
4:         if rx_pkts.PropagationDelay[i] ≤ 0 then
5:             nxt_pkt ← rx_pkts.Packet[i]; mark i for removal
6:             emit finishedReceiving
7:         end if
8:     end for
9:     remove all marked entries from rx_pkts
10: end if
11: if until > 0 then                                  ▷ transmission duration (any link)
12:     until ← until −t
13:     if until ≤ 0 then
14:         emit finishedSending
15:     end if
16: end if
```

Figure 42: Pseudocode for the LinkModel in the simulation: The propagation timer is the rx_pkts variable and the duration timer is the until variable.

### 5.1.2 Description of the Simulation

Before interpreting the simulation results, the underlying communication setup must be clarified. The routing evaluation focuses on the initial route establishment phase. Specifically, the metric of interest is the time taken by a source node to successfully discover and establish a route to the designated destination. Data packets can only be transmitted once this route discovery process is complete. The focus of the evaluation is only on the initial setup of the route. Afterwards, data traffic can take place. The user decides how long a route is valid (TTL) before a new route is built.

A comparison of three concepts is made in the routing study: The classic hybrid AODV, the extended AODV with the internal routing logic and extended AODV with the external routing logic. In addition, the extended AODV protocol is being tested for subsea routing

between UUVs and UAVs. The adapted error management system is also being tested. In the scenario, a source node (UAV 4) in the air domain initiates route discovery to four underwater destinations (UUV 2, UUV 3, UUV 13 and UUV 15). USVs act as potential gateways for Cross-Domain forwarding. Additionally, the routing logic of UUVs with Cross-Domain transitions is evaluated by examining routing paths that begin at UUV 6 and end at UUV 2, UUV 3, UUV 13 or UUV 15. A USV gateway mediates the domain change. Figure 43 depicts the simulation scenario. For clarity, the nodes are shown at depths of up to 200 m. However, in all simulation runs, the maximum depth is limited to 30 m, in accordance with the use case.

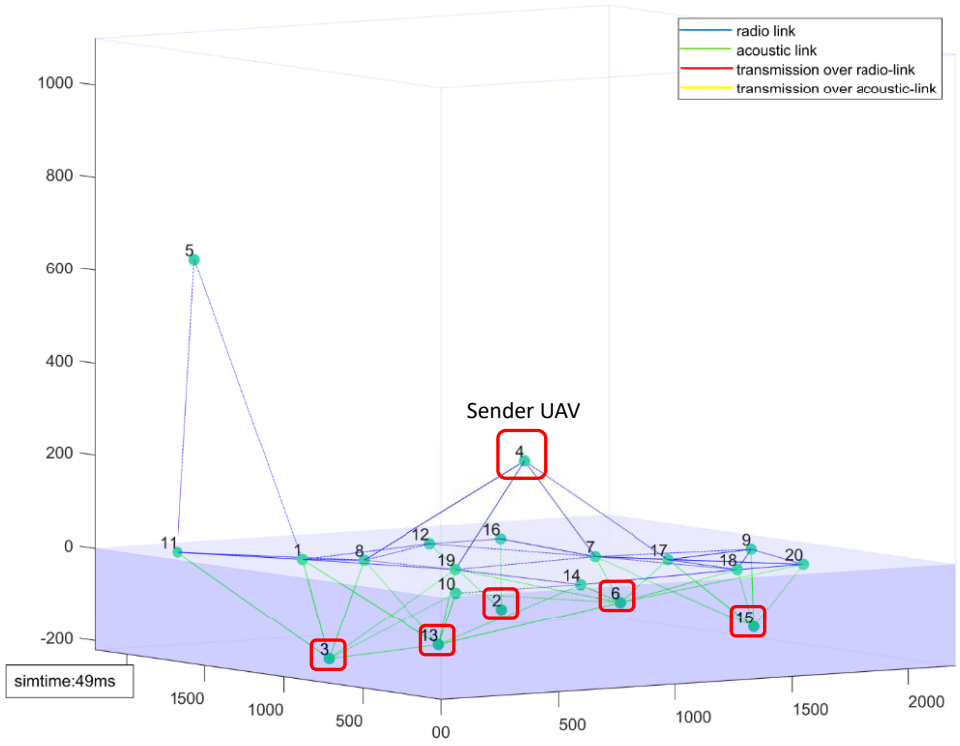


Figure 43: Visual representation of the simulation scenario. A route is established from the transmitting UAV 4 to UUVs 3, 13, 15, and 2, and from the transmitting UUV 6 to the remaining UUVs as well as to UAV 4.

The topology is identical across all routing concepts and node mobility is neglected. In all runs, the time taken to establish a route does not exceed 3 seconds, so transient network dynamics during discovery are negligible and are only introduced in error management, where dynamics and changes in technology are important. Table 22 summarizes the main simulation configuration parameters.

Table 22: Configuration properties of the MATLAB simulation with regard to simulation size. Radio and hydroacoustic channel.

| **Parameter** | **Value** | **Unit** |
|---|---|---|
| Nodes | 20 | - |
| USVs | 13 | - |
| UAVs | 2 | - |
| UUVs | 5 | - |
| Simulation Time | up to 40000 | ms |
| Height (maximum) | 1000 | m |
| Square | 2000 | m |
| Depth (maximum) | -30 | m |
| Range (radio comm.) | 800 | m |
| Frequency (radio comm.) | 2.4 | Ghz |
| Bit rate (radio comm.) | 10 | Mbit/s |
| Range (hydroacoustic comm.) | 1000 | m |
| Frequency (hydroacoustic comm.) | 26 | khz |
| Bit rate (hydroacoustic comm.) | 968 | bit/s |
| Wind | 8 | m/s |
| Water temperature | 20 | °C |

For over-water radio communication, a nominal data rate of 10 Mbit/s within a range of up to 800 m is assumed. While this configuration is ambitious, it remains plausible under line-of-sight conditions with sufficient Fresnel clearance. This is particularly true in maritime environments, where stable long-range links are favored by the absence of major obstacles and external interference. This is particularly the case when gateway nodes are equipped with antennas that provide moderate gain, such as sector or panel antennas. These antennas extend the practical range without requiring strict point-to-point alignment. Crucially, the simulation results remain unaffected by a reduction in the nominal bitrate: Reducing the rate from 10 Mbit/s to 6 Mbit/s increases the transmission duration in the radio domain by only approximately 0.02 ms due to the fixed 5 ms processing offset. Lower-rate long-range standards such as IEEE 802.11ah (Wi-Fi HaLow),

described in (Baños-Gonzalez et al., 2016) and (Alam et al., 2024) would also have a minimal impact on the results, as their airtime contribution for 48-byte control messages remains below 1 ms. Therefore, 10 Mbit/s are retained as a representative nominal value with no impact on the reported timings. The evaluation focuses on the following performance metrics:

Route Establishment Time

The elapsed time at the source is measured from the emission of the initial route request to the successful availability of the route. This includes all the control exchanges required by the respective concept, such as the RelReq and RelRep, as well as the gateway selection logic.

The hydroacoustic-channel load

This is the total number of bytes received by all nodes on the hydroacoustic interface during route discovery. Payload data is excluded to preserve comparability. This metric captures the extent to which the routing procedure stresses the underwater medium, and it is this quantity that should be minimized under NFR3.

Reception Drop Rate (RDR)

The Reception Drop Rate is introduced in this work as a metric to adapt the classical packet drop rate to the characteristics of broadcast-based route discovery. In contrast to the sender-based definition (drops per sent packets), which is not informative in this simulation because the data packets in this simulation typically reach the destination, the RDR is defined from the receiver's perspective and quantifies how many broadcast packets fail to arrive at a node. The objective of this study is to quantify redundancy during the process of establishing a route in an underwater channel. The RDR counts the number of discarded control packets at each node after reception and divides this by the total number of control packets received. The resulting network-wide ratio is indicative of redundant receptions.

Following the analysis of the initial route discovery process, the performance of the error management concept is evaluated. The simulation duration is increased and the node dynamics are enhanced. The evaluation reports on the number of errors detected and resolved, as well as the amount of application data received at the destination, both with and without error management.

## 5.2 Results of the Simulation based on Chapter 3

All relevant simulation metrics are logged by the *simstat* function, as shown in Figure 23, and emitted at the end of each run. Results are visualized and discussed in subsequent subsections. Unless stated otherwise, the physical configurations follow the parameterization defined in Section 4.3. The complete numerical results, including per-destination breakdowns and aggregated statistics, can be found in the Appendix (*Table 24, Table 25, Table 26, Table 27, Table 28, Table 29, Table 30 and Table 31).* In this section, all percentage deviations are computed using Formula (4.18), which has already been applied in Chapter 4. In contrast to the previous evaluations, $X_{Ref}$ now denotes the baseline hybrid AODV, while $X_{Test}$ refers to the extended AODV variants.

### 5.2.1 Hybrid AODV from UAV to UUV: Route Establishment Time

The first simulation evaluates the route establishment time of the hybrid AODV protocol. This metric denotes the time interval between the sender node (UAV 4) broadcasting a RREQ message and the designated target nodes (UUVs) receiving this message. Upon arrival at the destination, a RREP is generated as a unicast and is forwarded back to UAV 4 based on the routing tables maintained along the path. Once the RREP successfully reaches UAV 4, the route is established and ready for data transmission. Therefore, the measured Route Establishment Time covers the entire process from the initial RREQ broadcast until the route is available, just before the first data packet is created and transmitted. It is important to note that the RREQ transmission is triggered at a simulation time of 10 ms. Consequently, a time offset of 10 ms, denoted by $\Delta t$, must be considered, since all nodes are initialized and the network topology is configured prior to this event. Figure 44 shows the runtime of the route establishment process from UAV 4 to various UUVs. The shortest route establishment time is observed for the destination node UUV 15, at 1407 ms. The longest runtime occurs for the connection from the UAV 4 to UUV 3, at 246 ms longer. It should be noted that the runtime of the RREQ and RREP phases is influenced by two key factors in the physical model. When it comes to overwater communication, only the transmission duration and the processing time is taken into account. However, for underwater communication, both the transmission duration, the processing time and the propagation delay are considered.

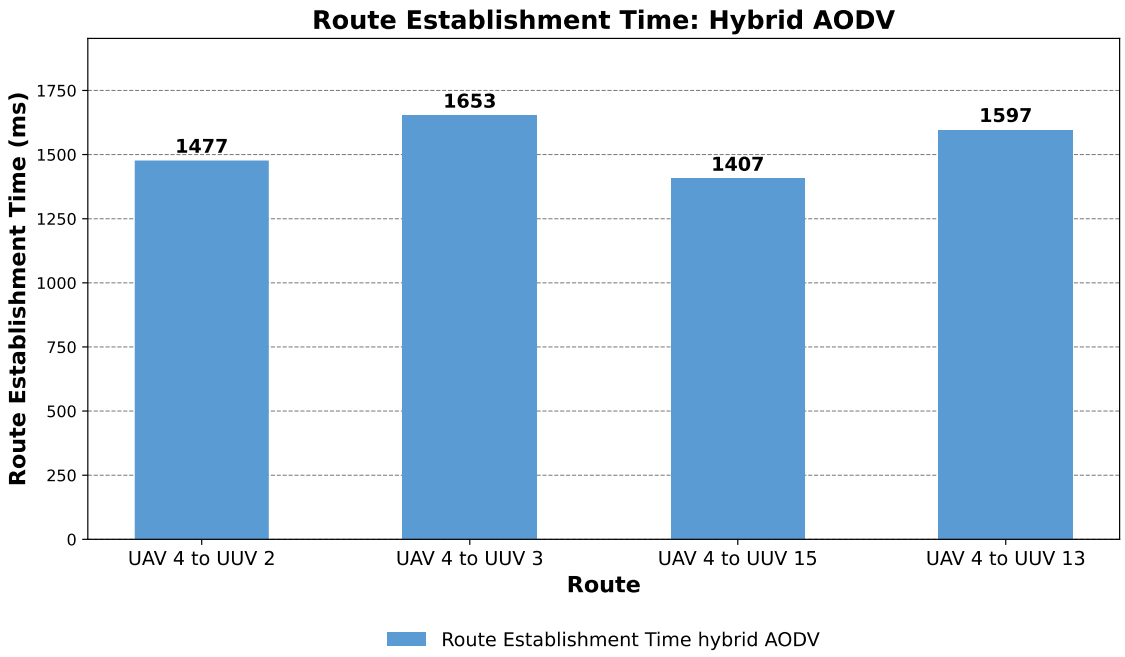


Figure 44: Route establishment time of the reference protocol hybrid AODV from the UAV to the UUVs 2, 3, 15, 13. Here, USVs are the central gateways between above-water and underwater communication.

As outlined in Section 2.1, transmission duration depends on packet size and bitrate, while propagation delay underwater is determined by distance, speed of sound in water and temperature. The parameters applied in these calculations are listed in Table 22. For a packet size of 48 bytes, a bit rate of 968 bps, a temperature of 20 °C, and a distance of 500 m, the transmission duration is approximately 516 ms, while the propagation delay is around 330 ms. This means that propagation accounts for roughly 39.00% of the total one-way transmission time, highlighting its dominant influence on underwater communication latency.

As Figure 45 shows, the Route Establishment Time remains the same for all four routing scenarios in the extended AODVs. For the route from UAV 4 to UUV 3, the hybrid AODV requires 1653 ms to establish a route. In comparison, the extended AODV with external routing logic reduces the runtime to 869 ms (−47.43% compared to hybrid AODV), while the internal routing logic achieves only 79 ms (−95.22% compared to hybrid AODV). The internal logic's particularly short establishment time of 79 ms is explained by the fact that all traffic remains within the overwater communication domain. As shown in Figure 12, the routing logic ensures that a gateway node forwards messages into the underwater domain only when the Relay flag has been explicitly set by the routing protocol.

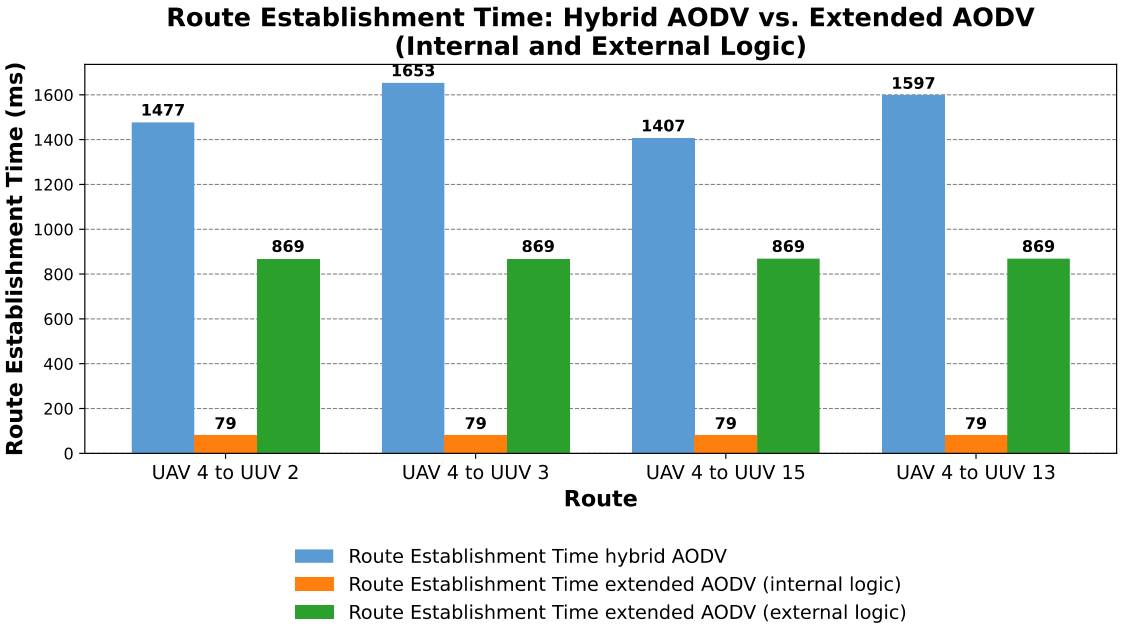


Figure 45: Route establishment time of the reference protocol hybrid AODV, the extended AODV (internal logic and external logic) from the UAV to the UUVs 3, 13, 15, and 2.

In the overwater domain, the RREQ is first transmitted. After 30 ms, RelReq messages are sent, to which all USVs with gateway functionality respond by issuing a RelRep and forwarding the RelReq further. Transmitting the RelReq message initiates a new timer, also set to 30 ms. Once the timer has expired, the Relay with the shortest geographical distance from all the available candidates is chosen. At this point, the route is fully established.

In the case of external routing logic, it takes 869 ms to establish the route. This delay occurs because the Relay Echo messages from the UUVs must first arrive at the gateway nodes. Consequently, the overwater routing process of the external logic at UAV 4 does not commence until 800 ms have elapsed. These 800 ms are required to ensure that a sufficient number of Relay Echo entries are available at the USVs, where the SNR of the received messages is determined and stored in the USV destination table (see Figure 16). However, this timer can be adapted depending on the mission requirements. It is not mandatory for all available UUVs to be registered in the USV destination table. Nevertheless, there must be a sufficient number of entries to allow the selection of the Relay with the best SNR score from the available options. Once the 800 ms have elapsed, the routing logic proceeds in the same manner as for the internal case. The only difference is that the RelRep also carries the SNR value. As the UUVs continue to transmit Relay Echo messages periodically after the initial transmission, the 800 ms waiting period is only required

for the initial route establishment, not for subsequent routes. Therefore, the net duration of route establishment is just 69 ms, consistent with the 79 ms observed for internal logic. This is notable since the internal logic routing process does not begin until the 10 ms initialization period has elapsed.

