Robust Communication Through Collective Adaptive Relay Schemes for Maritime Vessels

Published Version:

Baiardi, M., Pianini, D., Al-Falouji, G., Tomforde, S. (2025). Robust Communication Through Collective Adaptive Relay Schemes for Maritime Vessels. New York : Institute of Electrical and Electronics Engineers Inc. [10.1109/ACSOS66086.2025.00019].

Availability:

This version is available at: https://hdl.handle.net/11585/1036570 since: 2026-01-12

Published:

DOI: http://doi.org/10.1109/ACSOS66086.2025.00019

Robust Communication through Collective Adaptive Relay Schemes for Maritime Vessels

Martina Baiardi, and Danilo Pianini
Department of Computer Science and Engineering
University of Bologna
Cesena, Italy
{m.baiardi, danilo.pianini}@unibo.it

Ghassan Al-Falouji, and Sven Tomforde
Department of Computer Science
University of Kiel
Kiel, Germany
{gaf, st}@informatik.uni-kiel.de

Abstract—**Maritime communication networks face unique challenges due to the dynamic and sparse distribution of vessels, variable environmental conditions, and heterogeneous technological constraints. With the increasing trend toward autonomy in maritime operations, these challenges become more pronounced. Modern maritime navigation systems integrate numerous high-bandwidth sensors (including cameras and LiDAR) to enhance environmental perception, whose exploitation generates increased data rate demand. The increase is in contrast to traditional ship communication systems, which provide data rates in the order of kilobits per second. This paper proposes robust, multi-mean, collective adaptive software infrastructures to resiliently improve data collection by relaying data streams across multiple vessels. In particular, we introduce a method to form dynamic clusters of vessels whose information is summarised and then transmitted, raising the probability that the information reaches its destination. We validate our approach through simulation and show that the proposed clustering mechanism is capable of scaling up as new vessels are equipped with improved communication technologies. The research provides practical guidelines for the implementation of self-adaptive communication schemes in maritime environments, advancing the development of resilient communication systems capable of supporting real-time coordination, environmental monitoring, and emergency response for autonomous maritime operations.**

Index Terms—**Maritime Internet of Things, Aggregate Programming, Maritime communication network, Self-adaptive Networks, Cluster-based Routing, Self-organising Coordination Regions**

I. INTRODUCTION

Maritime transportation forms the cornerstone of global trade, with the volume of international maritime trade reaching $12,292$ million tons in 2023, representing 80% of world trade [1]. The United Nations Conference on Trade and Development anticipates that this volume will expand at an annual growth rate of 2% in 2024, driven by the resilience of the global economy [2, 3].

Currently, efforts are being made in the maritime industry to improve efficiency, safety, and sustainability of its operations. Rapid transformation is forming towards the automation of maritime operations, including the progression to Maritime Autonomous Surface Ships (MASS) [4]. These autonomous or semi-autonomous vessels rely heavily on the sophisticated integration of sensor suites for improved perception of the operating environment (including high-resolution cameras, Light Detection and Ranging (LiDAR), and RADAR) to achieve comprehensive Situational Awareness (SA), enabling functions such as real-time environmental perception [5] and Collision Avoidance (COLAV) [6].

Currently, commercially deployed MASS still depend on qualified human operators (either embarked as a safety crew or stationed in shore-based remote operation centres) and this human involvement is expected to remain indispensable throughout a prolonged transition in which autonomous and conventionally manned vessels will sail in parallel [7].

The transition to fully autonomous maritime systems demands reliable SA, which in turn depends on advanced sensor suites and a supporting communication infrastructure that surpasses the capabilities of traditional maritime systems. Modern maritime sensors – such as cameras, LiDAR, and radar – generate high-throughput data streams, necessitating communication links with high bandwidth, low latency, and high reliability [8]. Accordingly, selecting an appropriate data rate is critical to realistically evaluate the performance of any proposed maritime communication solution.

Maritime communication networks (MCNs) face unique challenges compared to terrestrial networks, including dynamic vessel distribution, variable environmental conditions, and diverse technological constraints. Traditional Very-High Frequency (VHF)-based systems, such as Automatic Identification System (AIS), provide limited bandwidth ($\sim 10\,\mathrm{kbit/s}$) and range (typically $40\,\mathrm{km}$, weather dependent). Satellite Communication (SatCom) offers global coverage at the price of high latency and costs (20-$50\,\mathrm{ms}$ for Low Earth Orbit (LEO) constellations). Cellular coverage extends $15\,\mathrm{km}$ from ashore, while Wi-Fi is limited to $\sim 1\,\mathrm{km}$ off the coast [9]. These challenges become more pronounced for autonomous operations in Ship-to-Vessel (S2V) and Ship-to-Shore (S2S) communication, particularly in COLAV systems that require real-time event detection and response.

Reliable communication with shore-base stations is essential for effective system management, emergency response, and

This research has been partly funded by: German Federal Ministry for Digital and Transport within the project “CAPTN Förde Areal II - Praxisnahe Erforschung der (teil)autonomen, emissionsfreien Schifffahrt im digitalen Testfeld” (45DTWV08D) and “WOOD4.0 - Woodworking Machines for Industry 4.0”, (CUP E69J22007520009) Emilia-Romagna regional project, call 2022, art. 6 L.R. N. 14/2014.

real-time data exchange in autonomous maritime operations. Accordingly, S2S communication is firmly embedded in actual maritime regulatory frameworks and operational protocols, as established by the International Maritime Organisation (IMO) [10].

To address existing limitations in such communication infrastructures, this paper proposes novel decentralised techniques to improve data transport and collection in maritime environments. We investigate the use of adaptive relay mechanisms, specifically employing Minimum Spanning Trees (MSTs) with different metrics and cluster-based data routing strategies, to establish efficient multi-hop communication pathways. In particular, in the cluster-based approach, we introduce the idea of collecting data from multiple vessels using high data rate communication channels, then summarising those streams in a leader node and forward to the next relay a summary of the entire cluster information.

The key contributions of this paper are the following.

(i) We propose two MST and one clustering-based self-adaptive decentralised relay mechanisms enabling multi-mean data transport in maritime environments;
(ii) we provide a reference implementation of the proposed approaches using Aggregate Programming; and
(iii) we validate the proposed approaches in simulation, using real-world data from the Kiel Fjord and canal in Germany, comparing the proposed approaches with a non-cooperative baseline.

The remainder of this paper is organised as follows. Section II describes the technological background of this article, with a focus on emerging maritime communication technologies and aggregate computing strategies. Subsequently, Section III presents the methodology used to develop adaptive relay mechanisms for the resilient multi-hop communication pathway. Section IV details the experimental setup to demonstrate the effectiveness of the proposed approach in a real maritime navigation scenario. Finally, Section VI summarises the paper and gives an overview of future work. This work uses several acronyms, which are summarised in Table I.