### 5.2.2 Extended AODV from UUV to UUV: Route Establishment Time

The Route Establishment Time from a UUV to additional UUVs and a UAV is presented in Figure 46 below. In this scenario, UUV 6 establishes routes to four additional UUVs and one UAV. For comparison, the hybrid AODV is employed alongside the extended AODV. Contrary to the initial assumption, the results indicate that the extended AODV does not always achieve a shorter Route Establishment Time. For the route from UUV 6 to UUV 15, however, the hybrid AODV yields a shorter establishment time, being 68 ms faster than the extended AODV (+2.60 % with respect to hybrid AODV). This is because the hybrid AODV transmits the RREQ and RREP directly underwater via a single Hop, resulting in a high signal propagation delay. By contrast, the extended AODV attempts to minimize the underwater path by leveraging the higher transmission speed of the overwater radio channel. It first identifies the optimal gateway and then completes the route with a single underwater Hop to the destination. However, the extended AODV logic introduces significant delays in timers. For instance, the sender waits 500 ms for a potential RREP from a nearby UUV. If such a reply does not arrive, it initiates a Ping towards the nearest USVs to start the RREQ via the overwater domain. This behavior is particularly evident in the route establishment towards UAV 4, where the hybrid AODV is 906 ms faster. The reason for this is that the hybrid AODV needs only one underwater Hop to a USV, sets up the overwater route in 10 ms, and gets the RREP via the same USV back to UUV 6. For the route from UUV 6 to UUV 13, the time taken to establish the route is identical for both approaches.

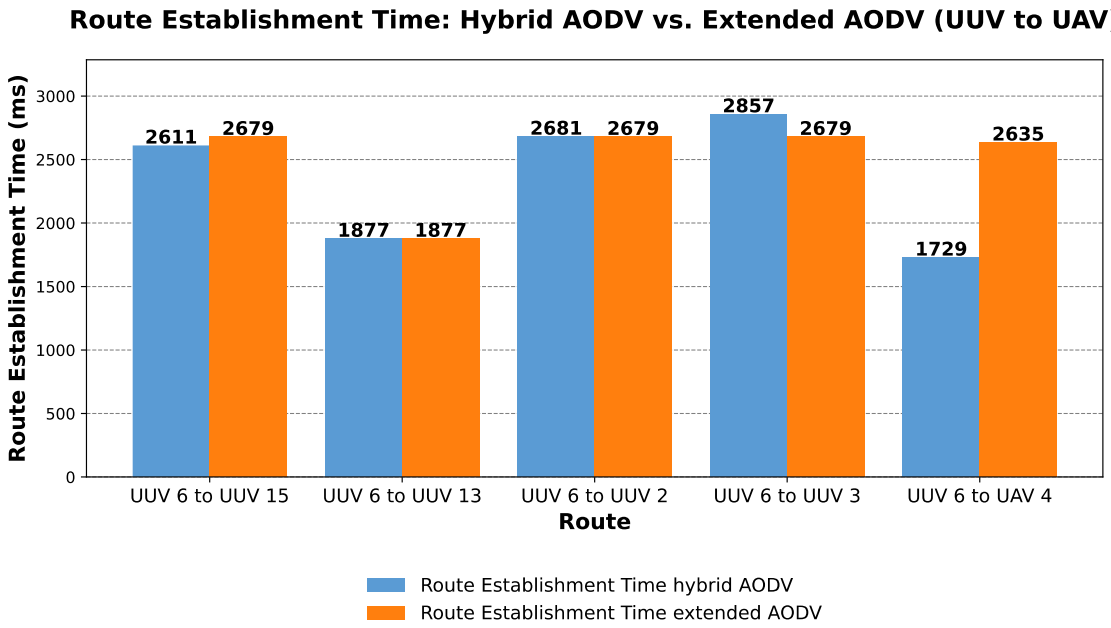


Figure 46: Route Establishment Time of UUV 6 to the UUVs (15, 13, 2, and 3) and UAV 4.

This can be attributed to the fact that UUV 13 is located very close to UUV 6 in the underwater domain. As a result, both the hybrid and the extended AODV approaches exhibit identical routing behavior when the RREP arrives at the transmitting node UUV 6. However, this only works if the destination node is one Hop away since the TTL is limited to one Hop, which is precisely the case in this scenario. For the routes from the UUV 6 to UUV 2 and UUV 3, the results show that the extended AODV achieves slightly lower establishment times. For UUV 2, the difference is negligible with a reduction of only 2 ms (−0.07 % with respect to hybrid AODV). In contrast, for the route to UUV 3, the extended AODV reduces the establishment time by 178 ms (−6.23 % with respect to hybrid AODV).

### 5.2.3 Hybrid and extended AODV from UAV to UUV: Received Bytes

Figure 47 illustrates the number of bytes received for the selected routing scenario from the UAV 4 to several UUVs. This metric serves as an indicator of overall data traffic. Each node that receives a packet records the number of bytes in the simstat function, and this counting procedure is implemented. At the end of the simulation, the simstat function outputs the total number of bytes received across all nodes. It should be noted that the total number of received bytes is higher than the initial transmission. This is because a single transmission, such as an RREQ, can be received by multiple nodes simultaneously

(e.g. five nodes), thereby increasing the cumulative count of received bytes. The received bytes metric is chosen because it reflects not only the traffic generated by a single transmission, but also the local node density. Although a packet is transmitted only once by the sender, it can be received by multiple nodes within range. Therefore, the sum of received bytes provides insight into how many nodes are within communication range and actively participating in data exchange. Thus, received bytes are a more suitable indicator for evaluating the overall network load and effective communication reach than considering only the number of sent bytes. The simulation distinguishes between the radio and hydroacoustic domains to quantify the amount of data traffic generated in each medium. This differentiation enables us to evaluate whether the traffic in the hydroacoustic channel can be significantly reduced, as required by NFR3, thereby reducing the communication load in the underwater domain.

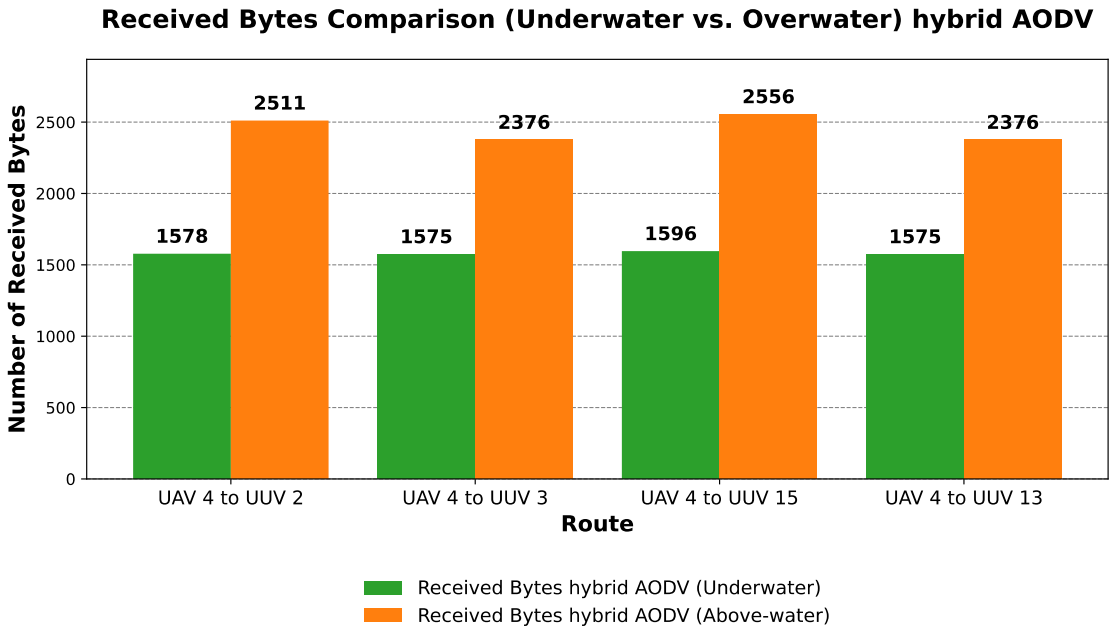


Figure 47: Comparison of the received bytes during route establishment of the reference protocol hybrid AODV. A distinction is made between the radio and hydroacoustic domains.

For reference, Figure 47 is presenting the distribution of received bytes in the radio and hydroacoustic channels for the hybrid AODV. As messages in the hybrid AODV traverse both domains, it is reasonable to expect the number of bytes received to be similar in both domains. Since the hydroacoustic domain consists of only five nodes, whereas the over-water domain contains significantly more, the radio channel receives substantially more traffic than the underwater channel. Across the evaluated links, the number of received bytes in the radio domain exceeds that of the underwater domain by between 50.86 % and

60.15 % (with respect to the underwater hybrid AODV). Furthermore, it is observed that the number of received bytes remains relatively constant across the different routes in both the overwater and underwater channels.

The subsequent analysis focuses exclusively on the hydroacoustic channel, as the overarching objective of this work is to reduce underwater traffic, which is the critical medium. Figure 48 compares the number of received bytes for the hybrid AODV and the extended AODV, taking into account both internal and external routing logic. A notable observation is that the number of received bytes in the underwater channel remains zero across all routing processes for the internal routing logic. This is because it is not necessary to transmit any data into the underwater domain for route establishment. This is deliberately avoided by setting specific flags within the control messages. It is only once the route has been successfully established and the data packet forwarded to the selected USV that the transition from radio to hydroacoustic domain occurs.

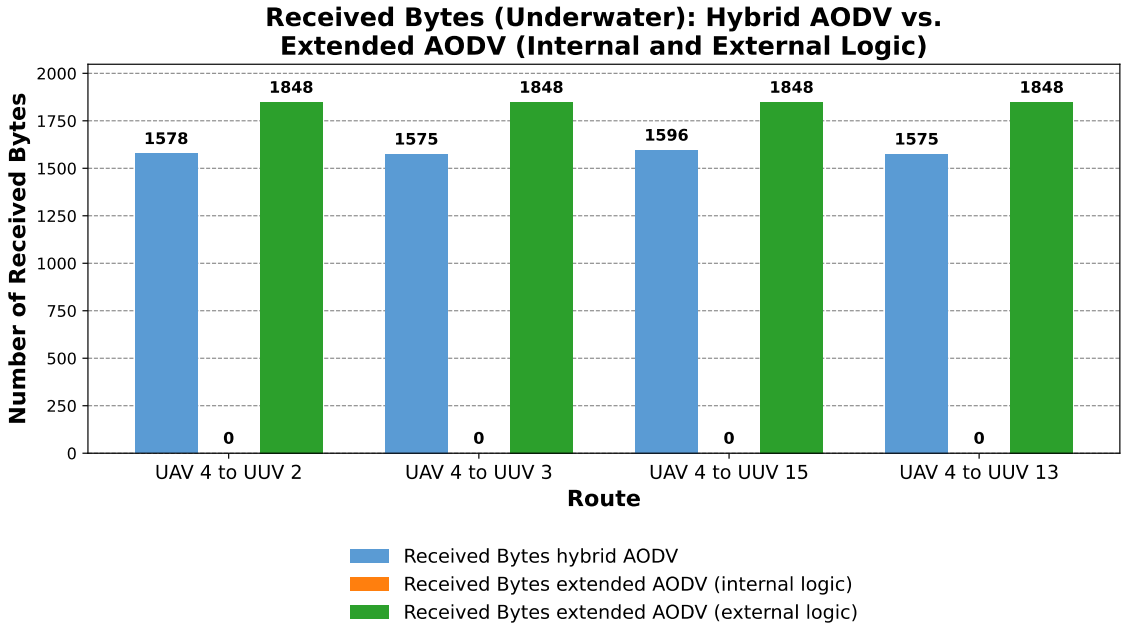


Figure 48: Comparison of the bytes received during route establishment of the reference protocol hybrid AODV and the extended AODV of the internal and external logic. It can be seen immediately that no data traffic is generated in the hydroacoustic channel with the internal logic of the extended AODV.

When the extended AODV with external routing logic is considered, the number of received bytes increases compared to the hybrid AODV. Across the evaluated links, the growths ranges from 15.79 % to 17.33 %, with a maximum increase of 17.33 % (with respect to the underwater hybrid AODV). This behavior is caused by the UUVs, which

periodically transmit Relay Echo messages to the USVs. These messages can be transmitted at intervals that are either approximated using a Poisson distribution (see Section 3.3.4) or selected randomly within a specified range, as is done in this simulation. In this case, a random offset of between –30 ms and +50 ms is used. Increasing the interval reduces the overall data traffic. However, this results in a decreased update frequency of the SNR values stored at the USVs when they receive the Relay Echo messages, which are then returned to the sender within the RelRep. It should be noted that the number of received bytes in the extended AODV with external logic is higher only because, within the simulation, all UUVs periodically transmit Relay Echo messages during the defined time window. In practice, however, Relay Echo messages would be used more adaptively. For example, a UUV would only transmit such messages when it detects that it is no longer following its mission plan.

### 5.2.4 Hybrid and extended AODV from UUV to UUV/UAV: Received Bytes

Figure 49 demonstrates that, although the extended AODV does not achieve shorter Route Establishment Times, it significantly reduces the overall data traffic in terms of received bytes when examining the received bytes from UUV 6 to the four different UUVs and the UAV 4. For the routes from UUV 6 to UUV 15, UUV 2, UUV 3, and UAV 4, the number of received bytes of the extended AODV remains constant at 662 bytes. Compared to the hybrid AODV, this corresponds to reductions between −68.11 % and −69.74% (with respect to the underwater hybrid AODV). The only exception is the route to UUV 13, where the extended AODV reaches 1067 bytes, resulting in a smaller reduction of −40.62 %.

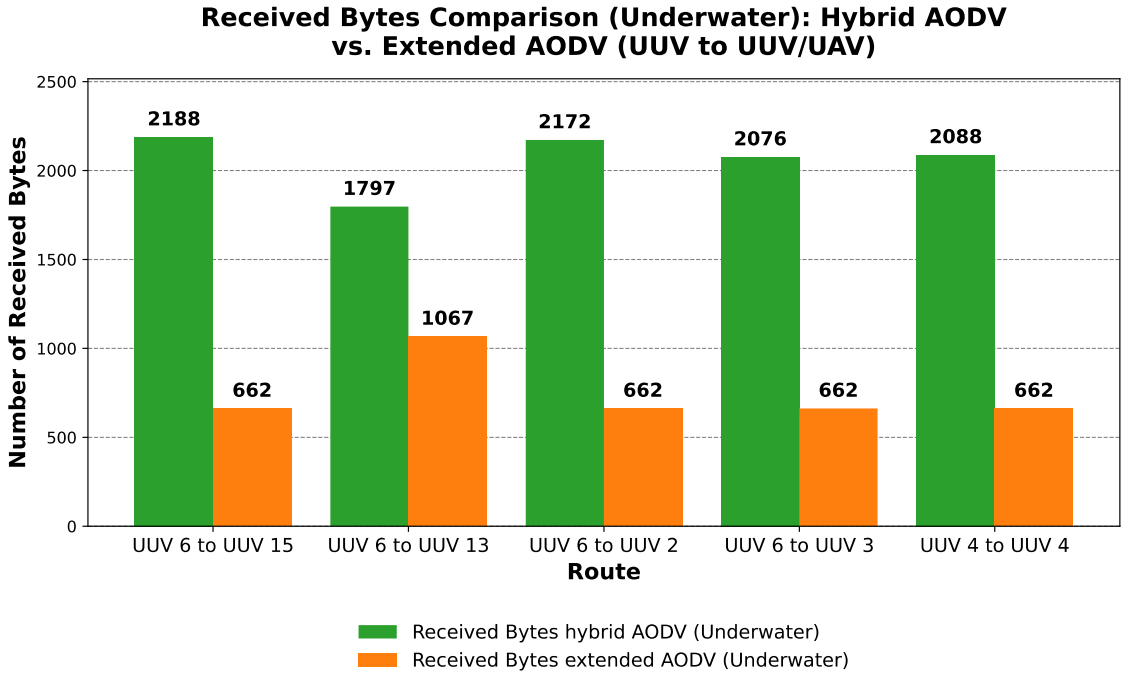


Figure 49: Comparison of the received bytes during route establishment of the hybrid AODV as a reference protocol and the extended AODV from UUV 6 to UUVs 3, 13, 15, and 2 and to UAV 4. The extended AODV shows significantly lower data volume.

This is because UUV 13 is located directly adjacent to UUV 6, meaning it can be reached directly with a RREQ, to which it subsequently responds with an RREP. Nevertheless, data traffic is still observed in the radio channel. This is because the RREP is not returned within the 500 ms interval, triggering a fallback to route establishment via the overwater domain. For the hybrid AODV, the destination node UUV 15 receives the highest number of bytes, indicating that the largest number of RREQ transmissions is required to successfully establish a route to this node.

### 5.2.5 Hybrid and extended AODV from UAV to UUV: RDR

Traditional loss metrics relate dropped packets to sent packets. However, such sender-centric measures do not reflect broadcast-based reactive routing, such as AODV, where a single transmission may be received by many neighbors, each of which makes an independent forward-or-discard decision. To capture protocol efficiency after reception in the hydroacoustic channel, a network-wide Reception Drop Rate is used. The counting process is executed at the node following the arrival of the network layer AODV and UWAODV. The value *rcvd_i(T)* denotes the number of packets successfully received by node *i* within the observation window *T*, while *drop_i(T)* denotes the number discarded

after reception. A discard is determined by a deterministic chain of checks. These include IP-level validity, such as TTL and self-origin rejection, and duplicate suppression via the message cache. Other checks include AODV/UWAODV routing logic, such as the presence of reverse or forward routes, and extended hydroacoustic-domain policies, such as USV-only participation in RelReqs handling and USV processing of Relay Echo from UUVs. There is also a one-hop limit for underwater requests. Losses prior to reception at the physical or MAC layer are excluded. Over the node set *N*, the metric is defined in following Formula 5.1:

$$RDR_{acoustic}(T) = \frac{\sum_{i=1}^{N} drop_i}{\sum_{i=1}^{N} rcvd_i}, \quad 0 \leq RDR_{acoustic}(T) \leq 1. \tag{5.1}$$

In this work, the RDR is mathematically defined as a normalized value between 0 and 1. This representation is used consistently in all plots to ensure comparability across simulation runs and communication domains. For interpretability, however, the corresponding percentage values are provided in the text (e.g., an RDR of 0.72 corresponds to 72%). This dual representation allows the RDR to remain mathematically consistent while offering an intuitive interpretation in the discussion of the results.

### 5.2.6 Hybrid and extended AODV from UAV to UUV : Comparison of RDR

Figure 50 depicts the drop rate for packets received by the hybrid AODV reference protocol. The route setup from sender UAV 4 to UUVs 15, 13, 2 and 3 remains the same. The RDR distinguishes between radio and hydroacoustic communication, showing that they are nearly identical. This is because the routing behavior of the hybrid AODV protocol does not change between the above-water and underwater domains. RREQ and RREP messages are transmitted across domains via USVs acting as gateways. Despite the smaller number of nodes in the underwater domain, the acoustic channel exhibits a higher Reception Drop Rate. While the overwater domain achieves RDR values of around 72% and 70%, the underwater domain remains consistently at approximately 79%, indicating greater redundancy in the received broadcast traffic.

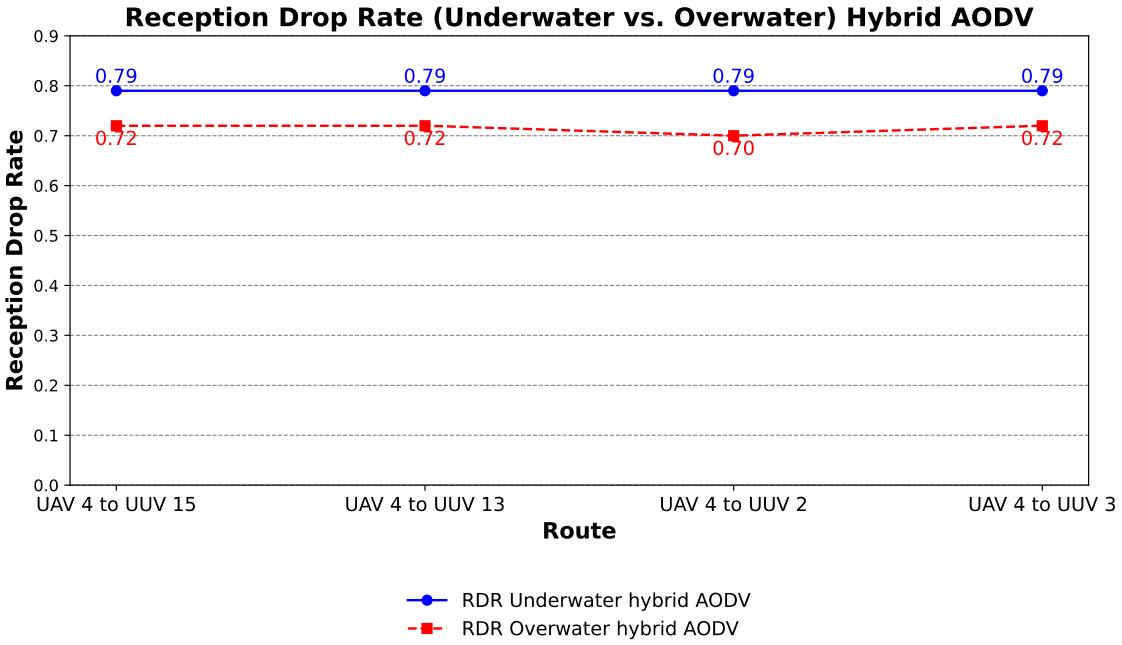


Figure 50: Reception Drop Rate of the reference protocol Hybrid AODV. The metric quantifies the proportion of received broadcast packets that are discarded at the nodes. The results are shown for both the overwater and the hydroacoustic domain.

Most drops occur during the RREQ dissemination phase, as each RREQ is broadcast and duplicate instances with the same RREQ ID are systematically discarded by receiving nodes. Additional drops occur for RREPs if a node receives the reply but is not part of the corresponding route. Finally, generic protocol checks such as TTL expiration or the absence of a valid forward/reverse route also lead to discards. These mechanisms explain why most dropped messages result from control packet redundancy rather than data packet loss.

Figure 51 compares the drop rate for reception of the hybrid AODV (used as a reference) with that of the extended AODV, under both internal and external routing logic. For the internal routing logic of the extended AODV, the RDR remains constant at zero. This is because no data traffic occurs within the internal routing logic, as the route is established entirely within the above-water domain. However, if data transmission were to occur, the RDR in the underwater channel would be expected to exhibit at least a minimal value. Considering the external routing logic of the extended AODV, it becomes evident that the RDR remains constant at 14%. This is because the only traffic within the hydroacoustic channel arises from UUVs periodically transmitting their Relay Echo messages to the USVs. However, other UUV participants also unavoidably receive these Echo messages and subsequently discard them.

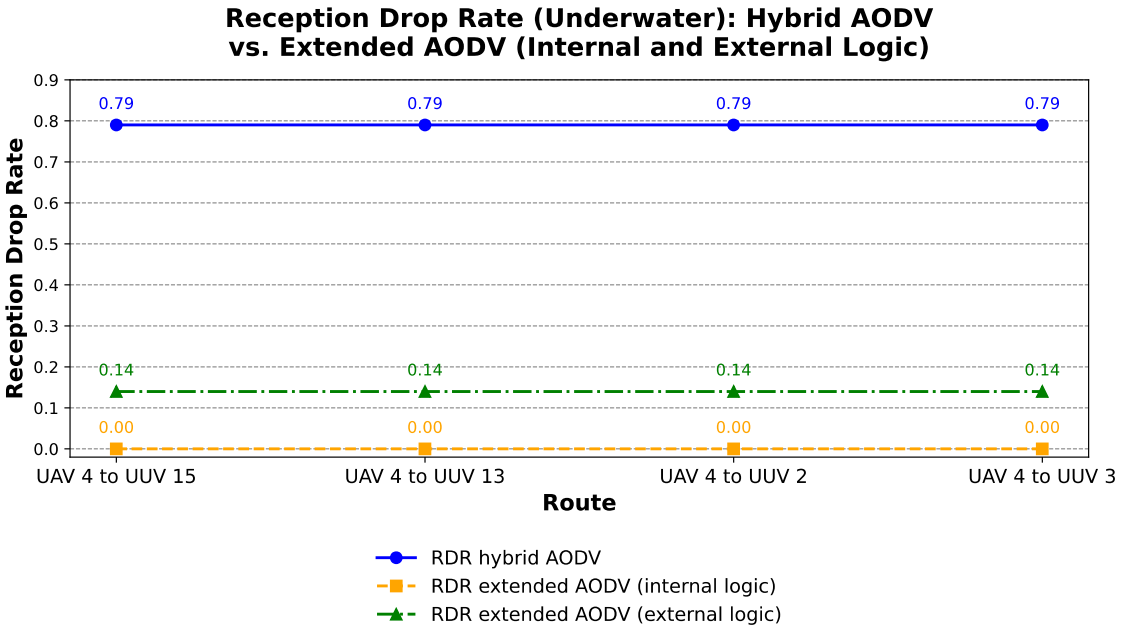


Figure 51: Reception Drop Rate of the reference protocol hybrid AODV, as well as the extended AODV of the internal and external routing logic. Since there is no data traffic in the hydroacoustic domain, the drop rate is zero.

### 5.2.7 Hybrid and extended AODV from UUV to UUV/UAV: Reception Drop Rate

Figure 52 depicts the Reception Drop Rate from UUV 6 to UUVs 13, 15, 2 and 3, as well as UAV 4, when using the extended AODV compared to the hybrid AODV reference. The hybrid AODV exhibits Reception Drop Rates between approximately 79% and 86%, whereas the extended AODV achieves RDR values in the range of 69% to 76%. Hence, the extended AODV consistently drops fewer packets than the hybrid protocol on the considered routes. For the route from UUV 6 to UUV 3, the RDR values of both protocols are almost identical (about 79% for the hybrid AODV and 76% for the extended AODV). This is because UUV 13, which serves as a Relay, is located only one hop away, so that the routing behavior of both protocols converges when a single hop is involved. The still comparatively high RDR of around 69%in the extended AODV on the remaining routes is caused by the large number of control messages that have to be transmitted over the hydroacoustic channel. These include the initial underwater RREQ to verify whether the target node is within range, subsequent Ping messages from the USVs to the sender UUV 6, return Pings from UUV 6 to the other participants and, finally, the underwater RREP transmitted from the USV to the UUV once the route has been established above-water.

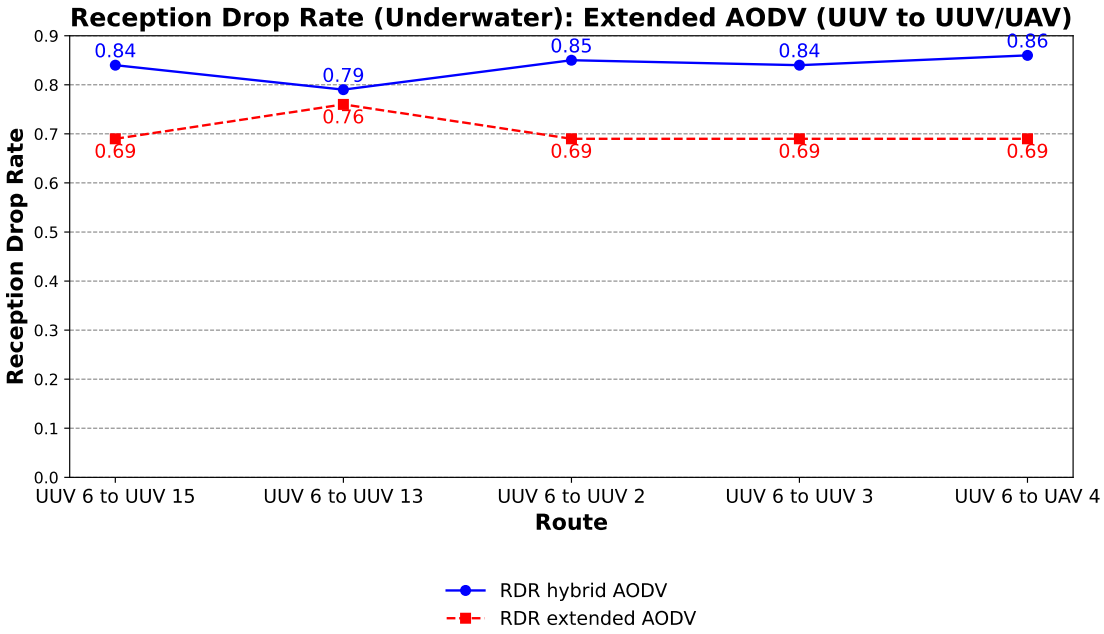


Figure 52: Reception Drop Rate of the reference protocol hybrid AODV, as well as the extended AODV from UUV 6 to UUVs 3, 13, 15, and 2, as well as UAV 4. The drop rate of the extended AODV is lower than that of the hybrid AODV.

Given the high volume of control signalling, multiple UUVs and USVs within communication range inevitably receive these messages, even though they are not part of the intended route. Upon recognizing this, they correctly discard the packets.

### 5.2.8 Hybrid and extended AODV from UAV to UUV: SNR

Figure 53 illustrates the SNR values for propagation distances ranging from −1,000 m to −100 m. While these values do not represent depth, they provide a technical overview of how the SNR behaves when the modem sensitivity is set to its maximum for large communication ranges. Although the actual water depth at Borkum Riffgrund 2 is only about 30 m, the UUVs are deployed horizontally, so that communication distances of up to 1,000 m between the nodes may still occur. The results show that the SNR remains around 28 dB re 1 µPa at −1,000 m and increases to approximately 48 dB re 1 µPa at −100 m. These values were obtained under the assumption that the EvoLogics modem operates at a source level of 166 dB re 1 µPa, which represents the long-range transmission mode.

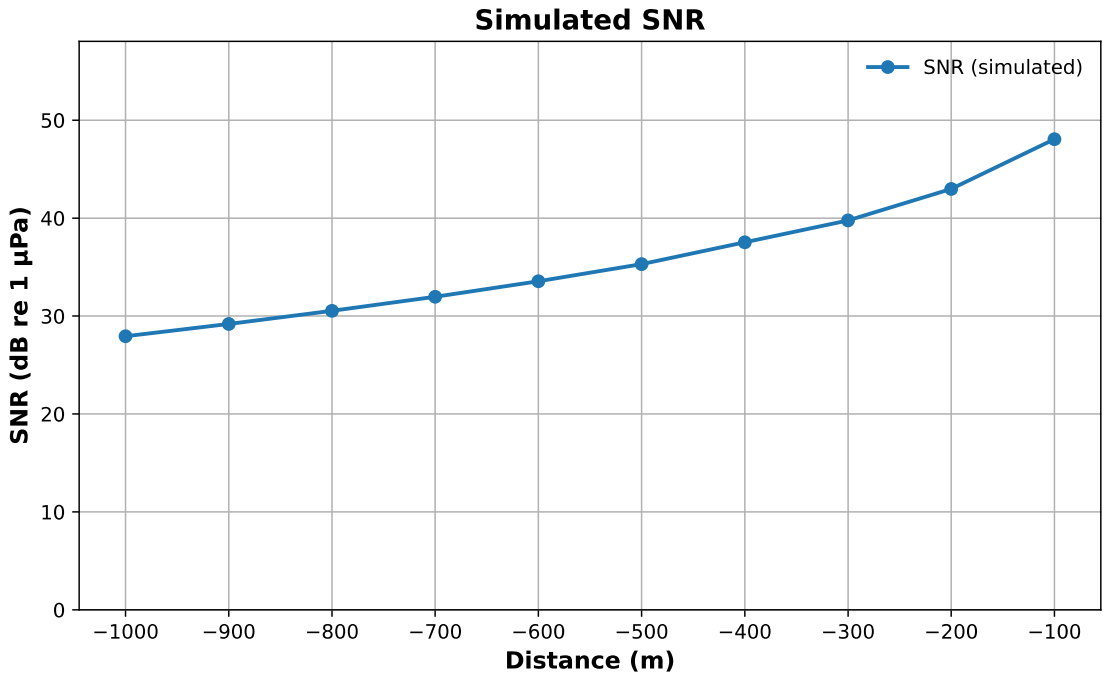


Figure 53: Representation of the SNR curve from -1,000m to -100m. It can be seen that the SNR values are significantly higher at shallow depths.

Furthermore, the simulation takes into account frequency-dependent attenuation as well as ambient noise. In practice, however, additional factors that cannot be captured in the simulation or during limited field tests may influence the SNR.

### 5.2.9 Extended AODV from UAV to UUV: Error Management

In the present simulation, the number of transmitted packets is only included in the simulation of the error management. This is because the size of data packets may vary, which would make the simulation results incomparable across different scenarios. In contrast, the route establishment procedure is independent of data packet size and therefore provides a consistent basis for evaluating routing performance. Once a route has been successfully established, the source node can transmit data packets to the destination node. A new route discovery process is triggered in one of two ways. Firstly, a route must be rebuilt when the routing table entry expires. The validity period needs to be adapted for each application scenario. In highly dynamic environments, routing table lifetimes should be kept short to ensure timely updates. Secondly, a new route discovery process is initiated if a routing error occurs during data transmission because an intermediate node is no

longer part of the route. In both cases, the protocol re-enters the route discovery phase to ensure communication can be restored before subsequent data transmissions proceed.

Figure 54 illustrates the impact of error management on overall network performance. It is important to note that the simulation covers both route establishment and the subsequent transmission of data packets. Error management can only be properly validated under these conditions. The simulation is set to run for 40 seconds, compared to a maximum of three seconds for route establishment simulations. This longer time frame is necessary in order to analyze the network dynamics under changing topologies.