Acronym	Meaning
AIS	Automatic Identification System
AP	Aggregate Programming
APRS	Automatic Packet Reporting System
ASV	Autonomous Surface Vessel
CEST	Central European Summer Time
CSC	Collective Summarisation Clusters
COLAV	Collision Avoidance
COLREG	International Regulations for Preventing Collisions at Sea
DR-MR	Data Rate-based Multi-Relay Communication
Dist-MR	Distance-based Multi-Relay Communication
DSL	Domain-Specific Language
DST	Daylight Saving Time
FC	Field Calculus
GPS	Global Positioning System
GT	Gross Tonnage
GPX	GPS eXchange Format
IMO	International Maritime Organisation
LEO	Low Earth Orbit
LiDAR	Light Detection and Ranging
LoRaWAN	Long Range Wide Area Network
LoS	Line of Sight
MASS	Maritime Autonomous Surface Ships
MCN	Maritime communication network
MIoT	Maritime Internet of Things
ML	Machine Learning
MMSI	Maritime Mobile Service Identity
MST	Minimum Spanning Tree
RF	Radio Frequency
S2S	Ship-to-Shore
S2V	Ship-to-Vessel
SA	Situational Awareness
SatCom	Satellite Communication
SCC	Shore Control Centre
SCR	Self-organising Coordination Regions
SOLAS	International Convention for the Safety of Life at Sea
UAV	Unmanned Aerial Vehicles
UTC	Universal Time Coordinated
VHF	Very-High Frequency

TABLE I: List of acronyms used throughout the paper

II. BACKGROUND

Maritime communication systems are a cornerstone of safe and efficient navigation, supporting critical operations such as vessel tracking, collision avoidance, and environmental monitoring. Traditional maritime communications are based on a combination of VHF radio systems, and SatCom technologies to ensure safety and operational efficiency [8], However, modern vessels incorporate advanced sensor suites – such as vision-based sensors – that generate data volumes far exceeding the capacity of conventional transmission systems. For instance, VHF-based systems, such as AIS, enable short-range, Line of Sight (LoS) communication between vessels (S2V) and shore-based (S2S) stations for navigation-relevant data exchange (such as identification, position, course, and speed), enhancing situational awareness; yet they offer limited data throughput, see Figure 1. Meanwhile, SatCom systems provide global coverage, albeit often with significant latency and cost [9].

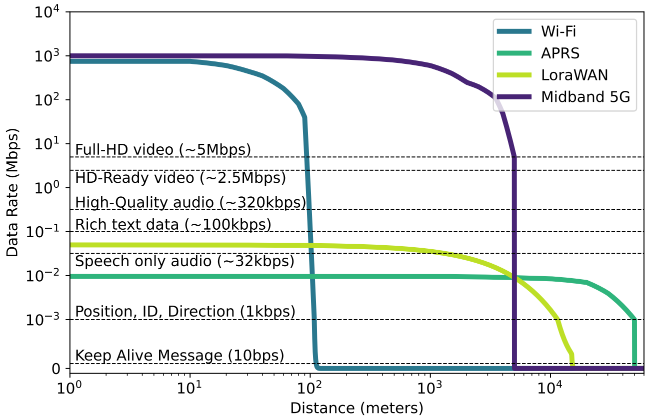


Fig. 1: Estimated LoS data rates for each technology with distance. Horizontal dashed lines show typical data rates for common network uses.

To underscore the demands of MASS, consider the *WaveLab* ship – our test bench for autonomous navigation – which integrates four 4K cameras, a LiDAR, a radar, and additional nautical sensors [11]. Such setup produces substantial data streams: each camera's compressed 720p video (1280×720 at 30 fps) typically requires approximately 2–3 Mbps under standard H.264 compression[1][2], while processed LiDAR and radar outputs (summarised point clouds or object tracking) add several additional Mbps when transmitted continuously[3]. This example highlights the widening gap between the modest capacities of traditional systems and the high-bandwidth, low-latency links required for reliable, read-time exchange of advanced maritime sensor data. Consequently, while legacy communication systems remain mandatory for essential safety and routine operations, they no longer suffice for the high-throughput demands posed by modern autonomous maritime applications and rich sensor-driven services.

Robust S2S communication is not only essential for effective system management and real-time data exchange, but is also mandated under current maritime regulatory frameworks. The IMO requires all vessels subject to the International Convention for the Safety of Life at Sea (SOLAS) to maintain continuous S2S communication capabilities, particularly for distress alerting [10]. This includes all passenger ships and cargo vessels of 300 Gross Tonnage (GT) and above operating in international waters [12].

With the advent of MASS, has established Shore Control Centres (SCCs) as critical infrastructure [13]. These centres require high-bandwidth data streams for remote monitoring and control, particularly for autonomous vessels operating in degrees 2-4 of autonomy, as classified by the IMO [14].

A. Emerging Maritime Communication Technologies: Potentials and challenges

Recognising the limitations of traditional systems, there is a growing interest in leveraging modern terrestrial wireless technologies to improve maritime connectivity, particularly in coastal areas, ports, and for specific S2V and S2S applications. Wi-Fi (IEEE 802.11 standards) is commonly used for local area networks onboard, but are also being explored for short-to-medium range S2S links in ports and for opportunistic S2V mesh networking [15]. Although offering high data throughput, Wi-Fi's LoS is limited, and its performance over water can be affected by reflections and interference [9, 15].

In contrast, Long Range Wide Area Network (LoRaWAN) is a promising solution for low-power and long-range applications, ideal to periodically transmit small data packets (e.g., environmental monitoring and vessel telemetry) over vast distances with minimal energy use [16, 17]. However, performance degrades significantly when LoS is obstructed (Fresnel zone blockage $> 40\%$) [16], and limited bandwidth makes it unsuitable for high-throughput real-time monitoring or multimedia streaming [18].

5G mobile networks hold significant promise due to their potential for high bandwidth, low latency, and massive device connectivity [19]. Various research attempts explored the 5G network for smart port operations, remote vessel control, improved situational awareness data transfer, and potentially extended coverage corridors along coastlines [19–21].

Although 5G networks promise transformative capabilities, their deployment in maritime environments faces critical challenges, especially in the context of remote control [22]. A primary concern is the potential interference with existing maritime and air-based systems operating in the 4800–4990 MHz band, as widely employed in coastal regions [23].

Furthermore, coverage gaps in the open ocean motivate hybrid infrastructure solutions combining satellites, shore-based terrestrial stations, and Unmanned Aerial Vehicless (UAVs). Practical deployment is hindered by harsh maritime conditions that can restrict UAV operations, the complexity of joint resource allocation and interference mitigation in heterogeneous networks, and limited channel state information due to dynamic propagation and high-latency environments [24].

B. Aggregate Programming in Hybrid Maritime Communication Schemes

The limitations of existing maritime communication systems highlight a critical gap between the increasing demand for data and the available infrastructure capabilities. Although traditional approaches provide basic connectivity, they cannot support the bandwidth requirements of modern sensor suites. Similarly, emerging technologies such as 5G offer promising capabilities, but are limited by range and coverage in maritime environments. Integrating these diverse technologies (VHF, 5G, Wi-Fi, LoRaWAN) into a cohesive and efficient maritime communication network presents challenges related to managing heterogeneity, network nodes sparsity, and network dynamic topology to ensure seamless handover. These challenges

[1] https://archive.is/TL1iJ
[2] https://archive.is/DmoIT
[3] https://archive.is/NCLy6

underscore the critical need for resilient and decentralised communication strategies that can overcome the sparse and heterogeneous nature of maritime networks. In particular, approaches capable of dynamically exploiting such diversity in the available communication technologies while adapting to real-time topological changes.

Aggregate Programming (AP) can provide a powerful tool to address these challenges by enabling composable self-organising application-level communication protocols.

C. Aggregate Programming in a nutshell

In this section, we offer a concise overview of AP and its execution model, introducing the core concepts needed to comprehend the algorithms presented.