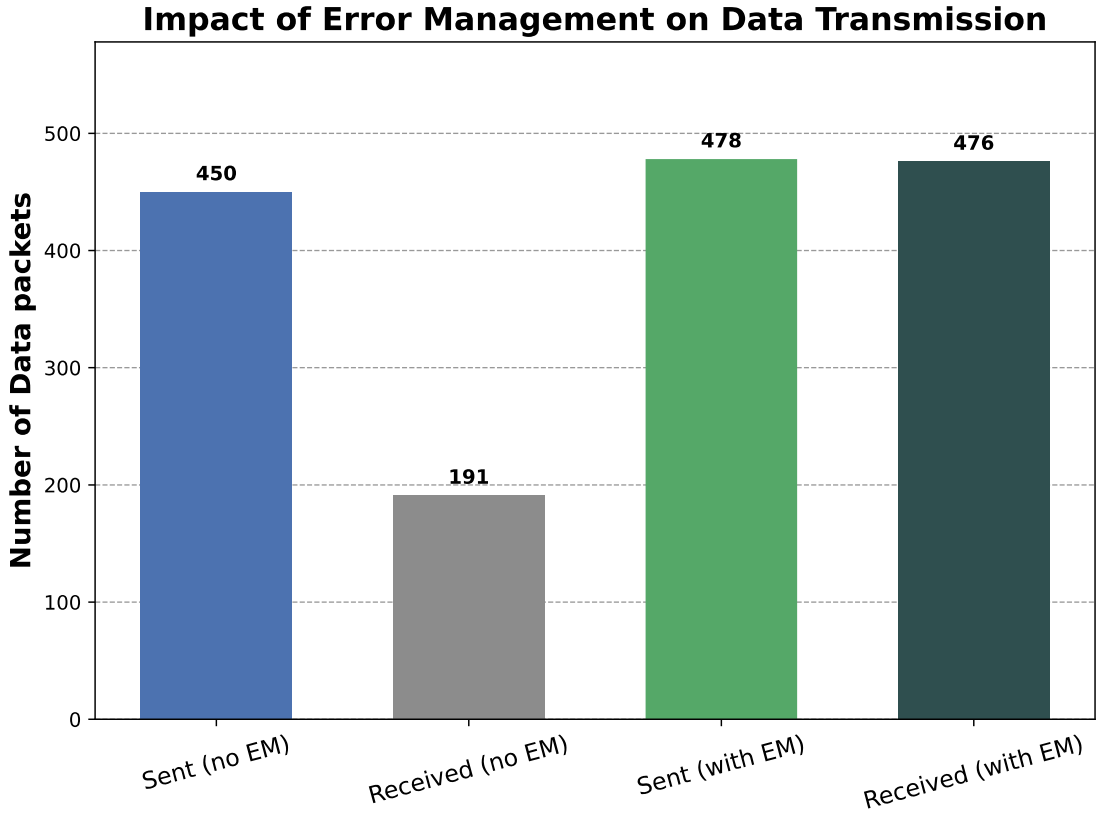


Figure 54: Number of data packets sent in direct comparison with and without error management. The duration of the simulation is 40 seconds, which is significantly longer than the time considered for route establishment.

The simulation configuration is identical to previous runs in terms of physical models, node size, height and depth. A total of 20 nodes are deployed. As in previous scenarios, UAV 4 acts as the sender and establishes routes to nodes 2, 3, 6, 13, 5, 10, 11, 16, 8, 12, 7, 14 and 20. This ensures that both above-water and underwater destinations are included. Once the routes have been established, UAV 4 begins transmitting data periodically to the destination nodes. However, during the simulation, the topology changes: The

nodes 4, 3, 5, 11, 18, 7, 12, 10, 4, 15 and 1 change position and nodes 12 and 5 fail completely. This means that certain next-hop nodes are no longer reachable during data forwarding. This results in an error along the route, which is recognized and resolved with error management. Nevertheless, if the node that fails is the destination node, it cannot be reached even with error management.

As shown in Figure 54, without error management, 450 data packets are transmitted in total, of which only 191 reach their destinations (42.44 %). With error management enabled, 478 packets are transmitted and 476 arrive successfully (99.58 %). This demonstrates a substantial improvement in delivery reliability when error management is applied. The loss of the remaining two packets is due to the failure of nodes 12 and 5, which makes delivery to those destinations impossible. The results of the simulation will be evaluated against the specified requirements in the following Chapter 6.

# 6 Discussion and Transferability

This chapter summarizes and discusses the key results of the simulations and field experiments that were conducted. The focus is on interpreting these results in relation to the defined objectives. Then, a comparison is made between the simulations and practice to highlight both consistencies and deviations. Additionally, it critically assesses the strengths, weaknesses, and limitations of the extended AODV protocol. Lastly, the chapter outlines potential future improvements and directions for further research.

## 6.1 Interpretation of Results in the Context of the Objectives

Interpreting the obtained results requires placing them in the context of the objectives and research questions defined in Chapter 1. The aim of this work was to develop, implement, and evaluate an enhanced Ad-Hoc routing protocol that enables Cross-Domain communication between UAVs, USVs, and UUVs. This protocol protects critical maritime infrastructure through the emergence of self-organization among UxVs. The functional and nonfunctional requirements formulated in Section 1.3 provided the conceptual foundation, and the research questions (RQ1–RQ4) guided the systematic evaluation. Keeping this in perspective, the following interpretation summarizes how well these objectives were achieved.

Regarding RQ1, the results show that hybrid AODV implementations have severe limitations in Cross-Domain hydroacoustic communication scenarios. According to the AODV RFC (Perkins et al., 2003) specification, the hybrid AODV attempts to forward Route Requests through any available gateway into the underwater domain. However, this strategy results in a large number of unnecessary RREQ transmissions that propagate acoustically until a destination node responds with a RREP. Consequently, the procedure is slowed by the long propagation delays inherent in underwater channels, rendering it highly inefficient for Cross-Domain route discovery.

Additionally, the simulations revealed that the hybrid AODV produces an excessive amount of redundant control messages. The high level of reception drops indicates that the hydroacoustic channel is unnecessarily loaded, further increasing the probability that

messages will never reach their intended destinations. Although the current simulation framework does not allow precise quantification of interference effects, such as signal overlap, it is reasonable to assume these factors would further degrade performance. Taken together, these findings demonstrate that conventional AODV is unsuitable for efficient, reliable Cross-Domain routing.

Regarding RQ2, the results show that the extended AODV can successfully support reliable Cross-Domain communication to protect maritime critical infrastructures. The hybrid AODV as the baseline protocol is selected because it is open, lightweight, and well-documented. This makes it suitable for integration into a simulation framework and as a reference for evaluation. On this foundation, an extended version of AODV is developed that explicitly recognizes USVs as gateway nodes and anticipates routing data traffic across domains.

This extension introduces two new control message types: RelReq and RelRep. These types enable nodes to query potential gateways and receive appropriate responses. Two approaches were designed and evaluated: an internal, geo-based method and an external, signal-quality-based method. The geo-based approach uses the known mission plans of UUVs to predict routing paths. This method is efficient, but it can be inaccurate if a UUV deviates from its planned trajectory. The external approach addresses this limitation by introducing periodic Relay Echo messages from UUVs. This allows USVs to estimate the SNR and use it as the basis for routing decisions. This mechanism provides a more adaptive solution that is robust against dynamic changes in the underwater environment.

The extended AODV protocol is applied to enable communication from UAVs to UUVs via USV gateways and in the reverse direction. This allows UUVs to reach other UUVs underwater and establish links back to UAVs. However, the evaluation did not include UAV to USV or UAV to UAV communication since conventional AODV can already address this scenario and it does not represent a Cross-Domain challenge. The extended AODV from UUV to UUV/UAV, also known as subsea routing, does have long runtimes because different timers have to be set in the routing. However, this significantly reduces data traffic underwater.

Additionally, a new error management concept is introduced to prevent routing failures in dynamic topologies. Rather than adopting existing error handling schemes, a new approach is designed to minimize traffic while ensuring reliability. This approach combines passive acknowledgments that exploit the isotropic radiation pattern of hydroacoustic

transmissions with explicit acknowledgment messages at USVs and UUVs. This mechanism reduces control overhead while maintaining robustness in highly dynamic maritime environments.

Overall, the extended AODV incorporates gateway awareness, domain-specific metrics, and an adapted error management strategy into its routing logic. This allows it to avoid flooding the underwater channel unnecessarily and ensures more reliable and efficient Multi-Domain Operation.

In RQ3, the extent to which a practice model initiator is suitable for parameterizing the simulation is examined. A central contribution of this work is the development of a time-discrete, Multi-Domain simulation framework in MATLAB. This framework integrates heterogeneous nodes, including UAVs, USVs and UUVs, each of which is equipped with domain-specific physical layer models for radio and hydroacoustic communication. The hybrid AODV protocol and the extended AODV proposed in this work were both implemented, including routing tables and control message handling (RREQ, RREP, and data), as well as the novel RelReq, RelRep, and Relay Echo message types. This environment provided the basis for systematically analyzing Cross-Domain routing behavior and evaluating the performance of the enhanced protocol under controlled and repeatable conditions. Therefore, the framework represents a core scientific contribution of this work as well as a supporting tool.

To parameterize the self-developed simulation framework, a physical demonstrator consisting of four Raspberry Pis, two hydroacoustic modems, and a drone acting as a Relay node is constructed. The parameterization is carried out separately for the overwater and underwater domains. Practical tests determined that a reliable communication range of approximately 200 meters per Hop could be achieved using the Raspberry Pis's built-in Wi-Fi modules for the overwater part. Additionally, the offset between the internal clocks of nodes operating in Ad-Hoc mode is measured, resulting in an average deviation of approximately 7.73 ms. The processing time per Hop is also evaluated and showed an average delay of around 4.59 ms. The processing time of the Raspberry Pis is integrated directly into the MATLAB simulation.

For the underwater domain, the physical characteristics of the hydroacoustic modems were incorporated into the simulation models. Basic parameters, such as bit rate and transmission delays, were obtained from the manufacturer’s data sheets. Thus, the simulation

framework reflects both theoretical assumptions and empirically measured system behavior from the demonstrator.

For validation purposes, the demonstrator is used to compare the results of practical experiments with those obtained in the MATLAB simulation in both the above- and underwater domains for the reduced use case of 4 nodes. In the overwater domain, route and Data Establishment Times were measured. The simulation shows that the Route Establishment Time is approximately 8.83 ms shorter than in the practical demonstrator. For the Data Establishment Time, however, the deviation is only 1.05 ms, with the practical demonstrator being slightly faster in this case. This indicates a very close alignment between the simulated and real behaviors. For the parametrization presented in Section 4.4.2, the practical measurements of the Relay Echo runtime, the SNR, and the Data Packet runtime were consistently compared against the corresponding values of the MATLAB simulation, which served as the reference throughout this section. Negative values indicate that the practical measurements are lower than the simulation results. For the Relay Echo message, the practical latency values were consistently lower than the simulated ones. Using the simulation as reference, the practical measurements showed relative deviations between −3.31% and −4.35%, with a mean deviation of −3.80%. For the SNR values, the practical measurements showed both slightly higher and slightly lower values compared to the simulation. Using the simulation as reference, the relative deviations ranged between −2.40% and +3.07%. For the data packet latency, the practical measurements were consistently lower than the simulated values. Using the simulation as reference, the relative deviations ranged from −4.01% to −1.11%, with a mean deviation of −2.99%. These results confirm that the simulation framework captured the system's general behavior and reproduced its domain-specific characteristics with a high degree of accuracy. The minor discrepancies can be attributed to environmental effects and hardware-specific variations that are difficult to model. Therefore, the combination of a self-developed simulation framework and a physical demonstrator ensured realistic parameterization and credible validation, providing a robust foundation for evaluating the enhanced protocol.

Regarding RQ4, the focus is on quantifying the performance improvements of the extended AODV compared to the reference implementation. For this purpose, the use case of protecting the Borkum Riffgrund 2 wind farm is used as a scaled simulation. The physical properties are still based on the parameterization with the practical demonstrator. In

the case of communication between UAVs and UUVs, it is shown that the decisive factor is avoiding the hydroacoustic channel during route discovery. Discovery is fully performed in the radio domain instead, and the hydroacoustic channel is only activated for the actual data transfer. This design prevents the unnecessary flooding of the underwater medium and directly addresses one of the key limitations identified earlier. As a result, in terms of route discovery time, the extended AODV with internal routing logic is up to 95.22% faster than the hybrid AODV. The version with external routing logic is 47.43% faster. These results confirm that anticipating underwater routes through a gateway avoids the excessive delays caused by attempting to discover routes directly through hydroacoustic communication.

At the same time, for the scenario of routing UAVs to UUVs, the results show that the external routing logic of the extended AODV protocol generates additional Relay Echo messages. With the chosen configuration, the extended AODV with external routing logic generates up to 17.33% more received traffic in the hydroacoustic channel compared to the hybrid AODV. However, this increase is not a structural drawback, but rather a configurable trade-off. Adjusting the UUV transmission interval proportionally reduces the additional load while preserving the robustness benefits of signal-quality-based routing.

A different picture emerges when considering the routing scenario from UUV 6 to other UUVs. In terms of Route Establishment Time, the extended AODV did not outperform the classic AODV. In some cases, it is 906 ms slower. However, the decisive advantage lies in the reduced data traffic load. In the hydroacoustic channel, the number of received bytes decreases by between 40.62 % and 69.74 % when the extended AODV with external routing logic is applied, corresponding to a maximum reduction of 69.74%. This significantly reduces channel congestion and increases the probability of successful data transmission in underwater communication.

Overall, the findings confirm that the extended AODV achieves its primary goal of minimizing redundant underwater traffic while offering flexibility in balancing robustness and efficiency depending on mission requirements.

## 6.2 Performance and Limitations of the extended AODV Protocol

While Section 6.1 focused on interpreting the results in relation to the research objectives, this section provides a structured evaluation of the extended AODV protocol. The assessment is based on the functional and non-functional requirements defined in Section 1.3, which serve as a benchmark for determining the extent to which the protocol fulfills its intended goals. For clarity, Table 23 summarizes the degree of fulfilment for each requirement, complemented by brief explanatory comments. This structured perspective highlights the strengths of the proposed approach and the areas where limitations remain.

Table 23: Overview of the evaluation of functional and non-functional requirements for the entire system.

| **Requirement** | **Status** | **Comment** |
|---|---|---|
| **FR1: Independent neighbor discovery** | Fulfilled | The system has been implemented with a fixed TTL, enabling nodes to autonomously discover neighbors without the need for centralized control. The mechanism is confirmed by both simulation and the practical demonstrator. For UUV to UUV communication, discovery is optimized: one-hop neighbors are reached directly underwater, while multi-hop discovery is redirected via USV gateways to avoid unnecessary hydroacoustic flooding. For UAV to UUV communication, the extended AODV detects all available gateways and selects the best one at the current time to send the data packet to the hydroacoustic domain. Routing is avoided underwater. |
| **FR2: Routes between UAV, USV, UUV** | Fulfilled | The extended AODV enables reliable Multi-Domain routing between UAVs, USVs, and UUVs. The routes were established exclusively in the radio Domain for communication between UAV and UAV, between UAV and USV, and between UAV and UUV. For the transition to the hydroacoustic domain, gateways are anticipated using a geographically based and signal quality-based approach in order to ultimately send the payload. Between communication from UUVs and UUVs, the extended AODV was adapted to use the hydroacoustic channel as little as possible in order to take advantage of the speed of |

| | | |
|---|---|---|
| | | the radio channel. This also enables communication from the UUV to the UAV. |
| **FR3: USVs as gateways** | Fulfilled | USVs were successfully integrated as dynamic gateways bridging the radio and hydroacoustic domains. Two complementary mechanisms were implemented: An internal, geography-based logic that anticipates UUV positions, and an external, signal-quality-based logic based on Relay Echo messages. The internal approach enabled fast route selection, while the external approach improved robustness under dynamic conditions, together minimizing unnecessary hydroacoustic traffic. |
| **FR4: Status information transfer** | Partially Fulfilled | The extended AODV provided the foundation for transmitting mission-relevant data, such as position updates and sensor readings. This is confirmed through both simulation and the demonstrator. However, the present work focused on route establishment rather than large payload transmission. In the practical demonstrator, payloads were only tested up to 20 bytes. There are still open questions regarding the stability of routes during the transfer of larger data sets, such as images or sonar data, and whether route validity can be guaranteed until such transfers are complete. |
| **FR5: Reliability ≥ AODV** | Fulfilled | The extended AODV demonstrated equivalent reliability to the reference AODV in fulfilling all defined functions. It demonstrated clear advantages in several areas, including significantly faster route discovery (up to 95,22%), reduced redundant control traffic, and a decrease in underwater channel load of up to 69.74% in certain scenarios. However, certain cases, such as UUV-to-UUV route establishment, demonstrated no improvement in discovery time compared to hybrid AODV. It is important to note that the protocol is designed to remain interoperable with existing AODV implementations and other routing protocols. This is achieved by maintaining IP-header compatibility, ensuring smooth integration into heterogeneous networks. |
| **NFR1: Reduce redundant control messages** | Fulfilled | The extended AODV demonstrated a significant reduction in redundant control messages compared to the hybrid AODV. In the UAV-UUV scenario, the hybrid AODV generated 79.00% redundant |

| | | |
|---|---|---|
| | | messages, while the extended version reduced this to 0% with internal routing logic and 14.00% with external routing logic. In subsea communication between UUV to UUV and UUV to UAV, redundancy decreased from 85% with hybrid AODV to between 69.00% and 76.00% with the extended AODV. These reductions show that shifting detection to the radio domain and refining gateway selection effectively mitigate one of the major inefficiencies of classical approaches. |
| **NFR2: ≥ 30% faster route discovery** | Partially Fulfilled | In the UAV-UUV case, the requirement is clearly exceeded: the extended AODV is up to 95.22% faster with the internal logic and 47.43% faster with the external logic compared to the hybrid AODV. In contrast, the extended protocol did not improve discovery time for UUV-UUV/UAV routing and is in some instances even 906 ms slower than hybrid AODV. Therefore, the requirement of at least 30% faster route discovery is fulfilled in Cross-Domain UAV to UUV communication, but not in purely underwater multi-hop scenarios. |
| **NFR3: Minimize underwater load** | Fulfilled | The extended AODV approach resulted in a significant reduction in hydroacoustic channel load in key scenarios. The extended AODV of the internal routing logic does not require any network traffic in the hydroacoustic domain. This hydroacoustic domain can be used purely for subsequent payload data traffic. For UUV to UUV routing, the number of received bytes in the underwater channel decreased by up to 69.74%, directly reducing congestion and increasing delivery probability. In the UAV-UUV case, shifting discovery to the radio domain reduced unnecessary hydroacoustic flooding. However, the external logic introduced periodic Relay Echo messages, which resulted in 17.33% more hydroacoustic traffic under the chosen configuration. This is a tunable trade-off via the UUV transmission interval. |
| **NFR4: Decentralized, topology changes tolerated** | Fulfilled | The extended AODV operated in a fully decentralized manner and adapted to dynamic topology changes. In the scaled simulation (up to 40 seconds with targeted topology modifications), the protocol maintained stable routes and successfully reacted to changes in node availability. The error-management concept, which combines passive acknowledgements |

| | | |
|---|---|---|
| | | with explicit ACKs, has further improved robustness. However, further studies with larger node counts, different water depths, and hardware-specific physical models are required to fully confirm scalability under diverse mission conditions. |

Based on the evaluation against the defined functional and non-functional requirements, the extended AODV meets the majority of its design objectives. The protocol reliably enabled Cross-Domain routing between UAVs, USVs and UUVs, addressing a core challenge identified at the outset. Integrating USVs as gateways has proven to be an effective solution: Route discovery remained in the radio domain wherever possible, thereby preventing excessive flooding of the underwater medium. In the end, only the data packet is sent once the route is established. The traffic of the data packet was not included in the simulations because this would not allow for a comparison of the approaches. Therefore, the focus was placed on the traffic and the time it takes to establish the route. A strength of the extended AODV is the significant reduction in redundant control messages, which directly improves channel efficiency and frees hydroacoustic capacity for payload.

However, it should be noted that several aspects are only partially fulfilled. Payload transfer is only validated for small message sizes (48 bytes in the demonstrator), leaving open questions for larger data objects such as imagery or sonar bursts. Substantial gains in route discovery were achieved for UAV to UUV Cross-Domain routes, but the Route Establishment Time did not consistently improve for purely underwater and underwater-to-air uplink scenarios (UUV to UUV and UUV to UAV). In fact, it could be up to 906 ms slower than hybrid AODV. Nevertheless, the extended protocol significantly reduced the hydroacoustic channel load by up to 69.74%, thereby lowering contention and increasing the likelihood of successful data delivery under realistic underwater conditions. Furthermore, the scaled simulation with targeted changes demonstrated robustness under dynamic topologies. Nevertheless, full confirmation across larger node counts, varying depths, and hardware-specific physical models is still to be determined.

Overall, these findings emphasize the strengths of the proposed protocol, including gateway-aware Cross-Domain routing, minimized underwater traffic, and reduced redundancy. They also clarify the limits of the current evaluation and prompt discussion of limitations and future improvements.

In addition to the requirement-based evaluation, it is important to acknowledge several limitations of the extended AODV. While these aspects were not the primary focus of this work, they could influence the applicability and scalability of the protocol in real-world deployments. These results indicate significant areas for future investigation:

- MAC-layer constraints: In the demonstrator, the Raspberry Pis relied on the existing standard Wi-Fi MAC layer, while the *Evologics* hydroacoustic modems used proprietary MAC implementations. It should be noted that these were not specifically designed or optimized for Multi-Domain Ad-Hoc routing. In particular, there are still unresolved questions regarding collision handling and channel access efficiency in the underwater domain.

- Payload scalability: The experiments and simulations primarily focused on route establishment and small payloads. It is not yet clear how the protocol performs with larger mission-relevant data objects, such as imagery or sonar bursts, especially regarding route validity over the full transfer duration.

- Energy consumption: Power efficiency is not systematically evaluated. However, energy use is a critical factor for battery-limited UUVs and UAVs, particularly if additional signaling (e.g. Relay Echo messages) increases channel activity.

- Security aspects: It should be noted that the protocol is not evaluated under adversarial conditions. Standard security mechanisms of Wi-Fi and the *Evologics* modem stack were used without modification. Future work should assess resilience against spoofing, flooding, or jamming attacks, which are relevant in the context of protecting critical maritime infrastructure.

## 6.3 Potential for Future Improvements

While the extended AODV protocol has demonstrated significant advantages over conventional routing in Multi-Domain maritime networks, several avenues remain for further improvement. These opportunities concern both the optimization of the routing logic itself and the integration of additional mechanisms to increase robustness, scalability, and efficiency. The following points outline selected directions that could guide future research and development.

### 6.3.1 Refinements to Relay Request and Relay Reply Messages

Future improvements could focus on enhancing the efficiency and clarity of RelReq and RelRep messages. One potential improvement would be to restrict the transmission of RelReq messages to geographical areas where gateways are anticipated. This spatial pruning avoids unnecessary message flooding and effectively increases the usable range of each transmission. Secondly, the scope of RelRep messages could be expanded to encompass additional metrics beyond the binary gateway availability indicator. Parameters such as SNR, latency estimates, or battery level would enable the source node to select the most suitable gateway according to mission priorities. For example, low latency could be prioritized in critical missions, or energy efficiency in long-term deployments. Thirdly, the implementation of explicit negative replies e.g. Relay Negative Acknowledgement (RelNack) would prevent the unnecessary use of unsuitable gateways. Such replies could indicate reasons such as insufficient SNR or resource overload, enabling the source to redirect the route discovery process more quickly.

### 6.3.2 Adaptive Relay Echo Messages

In the current implementation, Relay Echo messages are transmitted at fixed intervals, independent of network state or mission dynamics. While this guarantees a continuous update of link quality information, it also generates unnecessary hydroacoustic traffic during stable phases of operation. One potential area for improvement would be to introduce adaptive Relay Echoes, where the transmission interval is adjusted dynamically. For instance, if a UUV detects that it is deviating from its planned trajectory or that the SNR value toward a gateway is degrading, the echo frequency could be temporarily increased to provide more accurate information to the network. Conversely, when link conditions remain stable and the UUV follows its expected mission path, the interval could be reduced, thereby lowering the overall hydroacoustic load. An adaptive approach would be an effective solution, balancing robustness and efficiency. It would ensure that critical routing updates are available when needed, while conserving bandwidth in the underwater channel.

### 6.3.3 Cross-layer Integration of MAC and Routing

The extended AODV protocol has so far been evaluated on top of existing MAC implementations, such as standard IEEE 802.11 for overwater communication and proprietary MAC protocols of commercial hydroacoustic modems. While this approach is functional, it does have limitations in terms of efficiency, particularly in the underwater domain where collisions and retransmissions can be highly detrimental. Future improvements should therefore concentrate on cross-layer integration between the routing and MAC layers. By exposing link-layer information such as channel access delays, collision indicators, or queue occupancy to the routing logic, more accurate decisions about route selection and message timing can be made. Furthermore, the MAC could be enhanced with lightweight priority mechanisms to ensure that critical routing messages (e.g. RelReq/RelRep) are transmitted with higher reliability than regular data traffic.

### 6.3.4 Security and Resilience Aspects

The design of the extended AODV protocol has so far been agnostic with respect to security mechanisms, relying only on the default features of the underlying technologies (i.e. IEEE 802.11 WPA2 for the overwater domain and the proprietary mechanisms of commercial hydroacoustic modems). Maintaining this agnostic design philosophy is a promising direction for future improvements. This approach will ensure that the routing layer remains open to integration with different security frameworks. This would allow system integrators to select the standards best suited to their specific mission requirements, whether prioritizing lightweight authentication, stronger encryption, or resilience against node compromise.

### 6.3.5 Scalability and heterogenous hardware

The experiments conducted as part of this work demonstrated the great potential of the extended AODV's route establishment and integrated error management system, as they greatly reduce data traffic in the hydroacoustic signal. To use the extended AODV in practical system hardware projects, scaling tests are necessary. An initial test has already been carried out and can be found in Figure 56 in the Appendix, as this is no longer the focus of this work. However, the results show that, in a dynamic scenario in which nodes

move and data is sent every five seconds for five minutes, including route establishment, the extended AODV behaves as described in Section 5.2.

A key challenge for future work will be to embed the extended AODV protocol into model-based system architectures. When implemented in real-world projects by system integrators or manufacturers, the routing protocol must operate seamlessly with existing subsystems. In order to achieve this, it is necessary to have open and well-defined interfaces rather than rigid dependencies. In this regard, two design choices in this work are beneficial. First, it builds on AODV, a widely recognized reference protocol already used in other industries, such as automotive. Second, it preserves IP compatibility through the protocol header design, ensuring interoperability with a wide range of communication frameworks.

### 6.3.6 AI-based Optimization

In addition to the protocol refinements outlined above, artificial intelligence presents a promising avenue for future innovation. Machine learning models have the potential to adapt routing parameters in real time, based on traffic patterns, variations in link quality, or mission-specific priorities. Predictive algorithms, for instance, could anticipate hydroacoustic channel degradation due to mobility or interference and proactively adjust gateway selection or Relay Echo intervals. While such approaches require additional computational resources and reliable training data, they have the potential to significantly increase the adaptability of the extended AODV protocol in highly dynamic environments. The design should remain modular to ensure that AI-based modules can be integrated without compromising interoperability with established standards.

### 6.3.7 Evaluation of Hydroacoustic Channel Interference for Future Adaptation

In addition to developing and simulating the extended AODV routing mechanism, initial investigations were conducted to assess the susceptibility of the hydroacoustic channel to interference. These accompanying studies aimed to analyze the impact of noise and interference on data transmission and establish preliminary statistical foundations for future routing optimization in underwater environments. Theoretical approaches based on the Bit Error Rate (BER) of quadrature phase shift keying (QPSK) modulation and practical

measurements using an acoustic modem and a controlled jammer were considered. The practical setup employed a user-defined interference signal within the 18–34 kHz frequency range.

# 7 Conclusion

This work addressed the challenge of developing a routing algorithm for autonomous, infrastructure-independent communication in maritime, Multi-Domain Operations. The communication dependencies between UAVs, USVs and UUVs were examined using the offshore wind farm Borkum Riffgrund 2 as a case study, in order to derive realistic requirements regarding latency, dynamic topologies and the limitations of underwater acoustic communication. Analysis of existing solutions revealed that the current routing algorithms were not designed for the high latency and low bandwidth that are typical of underwater environments. Consequently, they do not support Cross-Domain communication between the UxVs. In order to establish a baseline capability, a hybrid AODV model based on the classical AODV model was adapted to enable Cross-Domain communication between aerial, surface, and underwater nodes. Although this approach supports basic route discovery across domains, it does not address the fundamental challenge of reducing underwater data traffic. A high signaling load in the underwater domain increases the risk of collisions, amplifies delays, and reduces the overall reliability of routing. The central objective of this work was therefore to minimize underwater communication effort while maintaining stable Cross-Domain connectivity. In order to address this challenge, this work introduces the extended AODV routing architecture. This architecture comprises the following components:

- Hybrid AODV: A reference model enabling routing across domains without communication infrastructure.
- Extended AODV mechanisms, including:
    - internal routing logic, based on geographic Relay selection via USVs as surface gateways,
    - external routing logic, extending the internal logic by incorporating Relay Echo messages from UUVs to USVs to enable SNR-based gateway selection,
    - subsea routing logic, enabling multi-hop routing between UUVs as well as UUV–UAV communication.

- Passive error management, improving routing robustness through passive acknowledgment and reduced control overhead.

A time-discrete MATLAB simulation framework was implemented and parameterized using measurements from a Raspberry Pi-based prototype network and acoustic modem experiments. This enabled a reproducible evaluation to be carried out under realistic maritime communication constraints. The evaluation confirmed that the proposed routing concept fulfilled four out of five functional requirements and three out of four non-functional requirements, while the remaining requirements were partially fulfilled.

In summary, this work presents a routing algorithm that enables efficient, emergent Cross-Domain communication between UAVs, USVs and UUVs without the need for pre-existing infrastructure. The results demonstrate that effective routing for unmanned maritime systems requires domain-specific adaptations and reduced underwater signaling effort, rather than direct application of terrestrial routing concepts. The proposed approach provides a practical foundation for applications in offshore surveillance, subsea inspection, and the protection of critical maritime infrastructure.

# Appendix

Table 24: Results of the Simulation hybrid AODV from UAV 4 to UUV 2, 3, 13 and 15.

| **Parameter** | **Hybrid AODV** | **Hybrid AODV** | **Hybrid AODV** | **Hybrid AODV** |
|---|---|---|---|---|
| Start | 4 | 4 | 4 | 4 |
| Destination | 15 | 13 | 2 | 3 |
| Latency (ms) | 1407 | 1597 | 1477 | 1653 |
| Delta t (ms) | 10 | 10 | 10 | 10 |
| Hydroacoustic Packets: Sent | 20 | 19 | 21 | 19 |
| Hydroacoustic Bytes: Sent | 951 | 906 | 996 | 906 |
| Hydroacoustic Packets: Received | 33 | 33 | 33 | 33 |
| Hydroacoustic Bytes: Received | 1596 | 1575 | 1578 | 1575 |
| Hydroacoustic Packets: Dropped | 26 | 26 | 26 | 26 |
| Radio Packets: Sent | 18 | 17 | 19 | 17 |
| Radio Bytes: Sent | 855 | 810 | 900 | 810 |

| | | | | |
|---|---|---|---|---|
| Radio Packets: Received | 54 | 50 | 53 | 50 |
| Radio Bytes: Received | 2556 | 2376 | 2511 | 2376 |
| Radio Packets: Dropped | 39 | 36 | 37 | 36 |
| Drop Rate Hydroacoustic | 0.79 | 0.79 | 0.79 | 0.79 |
| Drop Rate Radio | 0.72 | 0.72 | 0.70 | 0.72 |

Table 25: Results of the Simulation extended AODV (internal routing logic) from UAV 4 to UUV 2, 3, 13 and 15.

| **Parameter** | **Extended AODV** | **Extended AODV** | **Extended AODV** | **Extended AODV** |
|---|---|---|---|---|
| Start | 4 | 4 | 4 | 4 |
| Gateway | 17 | 19 | 16 | 19 |
| Destination | 15 | 13 | 2 | 3 |
| Latency (ms) | 79 | 79 | 79 | 79 |
| Delta t (ms) | 10 | 10 | 10 | 10 |
| Hydroacoustic Packets: Sent | 0 | 0 | 0 | 0 |
| Hydroacoustic Bytes: Sent | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| Hydroacoustic Packets: Received | 0 | 0 | 0 | 0 |
| Hydroacoustic Bytes: Received | 0 | 0 | 0 | 0 |
| Hydroacoustic Packets: Dropped | 0 | 0 | 0 | 0 |
| Radio Packets: Sent | 53 | 53 | 53 | 53 |
| Radio Bytes: Sent | 2751 | 2751 | 2751 | 2751 |
| Radio Packets: Received | 237 | 237 | 237 | 237 |
| Radio Bytes: Received | 12429 | 12429 | 12429 | 12429 |
| Radio Packets: Dropped | 186 | 186 | 186 | 186 |
| Drop Rate Hydroacoustic | 0 | 0 | 0 | 0 |
| Drop Rate Radio | 0.78 | 0.78 | 0.78 | 0.78 |

Table 26: Results of the Simulation extended AODV (external routing logic) from UAV 4 to UUV 2, 3, 13 and 15.

| **Parameter** | **Extended AODV** | **Extended AODV** | **Extended AODV** | **Extended AODV** |
|---|---|---|---|---|
| Start | 4 | 4 | 4 | 4 |
| Gateway | 17 | 19 | 16 | 19 |
| Destination | 15 | 13 | 2 | 3 |
| Latency (ms) | 869 | 869 | 869 | 869 |
| Delta t (ms) | 800 | 800 | 800 | 800 |
| Hydroacoustic Packets: Sent | 14 | 14 | 14 | 14 |
| Hydroacoustic Bytes: Sent | 462 | 462 | 462 | 462 |
| Hydroacoustic Packets: Received | 56 | 56 | 56 | 56 |
| Hydroacoustic Bytes: Received | 1848 | 1848 | 1848 | 1848 |
| Hydroacoustic Packets: Dropped | 8 | 8 | 8 | 8 |
| Radio Packets: Sent | 53 | 53 | 53 | 53 |
| Radio Bytes: Sent | 2843 | 2843 | 2843 | 2843 |

| | | | | |
|---|---|---|---|---|
| Radio Packets: Received | 237 | 237 | 237 | 237 |
| Radio Bytes: Received | 12897 | 12897 | 12897 | 12897 |
| Radio Packets: Dropped | 186 | 186 | 186 | 186 |
| Drop Rate Hydroacoustic | 0.14 | 0.14 | 0.14 | 0.14 |
| Drop Rate Radio | 0.78 | 0.78 | 0.78 | 0.78 |

Table 27: Results of the Simulation extended AODV from UUV 6 to UUV 2, 3, 13 and 15.