AP [25] is a functional macro-programming paradigm centred around the notion of a computational field [26]: a distributed data structure that maps points in space and time to values. Building on computational fields, Field Calculus (FC) [26] extends this idea through higher-order functional programming abstractions, allowing manipulation and composition of fields based on local observations of their values. AP has seen successful application in domains that include crowd management [25] and multi-drone coordination [27].

a) Execution model: Within AP, all agents execute a common programme in indefinitely repeated asynchronous[4] rounds. Each round comprises the following phases:

- sense: the agent gathers information from its local environment (including any previous internal state) and from neighbouring agents (by processing any messages received during its dormant phase);
- compute: the agent runs the aggregate programme using the data collected during the sense phase, yielding a result, an updated state, and messages to be shared;
- share: the agent sends messages to neighbouring agents.

b) Observation of neighbouring values: Provided any variable local to the agent, the value of the same variable in neighbouring agents can be observed using **`neighboring`**. **`neighboring`** accepts any type of value and returns a field mapping each neighbouring agent to a value of the same type.

```
fun <T> neighboring(local: T): Field<T>
```

For example, in a line network of three devices: `id1 <-> id2 <-> id3`, the following program:

```
println(neighboring(localId))
```

would print in each device:

- `id1`: `ϕ(loc: id1=>id1, nbrs: [id2=>id2]);`
- `id2`: `ϕ(loc: id2=>id2, nbrs: [id1=>id1,id3=>id3]);`
- `id3`: `ϕ(loc: id3=>id3, nbrs: [id2=>id2]);`

c) Evolution in time and space: To evolve a field through time and space, AP utilises **`share`** [28], which models stateful communication among neighbouring agents. **`share`** takes a local value and a function operating over a field of the same type, producing a new value that is returned and propagated:

```
fun <T> share(intial: T, op: (Field<T>) -> T): T
```

Based on this, we can implement a self-healing variant of the Bellman-Ford algorithm estimating the hop-distance from the nearest device satisfying a certain condition [29]:

```
import Double.POSITIVE_INFINITY as infinity
fun bellmanFord(cond: Boolean): Double =
  share(infinity) { d: Field<Double> ->
    val minDistance = (d + 1.0).minValue(infinity)
    if (cond) 0 else minDistance
  }
```

d) Domain segmentation: The reader can notice that in the previous snippet we computed `minDistance` seemingly uselessly outside of the **`if`** statement. In reality, computing it in the **`else`** branch would break the algorithm. In fact, branching in AP implies domain segmentation: devices taking different branches do not communicate field information, thus, the devices where `cond` holds would not communicate with the others, generating a field of `infinity` values. Domain segmentation can be manually controlled by:

```
fun <T, R> alignedOn(pivot: T, () -> R): R
```

which computes the provided function communicating solely with the devices with the same `pivot`.

e) Computing distances: By estimating the distance to each neighbour, we can extend `bellmanFord` to compute the physical distances. For instance, assuming a function `gps()` returning coordinates and a `haversine()` function calculating the great-circle distance between two coordinates, we write:

```
fun metricDistance(): Field<Double> = neighboring(gps())
  .mapValues { otherPos -> haversine(gps(), otherPos) }
```

Bellman-Ford can now produce distance estimates:

```
import Double.POSITIVE_INFINITY as infinity
fun distanceTo(cond: Boolean, metric: Field<Double>) =
  share(infinity) { d: Field<Double> ->
    val minDistance = (d + metric).minValue(infinity)
    if (cond) 0.0 else minDistance
  }
```

f) Self-stabilising building blocks: The preceding algorithms are self-stabilising, meaning they are capable of recovering from any transient fault [30]. A library of self-stabilising building blocks has been developed in the context of AP [29], including functions for information propagation (`gradientCast`, including adaptive Bellman-Ford), aggregation, and network symmetry breaking. Importantly, the functional composition of these self-stabilising blocks is itself guaranteed to be self-stabilising.

g) Network partitioning and leader election: Self-stabilising distributed leader election and network partitioning can be achieved using `boundedElection` [31]. Given a “strength” for each candidate device and a maximum allowable diameter, `boundedElection` divides the network into

---

[4] Although not strictly required, it is generally assumed that devices operate at similar frequencies.

regions not larger than the specified diameter, automatically selecting new leaders in response to faults or topology changes, without requiring central coordination.

h) Self-organising coordination regions: Once the network has been partitioned and leaders elected, establishing bidirectional communication channels with the leader (e.g., through a combination of `gradientCast` and `convergeCast`) constitutes an implementation of the Self-organising Coordination Regions (SCR) pattern [32].

In this paper, we use AP to implement a set of decentralised algorithms implementing robust hybrid communication that adapt to the sparse distribution and technological heterogeneity of maritime networks. In the next section, we introduce collective adaptive relay schemes that allow vessels to cooperate in forming cluster-based relay chains that improve data delivery in challenging maritime environments.

III. APPROACH

To address the issues presented in Section I, we propose collaborative approaches to build relay schemes that attempt to maximise the amount of data transmitted overland.

We rely on the following (reasonable) assumptions.

(i) Vessels are equipped with a global positioning device (e.g. Global Positioning System (GPS) or Galileo);
(ii) vessels can use current VHF technologies to transmit, directly or via SatCom relay, their position to their surroundings (in the order of a few kilometres); and
(iii) the order of magnitude of the data rate of a S2V communication can be roughly estimated based on the communication technology, the distance separating vessels, and contextual information (e.g., weather conditions).

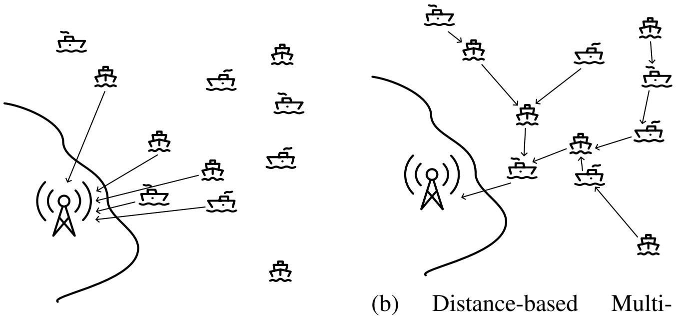
Assumptions 1 and 2 are satisfied using current mandatory AIS technology. Concerning Assumption 3, the key element is that the estimate does not need to be extremely precise, and it is well within the possibilities of current wireless channel measurement and characterisation techniques [33]. Additional precision can lead to better relay choices and therefore better overall performance, but a coarse estimate is sufficient for the proposed approaches to be viable.

All approaches (summarised in Figure 2) assume cooperation among vessels. For simpler data-relay schemes, redirection of the stream to the next vessel is sufficient. The Collective Summarisation Clusters approach, instead, requires the ability to summarise multiple streams into a single, compressed, aggregated, or otherwise size-reduced stream.

For each algorithm, we provide a reference implementation based on AP, using the Collektive aggregate programming Kotlin Domain-Specific Language (DSL)[5].

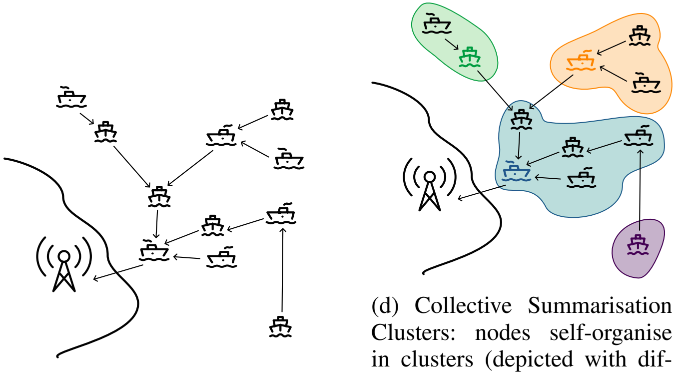
A. Distance-based Multi-Relay Communication

The idea behind Distance-based Multi-Relay Communication (Dist-MR) is: each vessel that cannot reach the land station directly relays its data stream to the vessel along the geographical shortest path to the land station (Figure 2b).



(a) Current state of the art: data streams flow to land stations only if vessels can directly communicate with them.

(b) Distance-based Multi-Relay Communication: each vessel relays the data stream along the geographical shortest path to the station.



(c) Data Rate-based Multi-Relay Communication: each vessel relays the data stream along the highest data-rate channel available towards the station.

(d) Collective Summarisation Clusters: nodes self-organise in clusters (depicted with different colours) with data rate guarantees towards a leader (coloured ship). The leader summarises data and forwards it to a relay in the next cluster towards the land station.

Fig. 2: Current state of the practice and proposed self-organising approaches.

Assuming that each land station can be identified by a variable `landStation`, an implementation of Dist-MR (using elements introduced in Section II-C) is as follows:

```
fun findRelay(origin: Boolean, d: Field<Double>): ID {
  val distanceToStation = distanceTo(landStation, d)
  val neighborDistances = neighboring(distanceToStation)
  // Distance through each neighbour
  val distanceThroughRelay = neighborDistances + d
  // neighbour through which distance is minimum
  val relay = distanceThroughRelay.minByValue().id
  return if (isStation) localId else nextRelay
}
fun distMR(): ID = findRelay(metricDistance())
```

Since every node finds a single relay, except for the land station, the overall network is organised into a set of trees originating from each land station. The relay can then send the local data stream if the data rate of the direct communication channel is sufficient. Unfortunately, there is no guarantee that this is the case: the data rate towards the relay may be insufficient, or the relay may be an intermediate node in a chain whose data rate cannot accommodate both the local and received data streams in the next uplink.

---

[5] https://collektive.github.io

B. Data Rate-based Multi-Relay Communication

Physical distance may not be the best metric for selecting the next relay. Indeed, there may be cases in which passing multiple hops could be more efficient than a single hop with a weak connection. Consider, for instance, the situation in which a vessel is located at $10\,\mathrm{km}$ from the land station, to which it could communicate using a LoRaWAN link with a capacity of approximately $1\,\mathrm{kbit/s}$. At about half the distance, but not in a straight line, there is another vessel which can communicate with the land station using a 5G link with a capacity of $80\,\mathrm{Mbit/s}$. It is very likely that relaying the data stream to the second vessel and then to the land station would have a much higher throughput than a direct connection.

Following this idea, Data Rate-based Multi-Relay Communication (DR-MR) alters the metric of Dist-MR with the goal of selecting the relay which maximises the data rate instead of minimising the distance; as per Figure 2c. Although using the data rate as a metric does not satisfy the triangular inequality, most algorithms that implement adaptive Bellman-Ford variations are robust to this kind of violation. We can thus design a data rate-based pseudometric to replace the distance. In the following example, we assume a function `datarateForDistance(Interface, Double)` which, provided a distance and a network interface, can estimate the data rate of the connection; and a `localInterfaces()` function returning the available network interfaces.

```
fun datarateDistance() = Field<Double> {
  val myIfaces = localInterfaces()
  return neighbouring(myInterfaces)
    .alignedMap(metricDistance()) { ifaces, dist ->
      val datarate = datarateForDistance(
        dist,
        ifaces.intersect(myIfaces)
      )
      1.0 / datarate
    }
}
```

Following this, it is now possible to feed this function to `findRelay`:

```
fun DRMR(): ID = findRelay(datarateDistance())
```

C. Collective Summarisation Clusters

In environments where connectivity is sparse and data rates are low and unevenly distributed, plain multi-hop communication may often fail at delivering the data stream. In maritime communication, data relevance tends to increase with proximity to the vessel that generates it. Furthermore, when vessels are closely located, aggregated information about the collective context often holds greater value than individual data streams. In other words, high-fidelity data can be collected locally, then summarised at strategic points within the network and forwarded toward the land station—potentially undergoing additional summarisation en route.

This process relies on inter-vessel cooperation. Even assuming each vessel can summarise data streams, it remains necessary to determine where data should be aggregated and by whom. Ideally, vessels self-organise into clusters where data can be reliably transmitted (directly or via hops) to a designated summariser (the cluster leader). The leader then compresses the collected streams, including its own, and forwards the summarised output to the land station or to a relay in the next cluster. This concept is illustrated in Figure 2d.

This priority-based leader election, network partitioning, and data summarisation is an instance of the SCR pattern [32]. This distributed data structure can be established on top of a leader election algorithm such as `boundedElection` [31]. It is important for the leader to be a node with a good data rate to relay on the next cluster or to the land station, so the `strength` parameter of `boundedElection` that selects the most suitable leader node should take this into account. As a first step, we propagate the same set of MSTs we use for DR-MR, and we run an election using the upload data rate as the strength. As bound for the leader influence area, we select data rate that guarantees that the data stream can be transmitted to the leader, so that we are guaranteed that inside the cluster these streams can be delivered to the leader without loss. Assuming that we have a constant `MAX_DR` providing such information, we can assign a leader to each vessel using:

```
val metric = datarateDistance()
val drDistance = distanceTo(landStation, metric)
fun findLeader(): ID =
  boundedElection(-drDistance, 1.0 / MAX_DR, metric)
```

Once this operation is done, we need to build the intra- and inter-cluster communication channels, as well as letting the leader know that it has been elected and should thus summarise the received data streams. Understanding whether we are leader is straightforward:

```
val leader = findLeader()
val imLeader = leader == localId
```

The inter-cluster communication can be built by selecting, in neighbouring clusters, the device to which we can communicate using the highest data rate. Contextually, in AP, we can detect whether we are relaying information from other clusters to our cluster leader. Once a relay to the next cluster has been selected, the data rate estimation provides information about how much the cluster data needs to be summarised for the channel to be able to accommodate the data stream.

```
val drAround = neighboring(drDistance)
val potRelays: Field<Boolean> = neighboring(leader)
  .alignedMap(drAround) { l, d ->
    l != leader && d < drDistance
  }
val dists = metric + drAround
val myRelay = potRelays.alignedMap(dists) { r, d ->
  if (r) d else Double.POSITIVE_INFINITY
}
val imRelay = neighboring(myRelay).any { it == localId }
val upstreamDR = 1 / metric[myRelay]
```

For intra-cluster communication, we need to build a MST internal to the cluster. We can do so by exploiting domain segmentation (cf. Section II-C), then finding the relay with maximum data rate:

```
val intraClusterRelay = alignedOn(leader){//Intra-cluster
  val toLeader = distanceTo(imLeader, metric)
  neighboring(toLeader + metric).minByValue(infinity).id
}
```

At this point, every device: (i) is associated to a leader; (ii) is aware of whether or not is a leader; (iii) if it is a leader, can estimate the summarisation factor; (iv) if it is a leader, is aware of the relay on the next cluster; (v) is aware of whether or not is a relay for another cluster; and (vi) if none of the previous, is aware of the relay internal to its cluster.

IV. EXPERIMENTAL EVALUATION

We evaluated our approaches through a simulation of data collection in a realistic maritime environment using real-world navigation data from the Kiel Fjord. Provided the trajectories of the ships, we applied our proposed relay schemes and measured whether and how much data could have been collected ashore. For the approach described in Section III-C, we also measure the degree of summarisation that the leader nodes would have needed to perform. Since 5G antennas are predicted to be increasingly installed on vessels, we observe the impact of a growing percentage of vessels capable of relaying 5G data streams on the organisation of the clusters and the overall data rate.

A. Scenario description

We conducted our experiment based on the AIS data from the Kiel Fjord, cf. [5]. More precisely, we used data from a six-hour time window from August 18, 2022, from 4:00 to 10:00 Universal Time Coordinated (UTC) (6:00 to 12:00 Central European Summer Time (CEST)—Kiel local time).

We assume each vessel to be equipped with:

- a VHF Automatic Packet Reporting System (APRS) device and a LoRaWAN class-C (bidirectional communication) device for long-range communication;
- a 5G consumer-grade module (similar to the ones used in mobile phones); and
- a Wi-Fi-direct capable device.

Furthermore, we assume that a vessel can be equipped with a probability 5G cellular tower p_{5G}. Vessel-mounted 5G towers are capable of communicating with consumer grade 5G modules on surrounding vessels and build wireless back-haul links to surrounding cellular towers (both ashore and mounted on other vessels). For the S2V backhaul communication, we make the very conservative hypothesis [34] that the backhaul link has the same capacity as the link between the consumer grade 5G module and the cellular tower.

We use openly available data[6] to locate the positions of the land stations, which act as the final destination of the collected data. Similarly, we use openly available data[7] to understand where the existing 5G infrastructure is located and their specific technology. We learnt that almost all stations deployed in the Kiel area use mid-band 5G, so we located them in their correct geographical position in the simulation and configured their wireless capabilities accordingly. We were unable to find open data on the backhaul of the 5G towers on land, so we conservatively[8] assumed a fibre backhaul link with a capacity of $10\,\mathrm{Gbit/s}$.