| **Parameter** | **Extended AODV** | **Extended AODV** | **Extended AODV** | **Extended AODV** | **Extended AODV** |
|---|---|---|---|---|---|
| Start | 6 | 6 | 6 | 6 | 6 |
| Gateway | none | none | none | none | 14 |
| Destination | 15 | 13 | 2 | 3 | 4 |
| Latency (ms) | 2679 | 1877 | 2679 | 2679 | 2635 |
| Delta t (ms) | 10 | 10 | 10 | 10 | 10 |
| Hydroacoustic Packets: Sent | 7 | 8 | 7 | 7 | 7 |
| Hydroacoustic Bytes: Sent | 290 | 335 | 290 | 290 | 290 |

| | | | | | |
|---|---|---|---|---|---|
| Hydroacoustic Packets: Received | 16 | 25 | 16 | 16 | 16 |
| Hydroacoustic Bytes: Received | 662 | 1067 | 662 | 662 | 662 |
| Hydroacoustic Packets: Dropped | 11 | 19 | 11 | 11 | 11 |
| Radio Packets: Sent | 55 | 55 | 55 | 55 | 17 |
| Radio Bytes: Sent | 2808 | 2808 | 2808 | 2808 | 765 |
| Radio Packets: Received | 237 | 237 | 237 | 237 | 68 |
| Radio Bytes: Received | 12213 | 12213 | 12213 | 12213 | 3060 |
| Radio Packets: Dropped | 184 | 184 | 184 | 184 | 51 |
| Drop Rate Hydroacoustic | 0.69 | 0.76 | 0.69 | 0.69 | 0.69 |
| Drop Rate Radio | 0.78 | 0.78 | 0.78 | 0.78 | 0.75 |

Table 28: Results of the Simulation hybrid AODV from UUV 6 to UUV 2, 3, 13 and 15.

| **Parameter** | **Hybrid AODV** | **Hybrid AODV** | **Hybrid AODV** | **Hybrid AODV** | **Hybrid AODV** |
|---|---|---|---|---|---|
| Start | 6 | 6 | 6 | 6 | 6 |
| Destination | 15 | 13 | 2 | 3 | 4 |
| Latency (ms) | 2611 | 1877 | 2681 | 2857 | 1729 |
| Delta t (ms) | 10 | 10 | 10 | 10 | 10 |
| Hydroacoustic Packets: Sent | 22 | 19 | 23 | 21 | 21 |
| Hydroacoustic Bytes: Sent | 1041 | 906 | 1086 | 996 | 999 |
| Hydroacoustic Packets: Received | 45 | 38 | 46 | 44 | 44 |
| Hydroacoustic Bytes: Received | 2118 | 1797 | 2172 | 2076 | 2088 |
| Hydroacoustic Packets: Dropped | 38 | 30 | 39 | 37 | 38 |
| Radio Packets: Sent | 18 | 15 | 19 | 17 | 17 |
| Radio Bytes: Sent | 855 | 720 | 900 | 810 | 807 |

| | | | | | |
|---|---|---|---|---|---|
| Radio Packets: Received | 72 | 60 | 78 | 68 | 68 |
| Radio Bytes: Received | 3420 | 2880 | 3690 | 3240 | 3228 |
| Radio Packets: Dropped | 56 | 48 | 61 | 53 | 52 |
| Drop Rate Hydroacoustic | 0.84 | 0.79 | 0.85 | 0.84 | 0.86 |
| Drop Rate Radio | 0.78 | 0.80 | 0.78 | 0.78 | 0.76 |

Table 29: Results of the SNR measurement (Simulated for the Use Case): $P_t$ = 166 dB re 1 µPa.

| **Distance (m)** | **SNR (dB re 1 µPa)** |
|---|---|
| 100 | 48.06 |
| 200 | 42.98 |
| 300 | 39.77 |
| 400 | 37.53 |
| 500 | 35.30 |
| 600 | 33.55 |
| 700 | 31.97 |
| 800 | 30.53 |
| 900 | 29.19 |

| | |
|---|---|
| 1000 | 27.94 |

Table 30: Results of the Simulation extended AODV: No Error Management.

| **Parameter** | **Number** |
|---|---|
| Radio packets sent | 2143 |
| Radio bytes sent | 93226 |
| Radio packets received | 7268 |
| Radio bytes received | 316063 |
| Radio packets dropped | 5439 |
| Hydroacoustic packet sent | 75 |
| Hydroacoustic bytes sent | 2302 |
| Hydroacoustic packets received | 336 |
| Hydroacoustic bytes received | 10925 |
| Hydroacoustic packets dropped | 315 |
| **Hydroacoustic Error detected** | **0** |
| **Hydroacoustic Error fixed** | **0** |
| **Data packet sent** | **450** |
| **Data packet received** | **191** |

Table 31: Results of the Simulation extended AODV. Error Management.

| **Parameter** | **Number** |
|---|---|
| Radio packets sent | 3622 |

| | |
|---|---|
| Radio bytes sent | 158222 |
| Radio packets received | 15895 |
| Radio bytes received | 694213 |
| Radio packets dropped | 12366 |
| Hydroacoustic packet sent | 138 |
| Hydroacoustic bytes sent | 4828 |
| Hydroacoustic packets received | 605 |
| Hydroacoustic bytes received | 20341 |
| Hydroacoustic packets dropped | 554 |
| **Hydroacoustic Error detected** | **37** |
| **Hydroacoustic Error fixed** | **36** |
| **Data packet sent** | **478** |
| **Data packet received** | **476** |

Table 32: Recordings of offset and processing time for parameterization in the above-water domain.

| **Trial** | **Processing time between Pis (ms)** | **Delay/ Offset time between Pis (ms)** |
|---|---|---|
| **1** | 4.5 | 9.0 |
| **2** | 4.0 | 8.0 |
| **3** | 4.5 | 9.0 |
| **4** | 5.0 | 8.0 |
| **5** | 4.5 | 8.0 |
| **6** | 4.5 | 8.0 |
| **7** | 5.0 | 9.0 |
| **8** | 5.0 | 8.0 |
| **9** | 5.0 | 7.0 |
| **10** | 5.5 | 8.0 |
| **11** | 4.0 | 9.0 |
| **12** | 4.0 | 7.0 |
| **13** | 4.0 | 8.0 |

| | | |
|---|---|---|
| **14** | 4.0 | 9.0 |
| **15** | 4.0 | 7.0 |
| **16** | 5.5 | 9.0 |
| **17** | 4.5 | 7.0 |
| **18** | 4.0 | 7.0 |
| **19** | 4.5 | 7.0 |
| **20** | 4.5 | 9.0 |
| **21** | 5.0 | 7.0 |
| **22** | 4.5 | 8.0 |
| **23** | 5.0 | 7.0 |
| **24** | 4.5 | 7.0 |
| **25** | 4.5 | 7.0 |
| **26** | 5.0 | 7.0 |
| **27** | 4.0 | 8.0 |
| **28** | 5.0 | 8.0 |
| **29** | 4.5 | 7.0 |
| **30** | 4.5 | 7.0 |
| **31** | 5.0 | 8.0 |
| **32** | 4.5 | 8.0 |
| **33** | 4.5 | 7.0 |
| **34** | 5.0 | 7.0 |
| **35** | 4.5 | 8.0 |
| **36** | 5.0 | 7.0 |
| **37** | 4.0 | 7.0 |
| **38** | 4.5 | 8.0 |
| **39** | 5.0 | 7.0 |
| **40** | 4.5 | 8.0 |

Table 33: Route and Data Establishment Time (above-water).

| Trial | Delta t (ms) Route est. | Delta t (ms) Data sent |
|---|---|---|
| **1** | 59 | 65 |
| **2** | 56 | 62 |
| **3** | 55 | 61 |
| **4** | 57 | 63 |
| **5** | 56 | 62 |
| **6** | 56 | 60 |
| **7** | 56 | 62 |
| **8** | 57 | 63 |
| **9** | 57 | 63 |
| **10** | 57 | 63 |
| **11** | 56 | 62 |
| **12** | 57 | 63 |
| **13** | 56 | 62 |
| **14** | 57 | 63 |
| **15** | 56 | 63 |
| **16** | 58 | 65 |

| | | |
|---|---|---|
| **17** | 59 | 65 |
| **18** | 56 | 62 |
| **19** | 57 | 64 |
| **20** | 58 | 65 |
| **21** | 56 | 62 |
| **22** | 57 | 65 |
| **23** | 56 | 62 |
| **24** | 57 | 63 |
| **25** | 58 | 63 |
| **26** | 56 | 62 |
| **27** | 57 | 63 |
| **28** | 56 | 63 |
| **29** | 57 | 64 |
| **30** | 57 | 64 |
| **31** | 57 | 63 |
| **32** | 58 | 64 |
| **33** | 58 | 64 |
| **34** | 56 | 62 |
| **35** | 56 | 62 |
| **36** | 57 | 63 |
| **37** | 56 | 63 |
| **38** | 57 | 62 |
| **39** | 58 | 63 |
| **40** | 57 | 63 |

Table 34: SNR Measurements at different distances from 5-20m. All values are given in dB re 1 µPa.

| Trial | Type | 5m | 10m | 15m | 20m |
|---|---|---|---|---|---|
| **1** | RSSI | 133 | 132 | 128 | 126 |
| | Noise | 85 | 85 | 85 | 85 |
| | SNR | 48 | 47 | 43 | 41 |
| **2** | RSSI | 133 | 132 | 128 | 126 |
| | Noise | 85 | 85 | 85 | 85 |
| | SNR | 48 | 47 | 43 | 41 |
| **3** | RSSI | 133 | 129 | 127 | 126 |
| | Noise | 85 | 85 | 85 | 86 |
| | SNR | 48 | 44 | 42 | 40 |
| **4** | RSSI | 134 | 132 | 126 | 122 |
| | Noise | 85 | 85 | 85 | 85 |
| | SNR | 49 | 47 | 41 | 37 |
| **5** | RSSI | 134 | 128 | 127 | 126 |
| | Noise | 85 | 85 | 85 | 85 |
| | SNR | 49 | 43 | 42 | 41 |
| **6** | RSSI | 134 | 132 | 128 | 122 |
| | Noise | 85 | 85 | 86 | 85 |
| | SNR | 49 | 47 | 42 | 37 |
| **7** | RSSI | 133 | 128 | 127 | 123 |

| | | | | | |
|---|---|---|---|---|---|
| | Noise | 86 | 85 | 85 | 87 |
| | SNR | 47 | 43 | 42 | 36 |
| **8** | RSSI | 134 | 132 | 128 | 124 |
| | Noise | 85 | 86 | 85 | 85 |
| | SNR | 49 | 46 | 43 | 39 |
| **9** | RSSI | 134 | 132 | 127 | 124 |
| | Noise | 85 | 86 | 86 | 85 |
| | SNR | 49 | 46 | 41 | 39 |
| **10** | RSSI | 134 | 132 | 128 | 122 |
| | Noise | 86 | 86 | 85 | 86 |
| | SNR | 48 | 46 | 43 | 36 |

Table 35: Duration of the underwater message Relay Echo for distances of 5-20m.

| **Trial** | **Type** | **5m** | **10m** | **15m** | **20m** |
|---|---|---|---|---|---|
| **1** | Send Start | 20:55:29.840 | 22:00:30.404 | 22:30:33.098 | 23:13:14.156 |
| | Receive End | 20:55:30.185 | 22:00:30.750 | 22:30:33.452 | 23:13:14.514 |
| | Delta t [ms] | 345 | 346 | 354 | 358 |
| **2** | Send Start | 21:22:00.428 | 22:01:38.370 | 22:32:44.088 | 23:14:25.245 |
| | Receive End | 21:22:00.777 | 22:01:38.720 | 22:32:44.434 | 23:14:25.600 |
| | Delta t [ms] | 349 | 350 | 346 | 355 |
| **3** | Send Start | 21:27:47.661 | 22:03:18.924 | 22:34:22.076 | 22:15:87.587 |
| | Receive End | 21:27:48.011 | 22:03:19.274 | 22:34:22.429 | 22:13:36.942 |
| | Delta t [ms] | 350 | 350 | 353 | 355 |
| **4** | Send Start | 21:31:29.516 | 22:05:53.095 | 22:35:31.043 | 22:13:47.067 |
| | Receive End | 21:31:29.863 | 22:05:53.446 | 22:35:31.395 | 22:13:47.425 |
| | Delta t [ms] | 351 | 351 | 352 | 358 |
| **5** | Send Start | 21:33:42.008 | 22:07:03.616 | 22:37:40.491 | 22:14:00.013 |
| | Receive End | 21:33:42.356 | 22:07:03.968 | 22:37:40.841 | 22:14:00.367 |
| | Delta t [ms] | 348 | 352 | 350 | 354 |
| **6** | Send Start | 21:34:89.124 | 22:11:12.379 | 22:38:21.389 | 22:15:11.175 |
| | Receive End | 21:34:89.471 | 22:11:12.729 | 22:38:21.642 | 22:15:11.530 |
| | Delta t [ms] | 347 | 350 | 353 | 355 |
| **7** | Send Start | 21:36:07.983 | 22:12:11.467 | 22:39:53.763 | 22:16:43.674 |

| | | | | | |
|---|---|---|---|---|---|
| | Receive End | 21:36:08.330 | 22:12:11.817 | 22:39:54.114 | 22:16:44.032 |
| | Delta t [ms] | 347 | 350 | 351 | 358 |
| **8** | Send Start | 21:37:98.125 | 22:13:98.987 | 22:41:02.456 | 22:18:00.343 |
| | Receive End | 21:37:98.470 | 22:13:99.337 | 22:41:02.811 | 22:18:00.700 |
| | Delta t [ms] | 345 | 350 | 355 | 357 |
| **9** | Send Start | 21:43:42.512 | 22:15:43.876 | 22:45:08.672 | 22:20:89.562 |
| | Receive End | 21:43:42.861 | 22:15:44.223 | 22:45:09.025 | 22:20:89.920 |
| | Delta t [ms] | 349 | 347 | 353 | 358 |
| **10** | Send Start | 21:45:02.641 | 22:18:01.007 | 22:46:78.189 | 22:22:09.451 |
| | Receive End | 21:45:02.991 | 22:18:01.358 | 22:46:78.542 | 22:22:09.809 |
| | Delta t [ms] | 350 | 351 | 353 | 358 |

Table 36: Duration of the underwater message Data for distances of 5-20m.

| **Trial** | **Type** | **5m** | **10m** | **15m** | **20m** |
|---|---|---|---|---|---|
| **1** | Send Start | 14:42:11.529 | 15:35:13.974 | 15:45:03.526 | 16:20:10.479 |
| | Receive End | 14:42:11.994 | 15:35:14.431 | 15:45:03.991 | 16:20:10.970 |
| | Delta t [ms] | 465 | 457 | 465 | 491 |
| **2** | Send Start | 14:52:58.858 | 15:19:43.124 | 15:47:28.543 | 16:21:29.490 |
| | Receive End | 14:52:59.319 | 15:19:43.601 | 15:47:29.028 | 16:21:29.974 |
| | Delta t [ms] | 461 | 477 | 485 | 484 |
| **3** | Send Start | 14:56:33.891 | 15:20:55.138 | 15:52:30.556 | 16:22:31.498 |
| | Receive End | 14:56:34.361 | 15:20:55.615 | 15:52:31.026 | 16:22:31.975 |
| | Delta t [ms] | 476 | 477 | 470 | 477 |
| **4** | Send Start | 14:58:16.907 | 15:21:47.146 | 15:53:43.566 | 16:28:36.506 |
| | Receive End | 14:58:17.370 | 15:21:47.626 | 15:53:44.033 | 16:28:36.993 |
| | Delta t [ms] | 463 | 480 | 467 | 487 |
| **5** | Send Start | 14:59:53.898 | 15:22:54.159 | 15:55:42.573 | 16:30:38.514 |
| | Receive End | 14:59:54.374 | 15:22:54.625 | 15:55:43.049 | 16:30:39.001 |
| | Delta t [ms] | 476 | 466 | 476 | 487 |

| | | | | | |
|---|---|---|---|---|---|
| **6** | Send Start | 15:05:09.356 | 15:24:25.187 | 15:56:67.769 | 16:31:34.421 |
| | Receive End | 15:05:09.821 | 15:24:25.653 | 15:57:68.254 | 16:31:34.921 |
| | Delta t [ms] | 465 | 466 | 485 | 500 |
| **7** | Send Start | 15:08:11.568 | 15:25:84.898 | 15:59:21.465 | 16:33:02.724 |
| | Receive End | 15:08:12.033 | 15:25:85.363 | 15:59:21.955 | 16:33:03.223 |
| | Delta t [ms] | 465 | 465 | 490 | 499 |
| 8 | Send Start | 15:14:23.125 | 15:27:23.276 | 16:01:02.457 | 16:37:31.123 |
| | Receive End | 15:14:23.590 | 15:27:23.755 | 16:01:02.937 | 16:37:31.614 |
| | Delta t [ms] | 465 | 479 | 480 | 491 |
| **9** | Send Start | 15:17:00.498 | 15:28:87.542 | 16:03:00.456 | 16:38:12.589 |
| | Receive End | 15:17:00.960 | 15:28:88.019 | 16:03:00.938 | 16:38:13.087 |
| | Delta t [ms] | 462 | 477 | 482 | 498 |
| **10** | Send Start | 15:20:15.359 | 15:30:00.367 | 16:04:89.124 | 16:39:87.571 |
| | Receive End | 15:20:15.826 | 15:30:00.844 | 16:04:89.606 | 16:39:88.062 |
| | Delta t [ms] | 467 | 477 | 482 | 491 |

Table 37: Measurement values for the entire runtime from transmitter Pi to listener Pi (radio + hydroacoustic).

| **Trial** | **Type** | **5m** |
|---|---|---|
| **1** | Send Start | 06:00:45.801 |
| | Receive End | 06:00:46.275 |
| | Delta t [ms] | 474 5 |
| **2** | Send Start | 06:00:56.585 |
| | Receive End | 06:00:57.085 |
| | Delta t [ms] | 500 |
| **3** | Send Start | 06:01:11.015 |
| | Receive End | 06:01:11.491 |
| | Delta t [ms] | 476 |
| **4** | Send Start | 06:02:13.793 |
| | Receive End | 06:02:14.327 |
| | Delta t [ms] | 534 |
| **5** | Send Start | 06:02:26.399 |
| | Receive End | 06:02:26.933 |
| | Delta t [ms] | 534 |
| **6** | Send Start | 06:03:09.891 |
| | Receive End | 06:03:10.357 |

| | | |
|---|---|---|
| | Delta t [ms] | 466 |
| **7** | Send Start | 06:03:26.693 |
| | Receive End | 06:03:27.167 |
| | Delta t [ms] | 474 |
| **8** | Send Start | 06:03:38.106 |
| | Receive End | 06:03:38.575 |
| | Delta t [ms] | 469 |
| **9** | Send Start | 06:03:48.335 |
| | Receive End | 06:03:48.879 |
| | Delta t [ms] | 544 |
| **10** | Send Start | 06:03:56.635 |
| | Receive End | 06:03:57.184 |
| | Delta t [ms] | 549 |

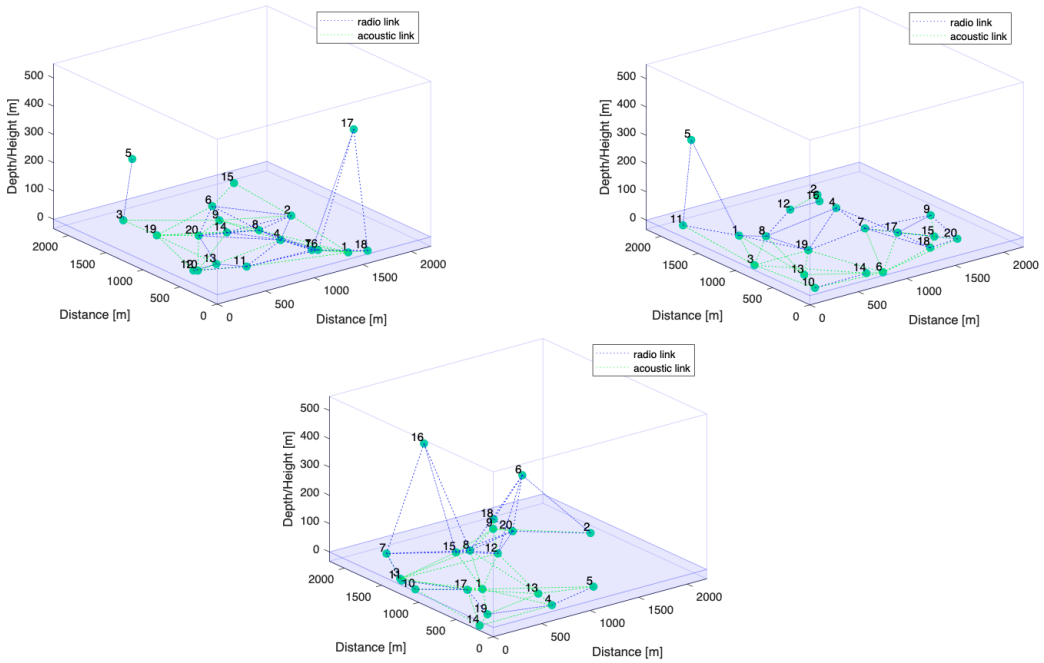


Figure 55: Dynamics of different simulation topologies for scaling and testing error management.

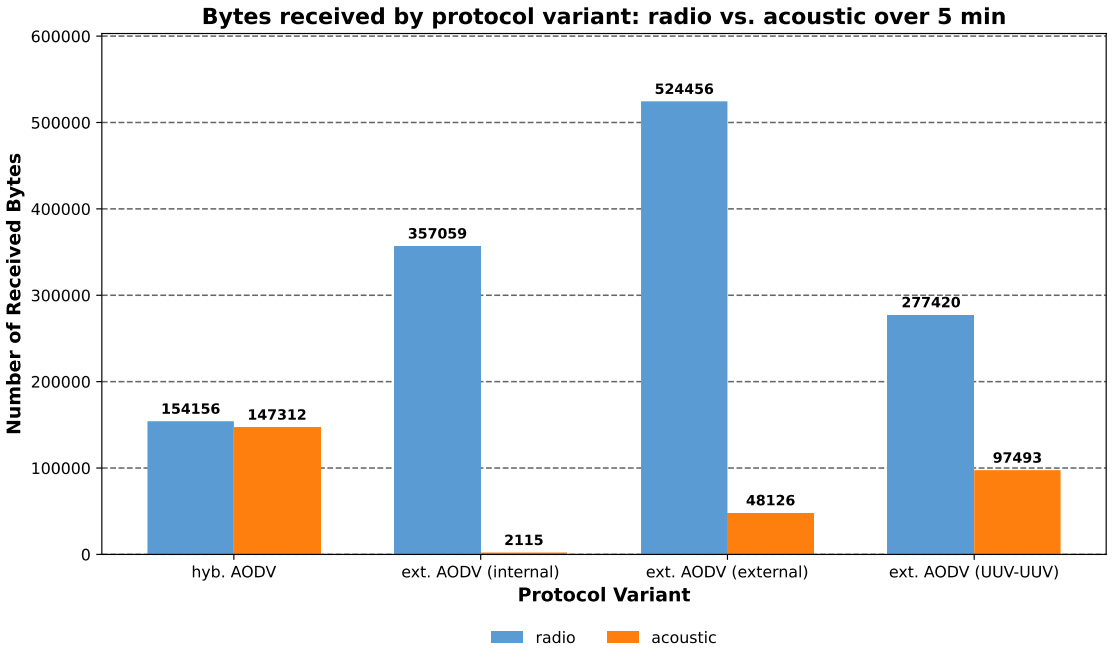


Figure 56: Overview of the bytes received by the four variants hybrid, extended (internal and external), and extended (UUV-UUV) AODV over 5 minutes. Routes are established and data is sent every 5 seconds. The topology is dynamic and changes at a speed of 10 m/s.

*In this work, the following AI-based tools were used to support spelling: ChatGPT and DeeplWrite.*

*The icons used in this dissertation were obtained on December 10, 2025 from icon-icons.com.*
*Some icons fall under the CC BY 4.0 license and the Apache License 2.0 are referenced accordingly. The following icons are used:*

1. *USV*(  )*: https://icon-icons.com/icon/boat/148230*
2. *UUV*(  )*: https://icon-icons.com/icon/submarine/39459*
3. *Antenna*(  )*: https://icon-icons.com/icon/antenna-with-signal-transmission/56571*
4. *UAV*(  ): *https://icon-icons.com/icon/drone/216206*
5. *Acoustic sound* (  ): *https://icon-icons.com/icon/acoustic-sound/155861*

*(All icons were only resized for visualization purposes)*

*Some of the results presented in this dissertation are based on research carried out as part of supervised student projects.*
*At the time of publication, some of these results are still in the process of being published or patented.*

# Bibliography

Ahmad, I., & Chang, K. (2019). Effective SNR Mapping and Link Adaptation Strategy for Next-Generation Underwater Acoustic Communications Networks: A Cross-Layer Approach. *IEEE Access*, *7*, 44150–44164. https://doi.org/10.1109/ACCESS.2019.2908018

Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks*, *3*(3), 257–279. https://doi.org/10.1016/j.adhoc.2005.01.004

Al-Absi, M. A., Al-Absi, A. A., Sain, M., & Lee, H. (2021). Moving Ad Hoc Networks—A Comparative Study. *Sustainability*, *13*(11). https://doi.org/10.3390/su13116187

Alam, M., Ahmed, N., Matam, R., & Barbhuiya, F. A. (2024). Analyzing the suitability of IEEE 802.11ah for next generation Internet of Things: A comparative study. *Ad Hoc Networks*, *156*, 103437. https://doi.org/10.1016/j.adhoc.2024.103437

Alqurashi, F. S., Trichili, A., Saeed, N., Ooi, B. S., & Alouini, M.-S. (2023). Maritime Communications: A Survey on Enabling Technologies, Opportunities, and Challenges. *IEEE Internet of Things Journal*, *10*(4), 3525–3547. https://doi.org/10.1109/JIOT.2022.3219674

Aman, W., Giorgi, F., Attenni, G., Al-Kuwari, S., Illi, E., Qaraqe, M., Maselli, G., & Di Pietro, R. (2023). Expanding Boundaries: Cross-Media Routing for Seamless Underwater and Aerial Communication. *arXiv Preprint arXiv:2307.12643*. https://doi.org/10.1016/j.adhoc.2024.103437

Ansa Shermin, S., Mehta, B., Sarang, C. D., & Chitre, M. A. (2024). Load-adaptive MAC protocol for frontier detection in Underwater Mobile Sensor Network. *Ad Hoc Networks*, *165*(10364), 1. https://doi.org/10.1016/j.adhoc.2024.103641

Arega, K. L., Raga, G., & Bareto, R. (2020). Survey on performance analysis of AODV, DSR and DSDV in MANET. *Computer Engineering and Intelligent Systems*, *11*(3), 23–32. https://doi.org/10.48550/arXiv.1402.2217

Aydin, S., & Onur, T. (2020). *Investigation of parameters affecting underwater communication channel*. https://doi.org/10.21272/jes.2020.7(1).f4

Azad, S., Casari, P., & Zorzi, M. (2014). Multipath routing with limited cross-path interference in underwater networks. *IEEE Wireless Communications Letters*, *3*(5), 465–468. https://doi.org/10.1109/LWC.2014.2342232

Baharuddin, M., Panggalo, S., & Gufran, M. (2020). *Performance of routing protocol olsr and batman in multi-hop and mesh ad hoc network on raspberry pi*. *875*(1), 012046. https://doi.org/10.1088/1757-899X/875/1/012046

Baños-Gonzalez, V., Afaqui, M. S., Lopez-Aguilera, E., & Garcia-Villegas, E. (2016). IEEE 802.11ah: A Technology to Face the IoT Challenge. *Sensors*, *16*(11). https://doi.org/10.3390/s16111960

Bauer, J., Klein, A., Schütz, B., & Stoppe, J. (2025). Investigating the Impact of Communication Delays and Bandwidth Restrictions on Remote Operations of Unmanned Systems. *Drones and Unmanned Systems*, 273.

Benadí, A. G., Cadena-Munoz, J., del Río Fernandez, J., Juan, X. R., & Manuel-Làzaro, A. (2015). Good Practice Guide for calibrating a hydrophone" in situ" with a non-omnidirectional source at 10 kHz. *Acta Imeko*, *4*(1), 26–34. https://doi.org/10.21014/acta_imeko.v4i1.157

Bindra, P., Kaur, J., & Singh, G. (2013). Effect of TTL parameter variation on performance of AODV route discovery process. *International Journal of Computer Applications*, *70*(4). https://doi.org/10.5120/11951-7769

Brandl, K., Hahn, A., & Weid, J. (2025). *Securing Maritime Critical Infrastructure: A Multi-Domain AODV-Based Communication Protocol for UxVs*. 10–21. https://doi.org/10.1007/978-3-031-87763-6_2

Brandl, K., Weid, J., Hahn, A., & de Paillette, T. D. (2024). *Enhancing Communcation in Multi-Domain Ad-Hoc Networks for Maritime Critical Infrastructure Protection*. 01–07. https://doi.org/10.1109/OCEANS55160.2024.10754334

Brinkmann, M., & Hahn, A. (2017). Testbed architecture for maritime cyber physical systems. *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 923–928. https://doi.org/10.1109/INDIN.2017.8104895

BSI-Kritisverordnung (BSI-KritisV), Pub. L. No. BGBl. I S. 958 (2016).

Bueger, C., & Liebetrau, T. (2023). Critical maritime infrastructure protection: What's the trouble? *Marine Policy*, *155*, 105772. https://doi.org/10.1016/j.marpol.2023.105772

Bueger, C., Liebetrau, T., & Franken, J. (2022). Security threats to undersea communications cables and infrastructure–consequences for the EU. *Report for SEDE Committee of the European Parliament, PE702*, *557*.

Busacca, F., Galluccio, L., Palazzo, S., Panebianco, A., Qi, Z., & Pompili, D. (2024). Adaptive versus predictive techniques in underwater acoustic communication networks. *Computer Networks*, *252*, 110679. https://doi.org/10.1016/j.comnet.2024.110679

Cannon, B. J., & Bhatt, P. (2024). The Quad and Submarine Cable Protection in the Indo-Pacific: Policy Recommendations. *The Institute for Security and Development Policy Policy Brief*, 1–12.

Ching, T. W., Aman, A. H. M., Azamuddin, W. M. H., & Attarbashi, Z. S. (2021). *Performance evaluation of AODV routing protocol in MANET using NS-3 simulator*. 1–4. https://doi.org/10.1109/CRC50527.2021.9392519

Chitre, M. (2020). *Underwater networks handbook*. Retrieved December 06, 2025, from https://unetstack.net/handbook/unet-handbook.pdf

Clare, M. (2023). *Submarine Cable Protection and the Environment*. An Update from the ICPC. International Cable Protection Committee (ICPC). Retrieved December 6, 2025, from https://iscpc.org/publications/submarine-cable-protection-and-the-environment/ICPC_Public_EU_September_2023.pdf

Coutinho, R. W. L., Boukerche, A., Vieira, L. F. M., & Loureiro, A. A. F. (2016). Geographic and Opportunistic Routing for Underwater Sensor Networks. *IEEE Transactions on Computers*, *65*(2), 548–561. https://doi.org/10.1109/TC.2015.2423677

D. Ramphull, A. Mungur, S. Armoogum, & S. Pudaruth. (2021). A Review of Mobile Ad hoc NETwork (MANET) Protocols and their Applications. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 204–211. https://doi.org/10.1109/ICICCS51141.2021.9432258

Deepika, S., Nishanth, N., & Mujeeb, A. (2021). An Assessment of Recent Advances in AODV Routing Protocol Path Optimization Algorithms for Mobile Ad hoc Networks. *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*, 1–6. https://doi.org/10.1109/ICMSS53060.2021.9673632

Ding, J., Wang, C., Jiang, M., Lin, S., & Yang, H. (2021). *Cross media routing and clustering algorithm for autonomous marine systems*. 289–296. https://doi.org/10.1109/ICPS49255.2021.9468149

Ding, J., Zhang, H.-T., & Hu, B.-B. (2024). *Coordinated Landing Control for Cross-Domain UAV-USV Fleets Using Heterogeneous-Feature Matching*. 12041–12047. https://doi.org/10.1109/ICRA57147.2024.10611392

Dong, S. Y. (2016). Optimization of OLSR routing protocol in UAV ad HOC network. *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 90–94. https://doi.org/10.1109/ICCWAMTIP.2016.8079811

Elgharbi, S. E., Iturralde, M., Dupuis, Y., & Gaugue, A. (2025). Maritime monitoring through LoRaWAN: Resilient decentralised mesh networks for enhanced data transmission. *Computer Communications*, 108276. https://doi.org/10.1016/j.comcom.2025.108276

Elias, S. J., Mohd Warip, M. N., Mansor, S., Muhamat Dawam, S. R., & Mansor, Ab. R. (2017). Experimental Model of Congestion Control Vehicular Ad Hoc Network Using OMNET++. In A.-R. Ahmad, L. K. Kor, I. Ahmad, & Z. Idrus (Eds.), *Proceedings of the International Conference on Computing, Mathematics and Statistics (iCMS 2015)* (pp. 25–35). Springer Singapore. https://doi.org/10.1007/978-981-10-2772-7_3

EvoLogics. (2014, February). *S2CR 18/34 Product Information*. Retrieved December 6, 2025, from https://www.evologics.com/product/attachments/s2c-r-18-34-datasheet-102

EvoLogics. (2020, March). *S2C Reference Manual*. Firmware Version 2.0 Edition Networking.

Gangopadhyay, S., & Jain, V. K. (2023). A Position-Based Modified OLSR Routing Protocol for Flying Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, *72*(9), 12087–12098. https://doi.org/10.1109/TVT.2023.3265704

Ganz, A., Camellini, M., Hine, E., Novelli, C., Roberts, H., & Floridi, L. (2024). Submarine cables and the risks to digital sovereignty. *Minds and Machines*, *34*(3), 31. https://doi.org/10.2139/ssrn.4693206

Garcia Benadí, A. (2014). The electrical noise level of the hydrophone and its drift with the changed of the environment parameters. *Instrumentation Viewpoint*, *17*, 61–62.

Goetz, M., & Nissen, I. (2012). GUWMANET-multicast routing in underwater acoustic networks. *2012 Military Communications and Information Systems Conference, MCC 2012*, 1–8.