We assume that all ships intend to communicate a data stream $d = 3\,\mathrm{Mbit/s}$ to the land station, approximately equivalent to the bitrate of a compressed 720p video stream.

B. Data Rate Estimation

For the proposed algorithms to operate, we needed a data rate estimate of every potential communication link. Provided the network configuration introduced in Section IV-A, we relied on the current literature to estimate the LoS data rate for 5G [35, 36], Wi-Fi [37], LoRaWAN[9] [38], and APRS[10].

Provided the reference data rates at key distances, we ran a log-linear interpolation of the data extracted from the aforementioned sources collecting the data presented in Figure 1.

C. Baseline and algorithms under test

As a baseline, we use a non-collective approach in which all ships try to reach the land station directly using the most favourable communication technology available to them (the current state-of-the-art in Figure 2a). We compare this approach with Dist-MR (cf. Section III-A), DR-MR (cf. Section III-B), and Collective Summarisation Clusters (CSC) (cf. Section III-C).

D. Free variables

We investigate the behaviour of our simulated systems with respect to two free variables.

p_{5G} Controlling the probability of a vessel being equipped with a 5G tower (see Section IV-A); and
f controlling the aggregate programs round frequency.

We let p_{5G} range in $[0, 0.01, 0.02, 0.05, 0.1, 0.5, 1]$. The idea is to learn how the proposed approaches benefit from the increasingly widespread use of 5G on vessels. Indirectly, this also provides hints on the expected return from the installation of an increasingly large amount of telecommunication infrastructure on vessels.

Instead, f provides information about the overhead these approaches introduce in the system. If results are poor at low frequencies but improve at higher frequencies, it is an indication that topology changes are too frequent for the aggregate programme to cope with. Although there are proposals for reactive versions that could self-optimise their execution [39, 40], in this case we are still interested in understanding whether sparse evaluations (e.g., on average one per minute) are sufficient to provide a good performance. We let f range in $[1/60, 0.1, 0.2]\mathrm{Hz}$.

---

[6] https://archive.is/QNhyS

[7] https://archive.is/fTMSV

[8] From survey in a 2023 IEEE ComSoc Technology Blog article (https://archive.is/qdsmT), 93% of the surveyed network operators expected the backhaul capacity to be equal to or larger than $10\,\mathrm{Gbit/s}$ in 2025.

[9] https://archive.is/EnvEC

[10] https://archive.is/MEVf7

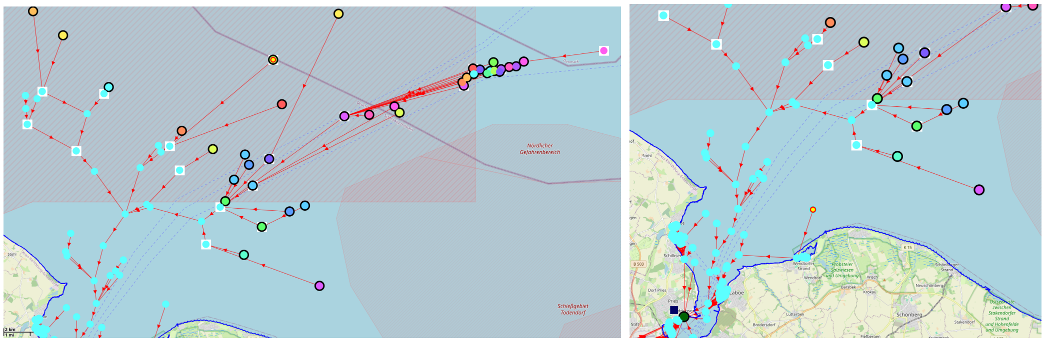


Fig. 3: Simulation snapshots of the Collective Summarisation Clusters algorithm. Ships (dots) are organised in clusters identified by the dot colour, whose leader is depicted with a thick black border. Leaders of clusters afar from the land station summarise information and deliver the summarised data stream to the relay of the next cluster (highlighted with a white square). The snapshot shows several clusters with a single participant in the upper part of the picture. This phenomenon is due to the link data rate being insufficient to sustain multi-vessel clusters. On the contrary, the light-blue cluster in the bottom part of the picture is composed of multiple vessels, as close to the shore they can use high data rate 5G connections.

E. Metrics

To compare the approaches, we estimate the actual data rate achieved by each vessel in its communication with the selected land station. For each algorithm, we compute the actual vessel n data rate DR_n^a as follows.

a) baseline: We measure the data rate of the direct communication link with the land station. If it is sufficient to transmit the data stream, then we consider the data stream as transmitted and discard the connection otherwise.

b) Dist-MR and DR-MR: We follow the spanning tree towards the land station and, at each hop, measure whether the data rate of the hop is enough to forward the data stream. We do so by prioritising the local data stream, then dividing the remaining data rate by the node's children count, and maintaining the data stream alive if the capacity is sufficient to transmit all the children's data streams. More formally, we compute the data transmission rate DR_n^f for each node n with an upstream data transmission rate DR_n^u and C children as: $DR_n^f = \frac{DR_n^u - d}{C}$, and we transmit the data stream if $DR_n^f > d$.

c) CSC: By cluster construction, once a vessel successfully joins a cluster, we are guaranteed that we can stream at least d to the leader node (directly or through intermediate hops). Hence, starting from the data rate d_i ($d_i = d$ when the computation starts), we look at the leader node, apply the reduction factor RF to d_i, and obtain the inter-cluster streaming capacity for this node $d_o = d_i RF$. We forward d_o to the next cluster as d_i, and repeat the process until we reach the land station.

For CSC, to better understand their behaviour, we measure some dedicated metrics.

Reduction factor RF_c: For each cluster c, this metric measures the compression (or drop, or summarisation) rate performed by the leader node when aggregating the data streams from its children before forwarding the data stream to its relay (or land station). We compute it as the ratio between the upstream data rate and the sum of all the data streams reaching the leader node, including the leader node stream; if the upload stream is larger than all those streams, the RF maxes out to 1 (no compression, drop, or summarisation required). More formally, for a cluster with C nodes, in which the upstream data rate of the leader is DR^u, $RF = min\left(1, \frac{DR^u}{dC}\right)$. Additionally, we extend the concept to every node: for any node $n \in c$, $RF_n = RF_c$.

Cluster size: The average number of vessels in a cluster. Clusters should get larger when better communication technologies are available so that summarisations can be more informative and less lossy.

Singleton clusters: The number of clusters with only one vessel, namely, vessels responsible to compress their own data stream and send it directly to their next relay, but with no need to summarise or aggregate data streams from other vessels.

Number of clusters: The number of clusters formed in the network. In general, the smaller the number of clusters and the larger the number of clusters, the higher the reduction factor, and thus the lower the lossiness of the data stream. The number of clusters is related to the number of singleton clusters and the size of the cluster.

F. Reproducibility and open science

Each experiment has been repeated 100 times with different random seeds. Different seeds control which vessels are equipped with 5G towers and the scheduling of AP rounds.

Simulations have been performed using Alchemist [41]. Snapshots of the simulations are shown in Figure 3. The aggregate code has been written using the Collektive DSL[11]. Data have been analysed using xarray [42]; visual reports have been created using matplotlib [43].

The ship movement traces were derived from real-world AIS data and got converted to GPS eXchange Format (GPX) format as follows:

(i) AIS Raw messages are collected for all vessels present during the observation window,

---

[11] https://collektive.github.io

(ii) Messages are decoded following the NMEA 0183[12] standard, this process requires to: concatenate raw messages split in multiple fragments using sequence numbers, validate the message using the checksum, decode the message information, and, finally, parse the payload to extract vessels' data.

(iii) Geospatial coordinates extracted from messages are grouped by ship unique identification number (Maritime Mobile Service Identity (MMSI)) and temporally sorted.

(iv) The sequence of coordinates is used to construct GPX[13]-formatted trajectories. These traces were used directly in the simulation to guide the spatial dynamics of the ships.

The complete simulation, including source code, chart generation script, and configuration files, has been released as open source under a permissive licence[14], and archived on Zenodo [44] for future reference. Since the AIS we relied on is not open and privacy concerns apply, we instead share anonymised GPX traces (MMSI numbers are not tracked). However, we included the AIS conversion programme in the open source release to simplify repurposing of the experiment in other contexts.

G. *Results*

The results for the mean actual data rate DR_n^a for each approach are summarised in Figure 4. In Figure 5, we show some selected combintations of p_{5G} and f over time, to analyse the temporal evolution of the data rate and its stability.

Impact of 5G-equipped vessels: The simulations reveal that equipping vessels with 5G tower backhaul capabilities (p_{5G}) has great potential to improve data transmission rates. However, the current and two of the proposed approaches (Dist-MR and DR-MR) exhibit limited improvements in data rates, even with a significant presence of 5G-equipped vessels. However, even a modest deployment ($p_{5G} = 0.01$) yields measurable improvements for CSC, which consistently outperforms all other strategies regardless of the update frequency or infrastructure setting.

Update frequency sensitivity: The frequency of AP execution (f) has a very modest impact on the performance of the proposed approaches. Even at low frequencies ($f = 1/60$ Hz), we observe stable and consistent results. Higher frequencies help reduce variance (see Figure 5), but the mean speed of ships is compatible with very low round frequencies, meaning that the impact in terms of channel overhead for self-organising algorithms is negligible in most cases.

Algorithm comparison: Among the tested methods, CSC achieves the highest average data rate in all simulation configurations. In particular, CSC is also the approach that scales better as infrastructure improves, demonstrating better adaptability to the presence of 5G equipped vessels. The DR-MR generally underperforms, even compared with Dist-MR and the current baseline. This counterintuitive result is most likely due to vessels with a good data rate being overly selected and ending up with too many streams to relay, thus causing overuse of few channels and underuse of others. These results suggest that the approach could be amended, for instance, by considering the number of children per relay as a factor in the relay selection process.

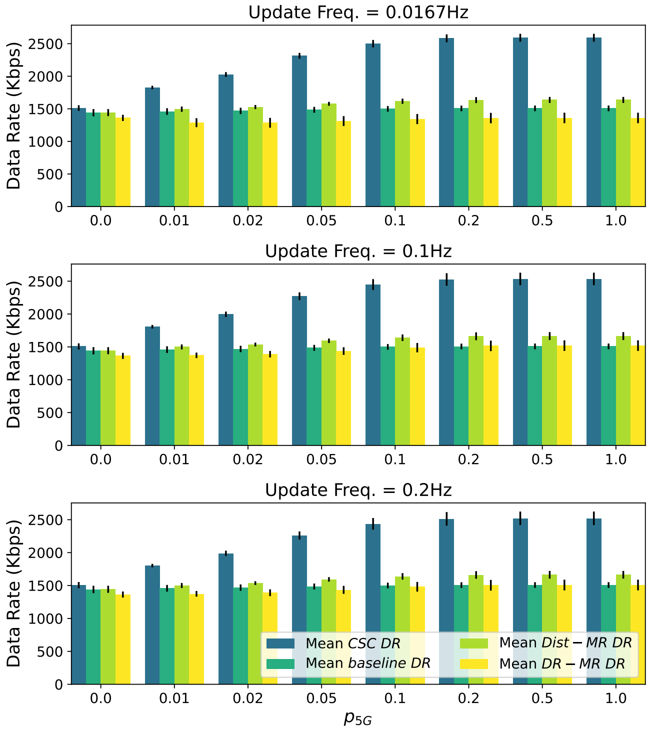


Fig. 4: Mean data rate comparison across different probabilities than a vessel is equipped with a 5G tower (P_{5G}), round execution frequency (f), and different algorithms. The presence of even few vessels equipped with 5G towers makes a significant difference in the data rate achieved by the system. The CSC approach achieves the best performance, in any situation, and it is the approach that better exploits the presence of vessel-mounted 5G towers.

H. *CSC detailed performance analysis*

To better understand the behaviour of the CSC strategy, we analysed its internal dynamics using dedicated metrics (cf. Section IV-E).

Reduction factor: The average reduction factor increases with p_{5G}, approaching a value of 1 when many vessels are equipped with 5G towers. This indicates that cluster leaders can forward a larger portion of the original data streams with minimal summarisation or loss, thereby preserving higher data fidelity in the transmission to land stations. Interestingly, performance reaches a peak at $p_{5G} = 0.2$, and then decreases. Our hypothesis is that, when clusters grow too large, the same problem which affects DR-MR occurs within clusters in CSC: since intra-cluster relay gets selected using the same logic, congestion phenomena may occur.

---

[12] https://archive.is/U7iFK

[13] https://archive.is/YQoiG

[14] https://github.com/anitvam/experiment-2025-acsos-ship-clustered-comm

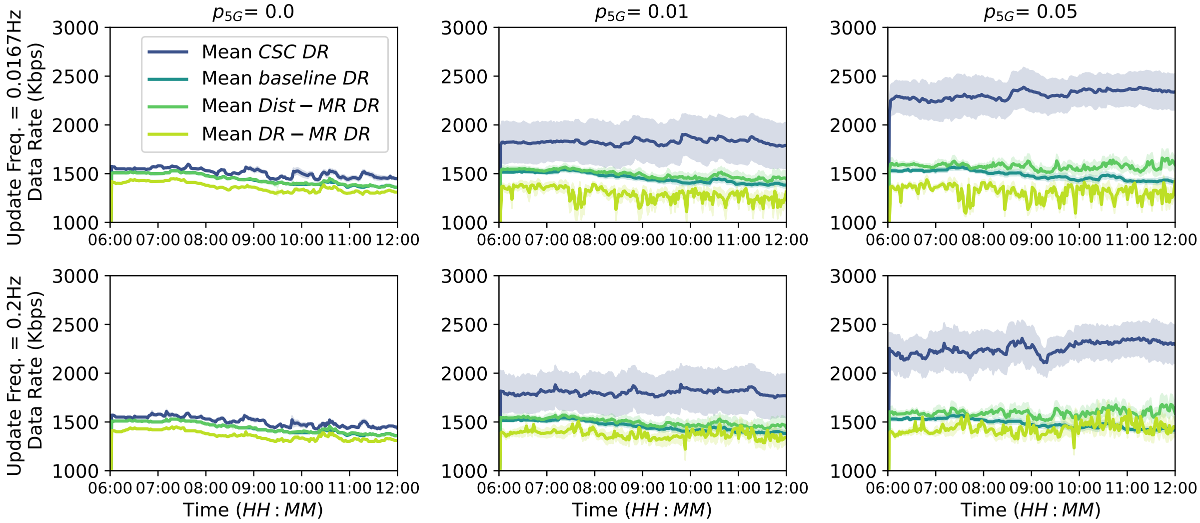


Fig. 5: Temporal evolution of mean data rate for varying probabilities of vessel-mounted 5G towers (P_{5G}), across different algorithms and update frequencies. Solid line are averages, shadows show mean $\pm 1\sigma$. Even a minimal presence of 5G-equipped vessels significantly enhances performance. The CSC approach consistently outperforms all others by better leveraging 5G availability, demonstrating both higher peak and sustained data rates throughout the simulation.