Goetz, M., & Nissen, I. (2015). *Akustisches mobiles ad-hoc Netzwerkprotokoll—GUWMANET*. DAGA 2015 - 41. Jahrestagung für Akustik

Gola, K. K., Dhingra, M., Gupta, B., & Rathore, R. (2023). An empirical study on underwater acoustic sensor networks based on localization and routing approaches. *Advances in Engineering Software*, *175*, 103319. https://doi.org/10.1016/j.advengsoft.2022.103319

Haenggi, M. (2010). Local Delay in Static and Highly Mobile Poisson Networks with ALOHA. *2010 IEEE International Conference on Communications*, 1–5. https://doi.org/10.1109/ICC.2010.5502205

Hakim, A. T., Dewi, A. P., Perdana, D., & Kurnaman, C. N. (2018). Energy Consumption Analysis of DBR and VBF Protocols in Underwater Sensor Networks Using Aqua-Sim at Network Simulator 2. *Jurnal Infotel*, *10*(4), 170–177. https://doi.org/10.20895/infotel.v10i4.394

Hyder, W., Pabani, J. K., Luque-Nieto, M.-Á., Laghari, A. A., & Otero, P. (2023). Self-Organized Ad Hoc Mobile (SOAM) Underwater Sensor Networks. *IEEE Sensors Journal*, *23*(2), 1635–1644. https://doi.org/10.1109/JSEN.2022.3224993

Ibrahim, D. M., Eltobely, T. E., Sallam, E., & Fahmy, M. (2015). Bounded side-based clustering vbf routing protocol in underwater wireless sensor networks. *International Journal on Advances in Networks and Services*, *8*, 130–138.

ITU-T. (1994). *Information technology – Open Systems Interconnection – Basic Reference Model: The basic model*. International Telecommunication Union (ITU-T). https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=2820

Iurii Voitenko. (2017). *SimpleManet*. Retrieved March 7, 2025, from https://github.com/iuriivoitenko/simpleMANET. License: GNU General Public License v3.0.

IWR. (2017). Fundament für Umspannstation im Windpark Borkum Riffgrund 2 ist fertig. Retrieved December 8, 2025, from *Https://Www.Offshore-Windindustrie.de/News/Nachrichten/Artikel-34167-Fundament-Fur-Umspannstation-Im-Windpark-Borkum-Riffgrund-2-Ist-Fertig*.

Jarzębowicz, A., & Weichbroth, P. (2021). A Qualitative Study on Non-Functional Requirements in Agile Software Development. *IEEE Access*, *9*, 40458–40475. https://doi.org/10.1109/ACCESS.2021.3064424

Jiang, Z., Guan, Q., Chen, F., Wei, N., Ji, F., & Yu, H. (2023). Opportunistic hybrid routing protocol for acoustic-radio cooperative networks. *IEEE Internet of Things Journal*, *10*(21), 19014–19026. https://doi.org/10.1109/JIOT.2023.3281084

Jubair, M. A., Mostafa, S. A., Muniyandi, R. C., Mahdin, H., Mustapha, A., Hassan, M. H., Mahmoud, M. A., Al-Jawhar, Y. A., Al-Khaleefa, A. S., & Mahmood, A. J. (2019). Bat Optimized Link State Routing Protocol for Energy-Aware Mobile Ad-Hoc Networks. *Symmetry*, *11*(11). https://doi.org/10.3390/sym11111409

Junejo, N. U. R., Sattar, M., Adnan, S., Sun, H., Adam, A. B., Hassan, A., & Esmaiel, H. (2023). A survey on physical layer techniques and challenges in underwater communication systems. *Journal of Marine Science and Engineering*, *11*(4), 885. https://doi.org/10.3390/jmse11040885

Kark, K. (2006). *Antennen und Strahlungsfelder* (Vol. 2). Springer. https://doi.org/10.1007/978-3-658-22319-9

Kartha, J. J., & Jacob, L. (2015). Delay and Lifetime Performance of Underwater Wireless Sensor Networks with Mobile Element Based Data Collection. *International Journal of Distributed Sensor Networks*, *11*(5), 128757. https://doi.org/10.1155/2015/128757

Kaur, P., & Singh, A. (2018). *Nature-inspired optimization techniques in VANETs and FANETs: A survey*. 651–663. https://doi.org/10.1007/978-981-10-8237-5_63

Khan, A., Ali, I., Ghani, A., Khan, N., Alsaqer, M., Rahman, A. U., & Mahmood, H. (2018). Routing Protocols for Underwater Wireless Sensor Networks: Taxonomy, Research Challenges, Routing Strategies and Future Directions. *Sensors*, *18*(5). https://doi.org/10.3390/s18051619

Kiran, K., Kaushik, N. P., Sharath, S., Shenoy, P. D., Venugopal, K. R., & Prabhu, V. T. (2018). Experimental Evaluation of BATMAN and BATMAN-Adv Routing Protocols in a Mobile Testbed. *TENCON 2018 - 2018 IEEE Region 10 Conference*, 1538–1543. https://doi.org/10.1109/TENCON.2018.8650222

Kumar, B. V. S., & Padmavathy, N. (2020). A Hybrid Link Reliability Model for Estimating Path Reliability of Mobile Ad Hoc Network. *Procedia Computer Science*, *171*, 2177–2185. https://doi.org/10.1016/j.procs.2020.04.235

Kurniawan, A., Kristalina, P., & Hadi, M. Z. S. (2020). *Performance analysis of routing protocols AODV, OLSR and DSDV on MANET using NS3*. 199–206. https://doi.org/10.1109/IES50839.2020.9231690

Li, B., Zheng, S., & Tong, F. (2019). Bit-error rate based Doppler estimation for shallow water acoustic OFDM communication. *Ocean Engineering*, *182*, 203–210. https://doi.org/10.1016/j.oceaneng.2019.04.045

Li, Z., Min, G., Ren, P., Luo, C., Zhao, L., & Luo, C. (2024). Ubiquitous and Robust UxV Networks: Overviews, Solutions, Challenges, and Opportunities. *IEEE Network*, *38*(2), 26–34. https://doi.org/10.1109/MNET.2024.3352691

Liu, L., Liu, J., Qian, H., & Zhu, J. (2018). Performance Evaluation of BATMAN-Adv Wireless Mesh Network Routing Algorithms. *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 122–127. https://doi.org/10.1109/CSCloud/EdgeCom.2018.00030

Luo, J., Chen, Y., Wu, M., & Yang, Y. (2021). A Survey of Routing Protocols for Underwater Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, *23*(1), 137–160. https://doi.org/10.1109/COMST.2020.3048190

Luo, X., Chen, H.-H., & Guo, Q. (2024). LEO/VLEO Satellite Communications in 6G and Beyond Networks–Technologies, Applications, and Challenges. *IEEE Network*, *38*(5), 273–285. https://doi.org/10.1109/MNET.2024.3353806

Martin, R., Rajasekaran, S., & Peng, Z. (2017). Aqua-Sim Next Generation: An NS-3 Based Underwater Sensor Network Simulator. *Proceedings of the 12th International Conference on Underwater Networks & Systems*. https://doi.org/10.1145/3148675.3148679

National Physical Laboratory. (2000). *Underwater Acoustics: Technical Guides—Speed of Sound in Sea-Water*. National Physical Laboratory, Teddington, Middlesex, UK, TW11 0LW. Retrieved December 8, 2025, from https://www.commtec.com/library/technical_papers/speedsw.pdf

Nayyar, A. (2018). Flying Adhoc Network (FANETs): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP. *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, 1–9. https://doi.org/10.1109/ICABCD.2018.8465130

Orfanidis, G., Apostolidis, S., Kapoutsis, A., Ioannidis, K., Kosmatopoulos, E., Vrochidis, S., & Kompatsiaris, I. (2019). *Autonomous swarm of heterogeneous robots for surveillance operations*. 794–796. https://doi.org/10.1007/978-3-030-34995-0_72

Ørsted. (n.d.). *Image gallery: Borkum Riffgrund 2*. Retrieved December 6, 2025, from https://orsted.de/presse-media/bildergalerie#borkum-riffgrund-2

Padhan, A. K., Sahu, H. K., & Das, B. (2025). Advancing Maritime Communication: A Hybrid PLC/RIS-RF/UAC System with Performance Analysis. *IEEE Transactions on Vehicular Technology*, 1–11. https://doi.org/10.1109/TVT.2025.3591568

Perkins, C., Belding-Royer, E., & Das, S. (2003). *RFC3561: Ad hoc on-demand distance vector (AODV) routing*. Retrieved December 8, 2025, from https://www.rfc-editor.org/rfc/rfc3561.html

Petrioli, C., Petroccia, R., Potter, J. R., & Spaccini, D. (2015). The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks*, *34*, 224–238. https://doi.org/10.1016/j.adhoc.2014.08.012

Petroccia, R., Pelekanakis, K., Alves, J., Fioravanti, S., Blouin, S., & Pecknold, S. (2018). *An adaptive cross-layer routing protocol for underwater acoustic networks*. 1–5. https://doi.org/10.1109/UComms.2018.8493225

Raspberry Pi Ltd. (2025, January). *Raspberry Pi5*. Retrieved December 8, 2025, from https://pip-assets.raspberrypi.com/categories/892-raspberry-pi-5/documents/RP-008348-DS-3-raspberry-pi-5-product-brief.pdf?disposition=inline

Ribeiro, J. P., Fontes, H., Lopes, M., Silva, H., Campos, R., Almeida, J. M., & Silva, E. (2017). *UAV cooperative perception based on DDS communications network*. 1–8. Retrieved December 8, 2025 from https://ieeexplore.ieee.org/abstract/document/8232324

S. Peng, Y. Wang, H. Xiao, & B. Lin. (2020). Implementation of an Improved AODV Routing Protocol for Maritime Ad-hoc Networks. *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 7–11. https://doi.org/10.1109/CISP-BMEI51763.2020.9263519

Schaller, C. (2024). *Spionage und Sabotage vor Europas Küsten-kritische Infrastruktur im Fadenkreuz: Völkerrechtliche Spielräume für Abwehrmaßnahmen*. Retrieved December 8, 2025, from https://www.swp-berlin.org/publikation/spionage-und-sabotage-vor-europas-kuesten-kritische-infrastruktur-im-fadenkreuz

Shaf, A., Ali, T., Farooq, W., Draz, U., & Yasin, S. (2018). Comparison of DBR and L2-ABF routing protocols in underwater wireless sensor network. *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 746–750. https://doi.org/10.1109/IBCAST.2018.8312305

Shang, Z., Yu, H., Li, M., & Li, H. (2025). *Inter cluster routing algorithm for air sea cross domain clustering network based on improved grey wolf optimization algorithm*. 138–144. https://doi.org/10.1109/WCCCT65447.2025.11027998

Shang, Z., Zhang, X., & Li, X. (2025). Maritime communication networks: A survey on architecture, key technologies, and challenges. *Computer Communications*, *241*, 108255. https://doi.org/10.1016/j.comcom.2025.108255

Shechtman, O. (2013). The Coefficient of Variation as an Index of Measurement Reliability. In S. A. R. Doi & G. M. Williams (Eds.), *Methods of Clinical Epidemiology* (pp. 39–49). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-37131-8_4

Shi, J., Wu, J., Zhao, Z., Qi, X., Zhang, W., Qiao, G., & Zuo, D. (2024). A Lightweight Secure Scheme for Underwater Wireless Acoustic Network. *Journal of Marine Science and Engineering*, *12*(5). https://doi.org/10.3390/jmse12050831

Sill Torres, F. (2025). Resilienz maritimer Kritischer Infrastrukturen. *Rechtsfragen Zur Resilienz Maritimer Infrastrukturen*, *18*, 13–28. Retrieved December 8, 2025, from https://elib.dlr.de/214391/1/Sill%20Torres%2C%202025%20-%20Resilienz%20maritimer%20Kritischer%20Infrastrukturen.pdf

Singh, S. K., & Tagore, N. K. (2019). UNDERWATER BASED ADHOC NETWORKS: A brief survey to its challenges, feasibility and issues. *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, 20–25. https://doi.org/10.1109/ICSPC46172.2019.8976642

Stepanov, G. V., Bakhtin, A. A., Muratchaev, S. S., Vinnichenko, A. I., & Riedel, V. V. (2023). Deploying MANET on a Raspberry Pi 4 Model B Test Bench Using the B.A.T.M.A.N. Protocol. *2023 Seminar on Digital Medical and Environmental Systems and Tools (DMEST)*, 139–143. https://doi.org/10.1109/DMEST60476.2023.10339567

Stojanovic, M. (2007). On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, *11*(4), 34–43. https://doi.org/10.1145/1161039.1161049

Su, Y., Fan, R., & Jin, Z. (2019). ORIT: A transport layer protocol design for underwater DTN sensor networks. *IEEE Access*, *7*, 69592–69603. https://doi.org/10.1109/ACCESS.2019.2918561

Sureshbhai, T. H., Mahajan, M., & Rai, M. K. (2018). An Investigational Analysis of DSDV, AODV and DSR Routing Protocols in Mobile Ad Hoc Networks. *2018 International Conference on Intelligent Circuits and Systems (ICICS)*, 281–285. https://doi.org/10.1109/ICICS.2018.00064

Tanenbaum, A., & Wetherall, D. (2011). *COMPUTER NETWORKS FIFTH EDITION*. (5th ed.). Pearson

The MathWorks, Inc. (2025). *Sonar Equation*. Retrieved December 8, 2025 from https://de.mathworks.com/help/phased/ug/sonar-equation.html

TKMS / ATLAS ELEKTRONIK GmbH SeaCat Mk1 image Available at https://www.tkmsgroup.com/atlas-elektronik/mine-warfare-systems/seacat-mk1. Accessed: 06.12.2025

Tong, L., Qilu, Z., & Linbo, Z. (2016). Modified AODV routing protocol in underwater acoustic networks. *2016 IEEE International Conference on Electronic Information and Communication Technology (ICEICT)*, 191–194. https://doi.org/10.1109/ICEICT.2016.7879681

Tréhu, J., & Roberts, M. (2024). Transatlantic Tech Bridge: Digital Infrastructure and Subsea Cables, a US Perspective. *Instituto Affari Internazionali*, *28*. Retrieved December 8, 2025 from https://www.iai.it/en/pubblicazioni/c03/transatlantic-tech-bridge-digital-infrastructure-and-subsea-cables-us-perspective

Ulrick, R. J. (2013). *Principles of Underwater Sound* (3rd ed.). Peninsula Publishing.

Wang, X., Liu, M., Zhang, S., & Dong, S. (2023). UWIA-AODV: Improved Ad-Hoc On-Demand Distance Vector Routing Protocol for Underwater Acoustic Sensor Networks. *2023 42nd Chinese Control Conference (CCC)*, 6176–6181. https://doi.org/10.23919/CCC58697.2023.10240911

Wang, Y.-H., & Chao, C.-F. (2006). Dynamic backup routes routing protocol for mobile ad hoc networks. *Information Sciences*, *176*, 161–185. https://doi.org/10.1016/j.ins.2004.09.016

Xu, Y., & Che, C. (2019). A Brief Review of the Intelligent Algorithm for Traveling Salesman Problem in UAV Route Planning. *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 1–7. https://doi.org/10.1109/ICEIEC.2019.8784651

Yan, B., Wang, L., Su, Y., Tian, Y., & Lu, B. (2021). *A novel routing metric for underwater optical wireless communication using ns-3*. 44–51. https://doi.org/10.1109/BigCom53800.2021.00024

Yang, S., Liu, X., & Su, Y. (2023). A traffic-aware fair MAC protocol for layered data collection oriented underwater acoustic sensor networks. *Remote Sensing*, *15*(6), 1501. https://doi.org/10.3390/rs15061501

Yu, B., Zuo, X., & Gao, S. (2024). Integrated Communication and Topology Design for Cross-domain Collaboration between Space and Marine Robots. *2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC)*, 261–265. https://doi.org/10.1109/EIECC64539.2024.10929574

Zafar, S., Tariq, H., & Manzoor, K. (2016). Throughput and delay analysis of AODV, DSDV and DSR routing protocols in mobile ad hoc networks. *International Journal of Computer Networks and Applications (IJCNA)*, *3*(2), 1–7.

Zhang, H., & Guo, J. (2017). Application of manet routing protocol in vehicular ad hoc network based on NS3. *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 391–394. https://doi.org/10.1109/ICEIEC.2017.8076589

Zhang, Q., Chen, R., Ma, S., & Liang, A. (2024). Nord stream 2, geopolitical conflicts and energy security: Evidence from EU regions. *Energy*, *313*, 133836. https://doi.org/10.1016/j.energy.2024.133836

Zhang, S., Yu, C., Yan, J., Yang, X., & Luo, X. (2024). *Cross-Domain Routing Optimization Based on Improved Multiobjective Genetic Algorithm*. 1–6. https://doi.org/10.1109/COA58979.2024.10723660

Zhu, R., Boukerche, A., Feng, L., & Yang, Q. (2023). A trust management-based secure routing protocol with AUV-aided path repairing for Underwater Acoustic Sensor Networks. *Ad Hoc Networks*, *149*, 103212. https://doi.org/10.1016/j.adhoc.2023.103212

# Declaration

I hereby declare that I have written the dissertation “Self-Organizing Ad-Hoc Routing for maritime Multi-Domain Systems” independently, using only the sources and tools listed. I confirm that I have complied with the general principles of good scientific practice of the Carl von Ossietzky Universität Oldenburg.

Oldenburg, 16 December 2025

Konstantin Johannes Brandl