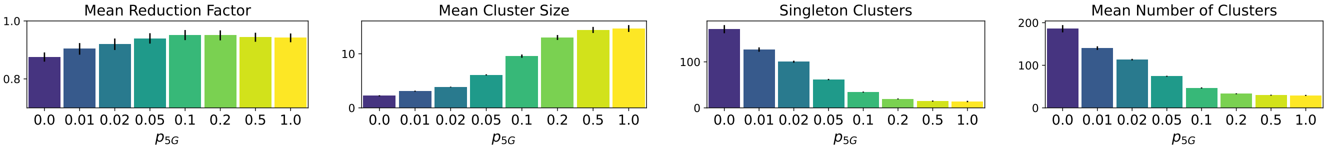


Fig. 6: CSC performance metrics: mean reduction factor (top left), mean cluster size (top right), number of isolated vessels (bottom left), and number of clusters (bottom right). All data has been measured for $f = 0.2$ Hz and for different probabilities of vessel-mounted 5G towers (P_{5G}). The proposed summarisation and relaying approach scales well with P_{5G}, demonstrating good self-optimisation capabilities.

Cluster size and distribution: The average size of the cluster grows with p_{5G}, from approximately 1–2 vessels per cluster in $p_{5G} = 0$ to more than 10 in $p_{5G} = 1.0$, indicating more effective grouping and relay capabilities. Concurrently, the number of singleton clusters sharply declines with increasing 5G availability, reflecting that more vessels can integrate into clusters with viable relaying paths. The total number of clusters decreases with p_{5G} because larger clusters are becoming increasingly convenient.

V. LIMITATIONS AND FUTURE WORK

This section outlines the primary limitations of the present study and highlight directions for future development.

a) Limited data: In our experiment, we relied on real-world data from the Kiel Canal. However, the data is limited to a six-hour time window of a single day, which may not be representative of the entire maritime traffic in the area (or worldwide). Further experimentation will be performed in future work.

b) Corner cases: We do not build synthetic corner cases (e.g., leader-node failures) to investigate the behaviour of the proposed approaches in specific challenging situations. In particular, due to the nature of `boundedElection` [31], a well-positioned, highly connected vessel that flakily joins and leaves the network may cause the clusters to continually restructure, leading to oscillatory behaviour. Construction and evaluation of corner cases is planned for future work.

c) Security: In this paper, we did not focus on security aspects of the proposed approaches. From the discussion of the previous paragraph, however, it is evident how a malicious vessel could join and leave flakily to cause a denial of service. Investigation of adversarial behaviour and security aspects of the proposed approaches need dedicated future studies.

d) Data summarisation: In this work we did not focus on finding summarisation algorithms. Promising research directions include context-aware segmentation, adaptive and progressive compression, natural language summarisation, and other fusion methods aimed at enhancing data efficiency [22]. Summarisation also intertwines tightly with security, power consumption, and latency concerns, each one deserving a future dedicated investigation.

e) Weather conditions: Adverse and mutating weather conditions have not been taken into account in this work. As mentioned in Section III, the system only requires that a reasonable estimate of the data rate of a communication channel can be produced, and such estimate could integrate current weather information. In the future, however, the system should be tested with improved communication modelling, including channel quality degradation due to adverse atmospheric conditions. A related investigation concerns the evaluation of the extent by which imprecise data rate estimations affect the overall quality of the communication.

f) Future improvements: In addition to addressing the aforementioned limitations, future work will focus on practical testing using the existing test carrier, extending the scenarios to heterogeneous and carrier-independent environments, and further approaches for adapting the methods using machine learning. Furthermore, we plan to work to improve the dynamic behaviour of the DR-MR approach to avoid congestion phenomena, and we expect these improvements to indirectly enhance the performance of CSC in high-data rate availability scenarios.

VI. CONCLUSION

Maritime vessels are subject to an increasing trend towards autonomous navigation behaviour that goes far beyond autopilot behaviour on the open sea. Especially in busy coastal environments, appropriate situational awareness is possible as a basis for (semi-)autonomous behaviour through the communication of pure sensor information (such as high-resolution camera, LiDAR and radar information in particular), as well as information derived from this between vessels and the shore. However, the high data volumes and insufficient coverage with the corresponding strong communication carriers of the bandwidth represent limitations for current technology.

In this paper, we present a novel approach for aggregated communication between maritime units. Using data collected in the Kiel Fjord and Kiel Canal environment, we were able to simulate that with a small amount of 5G coverage on vessels, data collection can already be greatly improved, leading to improved situational awareness. In addition, we were able to show that cluster-based methods represent a significant scaling gain through dynamic summaries of information, which means that the existing infrastructure is better utilised than when using simple data relays.

REFERENCES

[1] UN Conference on Trade and Development, Review of Maritime Transport 2023: Towards a Green and Just Transition, UN, Geneva, 2023, [Online]. Available at: https://bit.ly/3VgOItS.

[2] U. C. on Trade, Development, Review of Maritime Transport 2024: Navigating Maritime Chokepoints, UN, Geneva, 2024, [Online]. Available at: https://bit.ly/3ZKBQPn.

[3] DESA, World Economic Situation and Prospects as of Mid-2024, UN, 2024, [Online]. Available at: https://bit.ly/4giZMyn.

[4] G. Al-Falouji, T. Beyer, S. Tomforde, From social robots to autonomous surface vessels' navigation, in: ACSOS-C, IEEE, 2023, pp. 59–64. `doi:10.1109/ACSOS-C58168.2023.00037`.

[5] G. Al-Falouji, S. Gao, L. Haschke, D. Nowotka, S. Tomforde, Enhancing maritime behaviour analysis through novel feature engineering and digital shadow modelling: A case study in the kiel fjord, in: ARCS, Springer, 2024, pp. 97–111. `doi:10.1007/978-3-031-66146-4_7`.

[6] G. Al-Falouji, T. Beyer, S. Gao, S. Tomforde, Steering towards maritime safety with true motion predictions ensemble, in: ACSOS-C, IEEE, 2024, pp. 7–12. `doi:10.1109/ACSOS-C63493.2024.00021`.

[7] X. Li, K. F. Yuen, A human-centred review on maritime autonomous surfaces ships: impacts, responses, and future directions, Transport Reviews 44 (4) (2024) 791–810. `doi:10.1080/01441647.2024.2325453`.

[8] G. Xylouris, N. Nomikos, A. Kalafatelis, A. Giannopoulos, S. Spantideas, P. Trakadas, Sailing into the future: technologies, challenges, and opportunities for maritime communication networks in the 6g era, Frontiers Commun. and Netw. 5 (2024) 1439529. `doi:10.3389/frcmn.2024.1439529`.

[9] F. S. Alqurashi, A. Trichili, N. Saeed, B. S. Ooi, M.-S. Alouini, Maritime communications: A survey on enabling technologies, opportunities, and challenges, IEEE Internet Things J. 10 (4) (2022) 3525–3547. `doi:10.1109/JIOT.2022.3219674`.

[10] International Maritime Organization, Harmonization of gmdss requirements for radio installations on board solas ships, IMO Circular COMSAR.1/Circ.32/Rev.3, International Maritime Organization, 4 Albert Embankment, London SE1 7SR, United Kingdom, [Online]. Available at: https://bit.ly/4nFk35U (dec 2024).

[11] G. Al-Falouji, L. Haschke, D. Nowotka, S. Tomforde, Self-explanation as a basis for self-integration-the autonomous passenger ferry scenario, in: ACSOS-C, IEEE, 2023, pp. 65–70. `doi:10.1109/ACSOS-C58168.2023.00038`.

[12] UK Government, Advice Note 1033: Maritime Safety Information (MSI) Leaflet, Tech. rep., UK Government, [Online]. Available at: https://bit.ly/4liErbk (Apr. 2025).

[13] I. Mickevičiūtė, Challenges and opportunities for seafarer training in the context of a modernizing maritime industry: Research overview, Vilnius Uni. Open Ser. 3 (2020) 110–119.

[14] International Maritime Organization, Report of the Maritime Safety Committee on its 100th session, Tech. Rep. MSC 100/20, International Maritime Organization, [Online]. Available at: https://bit.ly/4lF7LZj (2018).

[15] P. M. d'Orey, M. G. Gaitán, P. M. Santos, M. Ribeiro, J. B. de Sousa, L. Almeida, Assessing short-range shore-

to-shore (s2s) and shore-to-vessel (s2v) wifi communications, Comput. Netw. 249 (2024) 110505. doi:10.1016/j.comnet.2024.110505.

[16] S. Pensieri, F. Viti, G. Moser, S. B. Serpico, L. Maggiolo, M. Pastorino, D. Solarna, A. Cambiaso, C. Carraro, C. Degano, et al., Evaluating lorawan connectivity in a marine scenario, J. Mar. Sci. Eng. 9 (11) (2021) 1218. doi:0.3390/jmse9111218.

[17] J. Pinelo, A. D. Rocha, M. Arvana, J. Gonçalves, N. Cota, P. Silva, Unveiling lora's oceanic reach: Assessing the coverage of the azores lorawan network from an island, Sensors 23 (17) (2023) 7394. doi:10.3390/s23177394.

[18] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, T. Watteyne, Understanding the limits of lorawan, IEEE Commun. Mag. 55 (9) (2017) 34–40. doi:10.1109/MCOM.2017.1600613.

[19] J. Wu, K. Yang, H. Zeng, S. Zhang, J. Ding, X. Jiang, The application of 5g wireless communication in maritime environment, in: J. Phys. Confe. Ser., Vol. 2618, IOP Publishing, 2023, p. 012003. doi:10.1088/1742-6596/2618/1/012003.

[20] S. Horsmanheimo, L. Tuomimäki, A. Gentili, J. Scholliers, A. Sadam, K. K. Kaare, Maritime multi-hop communication over 5g networks: An indoor testbed implementation and evaluation, in: VTC-Fall, IEEE, 2024, pp. 1–6. doi:10.1109/VTC2024-Fall63153.2024.10757897.

[21] N. Agarwala, S. S. K. Guduru, The potential of 5g in commercial shipping, Marit. Technol. Res. 3 (3) (2021) 254–267. doi:10.33175/mtr.2021.248995.

[22] N. Smirnov, S. Tomforde, Real-time data transmission optimization on 5g remote-controlled units using deep reinforcement learning, in: ARCS, Vol. 13949 of LNCS, Springer, 2023, pp. 281–295. doi:10.1007/978-3-031-42785-5_19.

[23] A. Pastukh, V. Sorokin, Evaluation of interference analysis from 5g nr networks to aeronautical and maritime mobile systems in the frequency band 4800–4990 mhz, J. 6 (1) (2022) 17–31. doi:10.3390/j6010002.

[24] X. Li, W. Feng, J. Wang, Y. Chen, N. Ge, C.-X. Wang, Enabling 5g on the ocean: A hybrid satellite-uav-terrestrial network solution, IEEE Wirel. Commun. 27 (6) (2020) 116–121. doi:10.1109/MWC.001.2000076.

[25] J. Beal, D. Pianini, M. Viroli, Aggregate programming for the internet of things, IEEE Comput. 48 (9) (2015) 22–30. doi:10.1109/MC.2015.261.

[26] G. Audrito, M. Viroli, F. Damiani, D. Pianini, J. Beal, A higher-order calculus of computational fields, ACM Trans. Comput. Log. 20 (1) (2019) 5:1–5:55. doi:10.1145/3285956.

[27] D. Grushchak, J. Kline, D. Pianini, N. Farabegoli, G. Aguzzi, M. Baiardi, C. Stewart, Decentralized multi-drone coordination for wildlife video acquisition, in: ACSOS, IEEE, 2024, pp. 31–40. doi:10.1109/ACSOS61780.2024.00021.

[28] G. Audrito, J. Beal, F. Damiani, D. Pianini, M. Viroli, The share operator for field-based coordination, in: COORDINATION, Vol. 11533 of LNCS, Springer, 2019, pp. 54–71. doi:10.1007/978-3-030-22397-7_4.

[29] M. Viroli, G. Audrito, J. Beal, F. Damiani, D. Pianini, Engineering resilient collective adaptive systems by self-stabilisation, ACM Trans. Model. Comput. Simul. 28 (2) (2018) 16:1–16:28. doi:10.1145/3177774.

[30] E. W. Dijkstra, Self-stabilizing systems in spite of distributed control, Commun. ACM 17 (11) (1974) 643–644. doi:10.1145/361179.361202.

[31] D. Pianini, R. Casadei, M. Viroli, Self-stabilising priority-based multi-leader election and network partitioning, in: ACSOS, IEEE, 2022, pp. 81–90. doi:10.1109/ACSOS55765.2022.00026.

[32] D. Pianini, R. Casadei, M. Viroli, A. Natali, Partitioned integration and coordination via the self-organising coordination regions pattern, Future Gener. Comput. Syst. 114 (2021) 44–68. doi:10.1016/J.FUTURE.2020.07.032.

[33] C. Li, J. Yu, J. Xue, W. Chen, S. Wang, K. Yang, Maritime broadband communication: Wireless channel measurement and characteristic analysis for offshore waters, J. Mar. Sci. Eng. 9 (7) (2021) 783. doi:10.3390/jmse9070783.

[34] B. Tezergil, E. Onur, Wireless backhaul in 5g and beyond: Issues, challenges and opportunities, IEEE Commun. Surv. Tutorials 24 (4) (2022) 2579–2632. doi:10.1109/COMST.2022.3203578.

[35] M. I. Rochman, V. Sathya, D. Fernández, N. Nunez, A. S. Ibrahim, W. Payne, M. Ghosh, A comprehensive analysis of the coverage and performance of 4g and 5g deployments, Comput. Netw. 237 (2023) 110060. doi:10.1016/J.COMNET.2023.110060.

[36] R. A. K. Fezeu, J. Carpenter, C. Fiandrino, E. Ramadan, W. Ye, J. Widmer, F. Qian, Z. Zhang, Mid-band 5g: A measurement study in europe and US, CoRR abs/2310.11000 (2023). arXiv:2310.11000, doi:10.48550/ARXIV.2310.11000.

[37] Z. Zhou, Z. Yang, C. Wu, L. Shangguan, H. Cai, Y. Liu, L. M. Ni, Wifi-based indoor line-of-sight identification, IEEE Trans. Wirel. Commun. 14 (11) (2015) 6125–6136. doi:10.1109/TWC.2015.2448540.

[38] D. Magrin, M. Capuzzo, A. Zanella, A thorough study of lorawan performance under different parameter settings, IEEE Internet Things J. 7 (1) (2020) 116–127. doi:10.1109/JIOT.2019.2946487.

[39] D. Pianini, R. Casadei, M. Viroli, S. Mariani, F. Zambonelli, Time-fluid field-based coordination through programmable distributed schedulers, Log. Methods Comput. Sci. Volume 17, Issue 4 (Nov. 2021). doi:10.46298/lmcs-17(4:13)2021.

[40] R. Casadei, F. L. D. Angelis, G. Aguzzi, D. Pianini, M. Viroli, Self-organisation programming: A functional reactive macro approach, in: ACSOS, IEEE, 2023, pp.

87–96. `doi:10.1109/ACSOS58161.2023.00026`.

[41] D. Pianini, S. Montagna, M. Viroli, Chemical-oriented simulation of computational systems with ALCHEMIST, J. Simul. 7 (3) (2013) 202–215. `doi:10.1057/jos.2012.27`.

[42] S. Hoyer, J. Hamman, xarray: N-D labeled arrays and datasets in Python, J. Open Res. Softw. 5 (1) (2017). `doi:10.5334/jors.148`.

[43] J. D. Hunter, Matplotlib: A 2d graphics environment, Computing in Science & Engineering 9 (3) (2007) 90–95. `doi:10.1109/MCSE.2007.55`.

[44] M. Baiardi, G. Al-Falouji, D. Pianini, S. Tomforde, Robust communication through collective adaptive relay schemes for maritime vessels (2025). `doi:10.5281/zenodo.16421102`.